Oracle® Retail Fiscal Management/RMS Brazil Localization

Implementation Guide

Release 13.2.5

E37587-01

November 2012

ORACLE®

Oracle Retail Fiscal Management/RMS Brazil Localization Implementation Guide, Release 13.2.5

E37587-01

Value-Added Reseller (VAR) Language

Oracle Retail VAR Applications

The following restrictions and provisions only apply to the programs referred to in this section and licensed to you. You acknowledge that the programs may contain third party software (VAR applications) licensed to Oracle. Depending upon your product and its version number, the VAR applications may include:

(i) the **MicroStrategy** Components developed and licensed by MicroStrategy Services Corporation (MicroStrategy) of McLean, Virginia to Oracle and imbedded in the MicroStrategy for Oracle Retail Data Warehouse and MicroStrategy for Oracle Retail Planning & Optimization applications.

(ii) the **Wavelink** component developed and licensed by Wavelink Corporation (Wavelink) of Kirkland, Washington, to Oracle and imbedded in Oracle Retail Mobile Store Inventory Management.

(iii) the software component known as **Access Via**™ licensed by Access Via of Seattle, Washington, and imbedded in Oracle Retail Signs and Oracle Retail Labels and Tags.

(iv) the software component known as **Adobe Flex**™ licensed by Adobe Systems Incorporated of San Jose, California, and imbedded in Oracle Retail Promotion Planning & Optimization application.

You acknowledge and confirm that Oracle grants you use of only the object code of the VAR Applications. Oracle will not deliver source code to the VAR Applications to you. Notwithstanding any other term or condition of the agreement and this ordering document, you shall not cause or permit alteration of any VAR Applications. For purposes of this section, "alteration" refers to all alterations, translations, upgrades, enhancements, customizations or modifications of all or any portion of the VAR Applications including all reconfigurations, reassembly or reverse assembly, re-engineering or reverse engineering and recompilations or reverse compilations of the VAR Applications or any derivatives of the VAR Applications. You

acknowledge that it shall be a breach of the agreement to utilize the relationship, and/or confidential information of the VAR Applications for purposes of competitive discovery.

The VAR Applications contain trade secrets of Oracle and Oracle's licensors and Customer shall not attempt, cause, or permit the alteration, decompilation, reverse engineering, disassembly or other reduction of the VAR Applications to a human perceivable form. Oracle reserves the right to replace, with functional equivalent software, any of the VAR Applications in future releases of the applicable program.

Contents

3 Setting up RMS for ORFM and Brazil Localization

4 Integrating ORFM

5   Publication and Subscription API Designs

6   Archiving and Purging Data

7   Integrating with TaxWeb Tax Rules – Retail Tax Integration Layer

8   Working with RTIL

11 Understanding Batch Processes

12 Configuring Security for Web Services

Send Us Your Comments

Oracle Retail Fiscal Management/RMS Brazil Localization Implementation Guide, Release 13.2.5

Oracle welcomes customers' comments and suggestions on the quality and usefulness of this document.

Your feedback is important, and helps us to best meet your needs as a user of our products. For example:

- Are the implementation steps correct and complete?
- Did you understand the context of the procedures?
- Did you find any errors in the information?
- Does the structure of the information help you with your tasks?
- Do you need different information or graphics? If so, where, and in what format?
- Are the examples correct? Do you need more examples?

If you find any errors or have any other suggestions for improvement, then please tell us your name, the name of the company who has licensed our products, the title and part number of the documentation and the chapter, section, and page number (if available).

> **Note:** Before sending us your comments, you might like to check that you have the latest version of the document and if any concerns are already addressed. To do this, access the Online Documentation available on the Oracle Technology Network Web site. It contains the most current Documentation Library plus all documents revised or released recently.

Send your comments to us using the electronic mail address: retail-doc_us@oracle.com

Please give your name, address, electronic mail address, and telephone number (optional).

If you need assistance with Oracle software, then please contact your support representative or Oracle Support Services.

If you require training or instruction in using Oracle software, then please contact your Oracle local office and inquire about our Oracle University offerings. A list of Oracle offices is available on our Web site at `http://www.oracle.com`.

Preface

The *ORFM/RMS Brazil Localization Implementation Guide* provides detailed information that is important when implementing Oracle Retail Fiscal Management (ORFM).

Audience

This document is intended for the Oracle Retail Fiscal Management application integrators and implementation staff, as well as the retailer's IT personnel.

Documentation Accessibility

Our goal is to make Oracle products, services, and supporting documentation accessible to all users, including users that are disabled. To that end, our documentation includes features that make information available to users of assistive technology. This documentation is available in HTML format, and contains markup to facilitate access by the disabled community. Accessibility standards will continue to evolve over time, and Oracle is actively engaged with other market-leading technology vendors to address technical obstacles so that our documentation can be accessible to all of our customers. 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.

Accessibility of Links to External Web Sites in Documentation

This documentation may contain links to Web sites of other companies or organizations that Oracle does not own or control. Oracle neither evaluates nor makes any representations regarding the accessibility of these Web sites.

Access to Oracle Support

Oracle customers 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.

Accessibility of Code Examples in Documentation

Screen readers may not always correctly read the code examples in this document. The conventions for writing code require that closing braces should appear on an otherwise empty line; however, some screen readers may not always read a line of text that consists solely of a bracket or brace.

Related Documents

For more information, see the following documents in the Oracle Retail Fiscal Management Release 13.2.5 documentation set:

- *Oracle® Retail Merchandising System (RMS) Installation Guide*

- *Oracle® Retail Fiscal Management and Brazil Localization Installation Guide*

- *Oracle® Retail Merchandising System (RMS) Operations Guide*

- *Oracle® Retail POS Suite 13.3.5/Merchandising Operations Management 13.2.5 Implementation Guide*

- *Oracle® Retail Fiscal Management (ORFM) User Guide and Online Help*

- *Oracle® Retail Merchandising System (RMS) User Guide and Online Help*

- *Oracle® Retail Sales Audit (ReSA) User Guide and Online Help*

- *Oracle ® Retail Merchandising System (RMS) Data Model*

- *Oracle® Retail Fiscal Management (ORFM) Data Mode*l

- *Oracle ®Retail Merchandising System with Brazil Localization Release Notes*

- *Oracle® Retail Merchandising Batch Schedule*

- *Oracle® Retail Merchandising Implementation Guide*

Supplemental Documentation at My Oracle Support list:

- *Oracle® Retail Fiscal Management/RMS Brazil Localization - Multi-currency Operations - White Paper, Release 13.2.5 (ID 1493604.1)*

- *Oracle® Retail Fiscal Management/RMS Brazil Localization - Enterprise Business Suite (EBS) Integration - White Paper, Release 13.2.5 (ID 1493605.1)*

- *Oracle® Retail Fiscal Management/RMS Brazil Localization - Transfer Price Nota Fiscal - White Paper, Release 13.2.5 (ID 1493606.1)*

Customer Support

To contact Oracle Customer Support, access My Oracle Support at the following URL:

https://support.oracle.com

When contacting Customer Support, please provide the following:

- Product version and program/module name

- Functional and technical description of the problem (include business impact)

- Detailed step-by-step instructions to re-create

- Exact error message received

- Screen shots of each step you take

Review Patch Documentation

When you install the application for the first time, you install either a base release (for example, 13.1) or a later patch release (for example, 13.1.2). If you are installing the base release, additional patch, and bundled hot fix releases, read the documentation for all releases that have occurred since the base release before you begin installation. Documentation for patch and bundled hot fix releases can contain critical information

related to the base release, as well as information about code changes since the base release.

Oracle Retail Documentation on the Oracle Technology Network

Documentation is packaged with each Oracle Retail product release. Oracle Retail product documentation is also available on the following Web site:

http://www.oracle.com/technology/documentation/oracle_
retail.html

(Data Model documents are not available through Oracle Technology Network. These documents are packaged with released code, or you can obtain them through My Oracle Support.)

Documentation should be available on this Web site within a month after a product release.

Conventions

The following text conventions are used in this document:

Convention	Meaning
boldface	Boldface type indicates graphical user interface elements associated with an action, or terms defined in text or the glossary.
italic	Italic type indicates book titles, emphasis, or placeholder variables for which you supply particular values.
monospace	Monospace type indicates commands within a paragraph, URLs, code in examples, text that appears on the screen, or text that you enter.

1

Oracle Retail Fiscal Management Overview

This chapter provides an overview of Oracle Retail Fiscal Management (ORFM).

The following topics are covered in this chapter:

- What is Oracle Retail Fiscal Management?
- ORFM Functional Assumptions
- What is Nota Fiscal (NF)?
- What is Nota Fiscal Eletrônica (NF-e)?
- What is Retail Tax Integration Layer?
- What is TaxWeb Tax Rules?
- What is Sistema Público de Escrituração Digital (SPED)?

What is Oracle Retail Fiscal Management?

ORFM manages Nota Fiscal (NF) processing through integration with Oracle Retail Merchandising System (RMS), Oracle Retail Warehouse Management system (RWMS), and Oracle Retail Store Inventory Management (SIM). In Brazil, before shipping any inventory out of the warehouse or store, it is mandatory to generate a Nota Fiscal to accompany the inventory movement. Similarly, prior to receiving physical inventory in a warehouse or store, it is mandatory to match the NF.

ORFM is integrated with RMS, RWMS, and SIM. Since ORFM and RMS share the same instance and database schema, ORFM can look up the RMS database tables and vice versa. For example, a purchase order (PO) or transfer created in RMS is directly accessible to ORFM while a supplier, location, or item level fiscal attribute that is set in RMS can be accessed by ORFM through direct database lookup. ORFM also integrates with SIM and RWMS using the Oracle Retail Integration Bus (RIB).

ORFM foundation data is divided into following sections:

- System Setup
- Fiscal Configuration

System Setup

The ORFM System set up involves the following tasks:

- System Options
- Fiscal Number
- Tolerances

■ NF-e Configuration

System Options

Set up the ORFM system parameters, including the Default Utilization codes. The configuration settings control system behavior based on these values.

Navigate: From the **Main** menu, select **Fiscal Management > System Setup> System Options > Edit**. The System Options window appears.

Fiscal Number

Use the Fiscal Number to set up minimum and maximum fiscal numbers allowed at a particular store or warehouse location.

Navigate: From the **Main** menu, select **Fiscal Management > System Setup > System Setup > Fiscal Number > Edit**. The Fiscal Number window appears.

Tolerances

Specify the amount of cost and quantity variance allowed on a fiscal document when compared to actual requisition. You can add tolerances at the system level or at the supplier level.

Navigate: From the **Main** menu, select **Fiscal Management > System Setup > Tolerances > Edit**. The Tolerances window appears.

NF-e Configuration

Use this parameter to associate utilization codes with stores and warehouses.

Navigate: From the **Main** menu, select **Fiscal Management > System Setup > NF-e Configuration > Edit**. The NF-e Configuration window appears.

Requisition Types

This functionality supports NF creations corresponding to various transaction types. For instance, PO, PO for Rural Producer, Transfer, Repairing, InterCompany Transfers, RTV, Return NF, Return Merchandize Authorization, Inventory Adjustments (Stock Out), Direct NF, Free Form NF, and Sales NF.

Fiscal Configuration

Fiscal Configuration involves the following:

■ Fiscal Doc Type

■ Fiscal Utilization Code

Fiscal Doc Type

In Brazil, the movement of items need a document called fiscal document which is basically a financial document that details items, quantities, cost, retail, and taxes. Create and attach a Fiscal Doc Type (FM_DOC_T YPE_UTILZATION and FM_ FISCAL_DOC_TYPE) to the fiscal document.

Navigate: From the **Main** menu, select **Fiscal Management > Fiscal Configuration > Fiscal Document Type > Edit**. The Fiscal Document Type window appears.

Fiscal Utilization Code

Use the Fiscal Utilization Code to identify the type of fiscal operation that is being carried out by fiscal document.

Navigate: From the **Main** menu, select **Fiscal Management > Fiscal Configuration > Fiscal Utilization > Edit**. The Fiscal Utilization Setup window appears. Click **Add** and enter fields such as Utilization ID, Description, Requisition Type, Nature of Operation, and Mode. Click **Options** menu to enter details for Doc Types, Parameters, and Reason Codes. Click **OK**. The Fiscal Utilization Setup window closes. The nature of operation codes are exported by Tax Rules.

Fiscal Utilization - Document Type

Use this option to associate the relationship between Fiscal Document Type and Utilization Code.

Fiscal Utilization - Reason Code

Use this option to set up reason codes for inventory adjustments and repairing operations.

ORFM Functional Assumptions

For Brazil localized users, some of the RMS functionalities are not restricted by the system. Oracle Retail assumes that you will not use RMS specific functionality. For example, the RTV and Inventory Adjustment RMS features are not available for Brazil localized users.

ORFM/RMS Brazil localization has the following functional assumptions:

- When you install ORFM, the RMS system indicator, AUTO_RCV_STORE, available on the RMS Supplier screen must be set to 'N'.

- The Location-level security is not enabled in ORFM.

- For EDI NF, if there are observations related to recoverable Imposto sobre Circulação de Mercadorias e prestação de Serviços (ICMS) and Imposto sobre Circulação de Mercadorias e prestação de Serviços-Substituição Tributária (ICMS-ST), they should appear as different fields. The system cannot parse the observations.

- During discrepancy identification and resolution, the system always assumes the cost components on the Nota Fiscal are correct.

- During discrepancy identification, the system compares discounted cost on the purchase order and discounted cost on the NF. Unit cost (non-discounted) and the discount values are not compared separately.

- For triangulation purchase order flow, the NFs from both the suppliers should match in cost and quantity and should be entered in the system at the same time.

- The taxes on the main NF and the delivery NF (complementary NF for triangulation) are mutually exclusive.

- The system generates correction documents for both main and delivery suppliers.

- If the purchase NF has taxes such as Imposto sobre Produtos Industrializados (IPI) and ICMS-ST that are added to the NF total, these taxes are embedded in the return to vendor (RTV) cost on the RTV NF.

- Freight or any other cost component is not reversed on an RTV NF.

- For triangulation purchase order flow, both the delivery and main supplier should be known when the purchase order is created.

- If the stock being returned from a location was received at that location through a transfer, the system cannot track the last purchase NF and hence the RTV happens at Weighted Average Cost (WAC).

What is Nota Fiscal (NF)?

In Brazil, all movements of products from one location to another, or from a supplier to the retailer's location and vice versa, must be accompanied by a fiscal document called Nota Fiscal (NF). This document contains all the information related to the transaction such as, items, quantities, costs and retail values. It can be compared to a bill of lading (BOL) because it has the items and quantities. NF also has financial information such as the cost, so it can be compared to the invoice too. In addition, this document has all the taxation and fiscal information, which make the NF a unique document with all the related information. The management and processing of this document is closely linked to the business process flows of receiving, shipping, and all types of transactions that affect the inventory.

Once NF is created, it is validated and submitted for receipt. Once items are received, receipt information flows to Oracle Retail Management System (RMS). NF is validated against this receipt information and approved if there are no discrepancies. Possible discrepancies include cost, quantity, and tax. If there is any discrepancy in receipt it needs to be validated as either system purchase order (PO) or NF. Inventory and transaction updates are done in RMS only after NF is approved.

What is Nota Fiscal Eletrônica (NF-e)?

Nota Fiscal Eletrônica (NF-e) is a Brazilian government project tasked with the objective of implementing a national model of electronic fiscal documentation to replace the current system of issuing the fiscal documents in paper. The virtual document will have juridical validity guaranteed by the digital signature of the issuer. It will simplify the fiscal obligations of the taxpayers and will allow the follow-up of the commercial operations by the tax authority.

The NF-e issuer will generate an electronic file with all NF information in a more detailed level than the regular NF. This file must be digitally signed to guarantee the integrity of the data and the authorship of the issuer. This electronic file that corresponds to the NF-e is transmitted by the internet to the Secretaria da Fazenda - Brazilian Tax Authority (SEFAZ) of the origin state of the issuer. The SEFAZ provides a pre-validation of the file and returns a receiving protocol (Authorization for Use), that will be necessary to the traffic of the goods.

To follow the goods, a graphic representation of the NF-e will be printed. The Documento Auxiliar da Nota Fiscal Eletrônica - Auxiliary Document of the Electronic Invoice (DANFE) will be printed in a common paper, one copy that will have highlighted the access key for consultation of the NF-e in the internet, and a bidimensional bar code which will facilitate the capture and confirmation of information of the NF-e by the fiscal units.

The DANFE is not a Nota Fiscal, and does not replace the NF. It is an auxiliary document for consultation of the NF-e. It has the access code of the NF-e, which allows its owner to confirm the real existence of the NF-e in the Receita Federal Brasileira - Brazilian Federal Tax Authority (RFB) environment or the SEFAZ web site.

What is Retail Tax Integration Layer?

The Retail Tax Integration Layer (RTIL) is implemented as a Java Enterprise Edition application that exposes a Retail Tax Data Model API for calculating tax and hosts the associated tax service provider adapter(s). This layer forms the conduit between the Oracle Retail applications and the tax service provider. The Retail Taxation Integration Layer is responsible for the assembling and disassembling of the vendor specific data model to retail tax data model based on the configuration. RTIL is envisioned to host vendor-specific connectors which can communicate to the external third-party services with various protocols such as Enterprise JavaBeans (EJB), Java Message Service (JMS), Simple Object Access Protocol (SOAP), Hypertext Transfer Protocol (HTTP), or native Plain Old Java Object (POJO) based on the configuration. The subscribing application is not aware of the tax service provider. RTIL acts as a bridge between the subscribing application and third-party tax service provider.

What is TaxWeb Tax Rules?

The TaxWeb Tax Rules aids retailers with multiple state operation, with a high level of complexity and large number of transactions, items and locations.

ORFM is integrated with the TaxWeb Tax Rules to address all of the Brazilian tax legislation with a high level of exception treatments. For all flows in ORFM that need to have tax calculations, TaxWeb Tax Rules verifies that all taxes are applied considering the input parameters.

The following processes in ORFM/RMS use the tax calculation integration:

- Inbound Nota Fiscal validation
- Outbound Nota Fiscal issuing
- PO tax breakdown
- Item creation
- Refresh tax
- Item fiscal reclassification

What is Sistema Público de Escrituração Digital (SPED)?

Sistema Público de Escrituração Digital (SPED) or Public System of DigitalBookkeeping is the result of several efforts from the Brazilian government to modernize and increase the level of control over the fiscal transactions for all companies. It is based on a digital file that is transmitted periodically to the government through the internet. Similar to the NF-e, the file is digitally signed through specific programs that validate its format and content.

The strategy adopted to address this requirement was to keep the transaction features in ORFM and the interface to the Fiscal Authority in Oracle's fiscal partners.

To support this strategy, views and tables make available all information to the fiscal partner based on the fiscal movements.

2

Understanding ORFM Implementation Checklists

This chapter discusses the prerequisites required for implementing the fiscal module of RMS. It also provides a checklist for standard implementation and for legacy systems.

The following topics are covered in this chapter:

- Prerequisites
- Implementation Checklist for Legacy Systems
- Implementation Checklist for Standard Systems

Prerequisites

Ensure the following prerequisites are met before you use the ORFM application:

- Make sure RMS, ORFM, RTIL, and TaxWeb Tax Rules are installed successfully.
- RIB Adapters (formerly e-ways) – In case ORFM is integrated with SIM and RWMS, verify that the RIB adapters are up and running. RIB adapters ensure interface of schedule and shipment details between ORFM and RWMS/SIM. For more information, see the Oracle Retail Integration Bus documentation set, including Integration Guide.
- Localization Flex Attributes setup (LFAS) - Set up the localization flexible attributes for a set of books, transfer entity, country, store, warehouse, supplier, outside locations, external finishers, and for the item-country combination. Localization Flex Attributes facilitate linking of fiscal attributes to respective entities. For more information, see the *Oracle Retail Fiscal Management and Brazil Localization Installation Guide*.

Implementation Checklist for Legacy Systems

Use the following checklist before you implement ORFM in a legacy environment:

1. Run the data conversion script to load the data from the legacy system. For more information, see the *Oracle Retail Merchandising Data Conversion Operations Guide*.

2. Ensure that the following are populated with correct values before making the TaxWeb Tax Rules calls. The script `transform_br_future_cost.ksh` populates the `future_cost` table if there is no record in future cost table.

    - Future_cost table

- Currency_rates table are populated with the EFFECTIVE_DATE prior to the transaction date

3. Refresh the mv_currency_conversion_rates materialized view.

4. Run the scripts manually in the following order as shown in Table 2–1.

Table 2–1   Shell Scripts

Script	Function
seed_extax_setup_cost.ksh	This module reviews all the valid item/supplier/origin country/location combinations in RMS. It determines the unique fiscal attribute groups that cover all the item/supplier/origin country/location combinations and puts the groups on a stage table (l10n_br_extax_stg_cost).
extax_process_cost.ksh	This module picks up the staged groups placed on l10n_br_extax_stg_cost by seed_extax_setup_cost.ksh and calls the external tax provider with them. The results of these calls are placed on the l10n_br_extax_res_cost and l10n_br_extax_res_cost_det tables.
seed_extax_finish_cost.ksh	This module picks up the group level tax call results that extax_process_cost.ksh placed on l10n_br_extax_res_cost and l10n_br_extax_res_cost_det and explodes them back to the item/supplier/origin country/location level. It then uses the item/supplier/origin country/location level information to write tax information to ITEM_SUPP_COUNTRY, ITEM_SUPP_COUNTRY_LOC, FUTURE_COST, ITEM_COST_HEAD, and ITEM_COST_DETAIL.
seed_extax_setup_retail.ksh	This module reviews all the valid item/location combinations in RMS. It determines the unique fiscal attribute groups that cover all the item/locations combinations and puts the groups on a stage table (l10n_br_extax_stg_retail).
seed_extax_process_retail.ksh	This module picks up the staged groups placed on l10n_br_extax_stg_retail by seed_extax_setup_retail.ksh and calls the external tax provider with them. The results of these calls are placed on the l10n_br_extax_res_retail and l10n_br_extax_res_retail_det tables.
seed_extax_finish_retail.ksh	This module picks up the group level tax call results that seed_extax_process_retail.ksh placed on l10n_br_extax_res_retail and l10n_br_extax_res_retail_det and explodes them back to the item/location level. It then uses the item/location level information to write tax information to GTAX_ITEM_ROLLUP and POS_MODS_TAX_INFO.
seed_extax_future_cost.ksh	This module gets a list of fiscal attributes that have tax law changes scheduled against them. It then uses this information to determine what item/supplier/origin country/locations will be affected by those scheduled tax law changes. These item/supplier/origin country/location combinations are then used to create new cost changes on the date of their tax law changes. The normal cost change processes will then take care of updating RMS with the new tax law information when necessary.
seed_extax_future_setup_retail.ksh	This module gets a list of fiscal attributes that have tax law changes scheduled against them. It then uses this information to determine what item/locations will be affected by those scheduled tax law changes. It determines the unique fiscal attribute groups that cover those item/locations combinations and puts the groups on a stage table (l10n_br_extax_stg_retail).

Table 2–1   Shell Scripts

Script	Function
seed_extax_future_process_ retail.ksh	This module picks up the staged groups placed on l10n_br_ extax_stg_retail by l10n_br_extax_stg_retail.ksh and calls the external tax provider with them. The results of these calls are placed on the l10n_br_extax_res_retail and l10n_br_extax_res_ retail_det tables.
seed_extax_future_finish_ retail.ksh	This module picks up the group level tax call results that seed_ extax_future_process_retail.ksh placed on l10n_br_extax_res_ retail and l10n_br_extax_res_retail_det and explodes them back to the item/location level. It then uses the item/location level information to write tax information to GTAX_ITEM_ROLLUP and POS_MODS_TAX_INFO.
transform_br_country_ attrib_final.ksh	This script updates the columns default_po_cost, default_deal_ cost, and default_cost_comp_cost to NIC and updates the column item_cost_tax_incl_ind to Y for the country BR in the country_attrib table.
transform_br_upd_nic_ cost_type.ksh	This script updates the unit cost to negotiated item cost (NIC) for the tables item_supp_country_loc and item_supp_country. This script also updates the default_costing_type column of the future cost table to NIC.

Implementation Checklist for Standard Systems

Use the following checklist to implement ORFM in a standard environment:

1. RMS Security setup for ORFM - Finalize ORFM user roles. In ORFM, you can control access only at the menu level. ORFM uses the RMS security infrastructure to restrict access. For instance, one role could be set up with access rights to edit or view NF while another role can be set up to resolve any discrepancy on NF. For more information on setting up RMS security, see the *Oracle Retail Merchandising System Operations Guide*.

2. RMS System setup - Verify the following in RMS before using the ORFM application:

    – RMS system options Base Country is set as BR for Brazil.

    – RMS system option Currency Code is set as BRL.

    – RMS system options VAT indicator is set as Y and the default_tax_type is set as GTAX to enable tax functionality.

    – External Invoice Matching indicator is set to Y.

    – CREDIT_MEMO_LEVEL – Ensure that the CREDIT_MEMO_LEVEL in the system options table is set at location level 'L'. This system option indicates at what level the invoicing will happen for deals.

    – RMS system options Allow Auto Shipping/Receiving at Store is not selected.

    – All tax codes in the RMS VAT_CODES table are exactly the same as the tax codes set up in TaxWeb Tax Rules. This enables better tax integration with merchandising system.

    For more information, refer to the *Oracle Retail RMS-ReSA User Guide for Brazil Localization*.

3.   Verify the following setup in RMS:

- The localization_ind indicator in the COUNTRY_ATTRIB table is set to Y for Brazil.

- The default location is set up for Brazil using the COUNTRY_ATTRIB table. The default location is set up after you create the first location and is most likely to be set as the location for which most of the purchases occur.

- In the COUNTRY_ATTRIB table, ensure the ITEM_COST_TAX_INCL_IND parameter is set to Y with the DEFAULT_DEAL_COST and the DEFAULT_ COST_COMP_COST DEFAULT_PO_COST parameter to (NIC).

- The Flex field values are set to country BR. For the country BR, the fiscal attribute value should be 105.

- The Flex field values for all entities (Store, Warehouse, Supplier/Site, Partner, Outside Location, Transfer Entity, Set of Books) are set up appropriately.

4. **RTIL URL and Server Configure** – Capture valid RTIL URL details based on where it is deployed. Update the RMS RETAIL_SERVICE_REPORT_URL table with the URL for RS_CODE of RTIL. Supply a timeout value (the unit is in milliseconds) in the TIMEOUT column. You can arrive at this value in conjunction with the RIB timeouts configured at the customer premise. This value should be less than the configured RIB timeout.

5. **RIB Adapters setup** – Verify that the Schedule Submission adapter (ORFM Specific) and SO Status adapter (for transfers from RWMS or SIM) are up and running. This ensures that ORFM is successfully integrated with the external warehouse and store management systems.

6. **Fiscal Information from TaxWeb Tax Rules** – Execute the fiscal download batch to get the various fiscal codes from TaxWeb Tax Rules. The fiscal attributes provided by TaxWeb Tax Rules are associated with master entities such as locations and suppliers in RMS. It is also used during transaction processing in ORFM. One of the fiscal codes is Nomenclatura Comum do Mercosul (NCM). Verify that following fiscal attributes are available in the system:

   - NCM_CODES

   - NCM_CHAR_CODES

   - NCM_PAUTA_CODES

   - NCM_IPI_CODES

   - CNAE_CODES

   - SERVICE CODES

   - FEDERAL SERVICE CODES

   - NOP Codes

   Note: Fiscal download accepts any one of the parameters for instance, NCM, NCMCHAR and so on. Or, you can choose the parameter as 'ALL'.

   ---

7. **Document Type setup** – Set up the Document Type information in ORFM. The document types represent the fiscal codes defined by fiscal authorities in Brazil for various documents. You have to manually set up predefined document types and associate them to one or more utilizations. ORFM requires mandatory association

of a document type with an NF. For certain features, ORFM uses predefined document types. For example, NF-e can only be associated with document type 55.

8. Utilization Codes setup – Set up the utilization codes for different transaction types in ORFM. Utilization codes control the NF processing logic. The utilization codes have to be captured for every inventory transaction. The utilization code parameters control the application behavior. You need the NOP codes for setting up utilization codes.

Table 2–2 Utilization Code Parameters

Utilization Parameter	Description
Complementary NF	Indicates the utilization is of type Complementary NF. When you select the Complementary NF check box, the Complementary NF Type list is enabled.
Complimentary NF Type	Choose the Complementary NF type from the Complementary NF Type list: ■ Triangulation ■ Non-Merchandise Cost ■ Cost/Tax Correction
ICMS-ST Recovery	Controls whether the particular transaction is marked for Imposto sobre Circulação de Mercadorias e prestação de Serviços (ICMS)-ST tax recovery or not.
Item Utilization	The Item Utilization is sent to Tax Rules when populating the Freight Object. Choose the valid values from the following Utilizations LOV: ■ Commercialization (C) ■ Industrialisation (I)
Automatic NF Approval	Controls whether or not the outbound fiscal documents are validated, matched and approved automatically. This is relevant only for outbound NFs.
Allow Receiving	Select this check box to enable the schedule submission message flow to RWMS and SIM. In case of complementary NFs, where no receiving happens, this check box is disabled. This indicator is relevant for inbound NFs only.
Choose NF	Select this check box to choose a reference fiscal document. This is relevant for returns to vendor (RTV) related utilizations only.

> **Note:** Both Complementary NF and Complementary NF Type check boxes are not editable if you use the utilization code in an NF.

9. ORFM System Configuration – Set the ORFM system parameters in fm_system_options table. These system parameters ensure alignment of ORFM features with the customer's business requirements.

The following are a few important system parameters (categorized under respective functionality) required to be set up appropriately:

– Tolerance – Certain parameters have to be set around usage of tolerances in ORFM.

Table 2–3   System Options for Tolerance

System Option	Description
TOLERANCES_MANDATORY	Indicates if tolerances are applied.
TOLERANCES_PRIORITY	Indicates if the tolerances priority is applied - All, Supplier, Location, Company.
CALC_TOL_TYPE	Indicates the calculation tolerance type (P/V).
CALC_TOL_VALUE	Indicates the calculation tolerance value.

- Discrepancy Resolution – The discipline around discrepancy resolution has to be set up in system options. This enables system to resolve any PO receipt discrepancy in the favor of system or NF or lets you decide between the two.

Table 2–4   System Options for Discrepancy Resolution

Variable	Variable_Type	Description	Valid Values
COST_RESOLUTION_RULE	CHAR	Default resolution action for cost descrepancies	RECONCILE,NF, SYS
QTY_RESOLUTION_RULE	CHAR	Default resolution action for quantity descrepancies	RECONCILE,NF, SYS
TAX_RESOLUTION_RULE	CHAR	Default resolution action for tax descrepancies	RECONCILE,NF, SYS

- Electronic Data Interchange (EDI) – It is a mechanism through which data is exchanged between two systems or parties. In retail, it is used to exchange data between a retailer and a vendor or third party. Certain system options need to be set up around EDI and supporting partners.

Table 2–5   System Options for EDI

System Option	Description
EDI_DEF_PTNR_ID	Indicates the default partner ID when importing EDI documents.
EDI_DEF_PTNR_TYPE	Indicates the default partner type when importing EDI documents.
DEFAULT_FREIGHT_TYPE	Indicates the default freight type the process uses to create an EDI document based on external systems.

- History and Purge Days – Set the system options for the number of days of data to be held or purged in the system.

Table 2–6   System Options for Purging Operations

System Option	Description
HISTORY_DAYS_ COMPLETED_NON_POs	Indicates the number of days for completed documents other than POs.
HISTORY_DAYS_ COMPLETED_PO	Indicates the number of days for the completed POs.
HISTORY_DAYS_ DELETED_DOCS	Indicates the number of days for the deleted NF documents.
HISTORY_DAYS_HIST_ TABLES	Indicates the number of days the history table can hold the records.
HISTORY_DAYS_ WORKSHEET_STATUS	Indicates the number of days the NF can be in worksheet status.
PURGE_DAYS	Indicates the days before records are purged from the tables.

– Default Utilization Codes – Default utilization codes cannot be seeded as they are user defined. Set up these codes manually for certain transactions. This enables association of a mandatory or most probable utilization code to a transaction.

Table 2–7   System Options for Utilization Codes

System Option	Description
DEFAULT_INBOUND_IC_UTIL_ID	Default Utilization ID for inbound IC transfers
DEFAULT_INBOUND_REP_UTIL_ID	Default Utilization ID for inbound repairing transfers
DEFAULT_INBOUND_TSF_UTIL_ID	Default Utilization ID for inbound transfers
DEFAULT_OUTBOUND_IC_UTIL_ID	Default Utilization ID for outbound IC transfers
DEFAULT_OUTBOUND_REP_UTIL_ ID	Default Utilization ID for outbound repairing transfers
DEFAULT_OUTBOUND_TSF_UTIL_ ID	Default Utilization ID for outbound transfers
DEFAULT_PO_TYPE	Default PO Type when null in RMS (fiscal utilization code in ORFM)
DEFAULT_RMA_UTIL_ID	Default Utilization ID for Return Merchandise Authorization (RMA)
DEFAULT_RNF_UTILIZATION_ID	Default Utilization ID for Return NF
DEFAULT_RTV_UTIL_ID	Default Utilization ID for RTVs
DEFAULT_RURAL_PROD_ UTILIZATION	Default Utilization Id for Rural Producer NF
DEAL_DEFAULT_UTIL	Default Utilization ID for Deals
DEAL_DEF_CMPLX_TYPE	Default Document Type for Complex Deals
DEAL_DEF_FXMRH_TYPE	Default Document Type for Merchandise Fixed Deals
DEAL_DEF_FXNMRH_TYPE	Default Document Type for Non-merchandise Fixed Deals
DEFAULT_OUTBOUND_CUST_ UTIL_ID	Default Utilization ID for outbound Customer Order Transfers

Table 2–7 (Cont.) System Options for Utilization Codes

System Option	Description
DEFAULT_OUTBOUND_CUST_DOC_TYPE	Default Document Type for outbound Customer Order Transfers
DEFAULT_REF_MESSAGE	This NF number %s1 is referenced to the original NF %s2 issued on %s3 respectively
DEBIT_MEMO_TRAN_CODES	Tran Codes which have to be posted as look up code CREDIT in AP header staging
HISTORY_DAYS_PROCESSED_EDI	Indicates the number of days the EDI NFs can exits in the EDI tables before getting purged

> **Note:** You need to create the Utilization IDs before setting up the system options.

– Default Document Types – Default document type has to be associated with each transaction, this enables association of a mandatory or most probable document type to a transaction.

Table 2–8 System Options for Document Types

System Option	Description
DEFAULT_DOCUMENT_TYPE	Default document type used by the process to create an EDI document based on external systems
DEFAULT_INBOUND_IC_DOC_TYPE	Default Document Type for Inbound IC Transfers
DEFAULT_INBOUND_REP_DOC_TYPE	Default Document Type for Inbound Repairing Transfers
DEFAULT_INBOUND_TSF_DOC_TYPE	Default Document Type for Inbound Transfers
DEFAULT_NFE_DOC_TYPE	Default Document Type for NF-e
DEFAULT_OUTBOUND_IC_DOC_TYPE	Default Document Type for Outbound IC Transfers
DEFAULT_OUTBOUND_REP_DOC_TYPE	Default Document Type for Outbound Repairing Transfers
DEFAULT_OUTBOUND_TSF_DOC_TYPE	Default Document Type for Outbound Transfers
DEFAULT_RMA_DOC_TYPE	Default Document Type for RMA
DEFAULT_RTV_DOC_TYPE	Default Document Type for RTVs
DEFAULT_RURAL_PROD_DOC_TYPE	Default Document Type for Rural Producer
DEFAULT_STOCK_DOC_TYPE	Default Document Type for Inventory Adjustments
DOC_TYPE_PO	Default fiscal doc type code (e.g. 1 - Nota Fiscal - Modelo 1, 1A)

> **Note:** You need to create the Document Type before setting up the system options.

– ORFMi/Days – Define the system options to indicate the minimum and maximum number of days of acceptable variance from system date. This has to be the time frame within which the fiscal document related dates can be defined for an inbound or outbound shipment.

Table 2–9 System Options for Days

System Option	Description
ORFMI_MAX_DAYS	Number of days minus the ORFMi date (Inbound)
ORFMI_MIN_DAYS	Number of days minus the ORFMi date (Inbound)
ORFMO_MAX_DAYS	Number of days minus the ORFMo date (Outbound)
ORFMO_MIN_DAYS	Number of days minus the ORFMo date (Outbound)

– Field Format Validation – Define the system options data field formats as per your requirements.

Table 2–10 System Options for Field Format

System Option	Description
HOUR_FORMAT	Indicates the format to represent fields of hour. Use 'HH24:MI' or 'HH:MI AM'.
NUMBER_FORMAT	Indicates the Value Field format.
PERCENT_FORMAT	Indicates the Percentage Field format.
QTY_FORMAT	Indicates the Quantity Field format.

Table 2–11 System Options for Transfer Price

System Option	Description
TSF_PURCHASE_PRICE	Indicates whether the cost on the transfer NF generated is WAC 'N' or last purchase price 'Y'
INCL_DISC_EXP	It is used to specify inclusion (TRUE) or exclusion (FALSE) of expenses and discounts in the last purchase price

– Other Options – A few other important system options affecting system's behavior are provided in Table 2–12.

Table 2–12 Other System Options

System Option	Description
CHECK_APPROVED_ DOCS	Indicates a variable value to check whether it is mandatory to approve all fiscal documents before approving the schedule.
DEFAULT_COUNTRY	Indicates the default supplier country.
DEFAULT_CURRENCY	Indicates the default currency code.
ORIGIN_CST_IND	Indicates the default Origin Code and CST Type for ICMS tax situation.
RECOVERABLE_TAX_CST	Indicates the value related to the selected CST.

Table 2–12   (Cont.) Other System Options

System Option	Description
DEFAULT_NOP	Indicates the default Nature of Operation.
CREDIT_MEMO_LEVEL	Indicates the level at which invoicing happens for deals.

10. Tolerance setup – Set up tolerances at cost, quantity (percentage or value set for system/supplier) and calculation level (for rounding or truncation) in FM_ TOLERANCES table. During fiscal processing, the entry NF values are compared with the PO details for variances; tolerances facilitate identification of only those discrepancies which fall over and above applicable tolerance levels.

11. NF-e configuration – Set up locations that need to issue NF-e in FM_NFE_ CONFIG_LOC table to ensure that the NF proceeds to SEFAZ for getting authorization key. ORFM enables generation of NF-e by location and utilization code. Location has to be identified for NF-e issuance for a particular transaction, and for that location NF-e parameters have to be configured. The transactions for which NF-e can be issued such as transfers, are initiated in RWMS. The utilization associated with each location must be the same utilization set in system variables as the default utilization for outbound transactions such as transfers, RTVs, intercompany transfers. The default utilization code is further used by the warehouse management system to generate the NF. This behavior is not controlled in the system and has to be defined by you.

12. ORFM Tax setup – Enter the tax codes as in TaxWeb Tax Rules in the VAT Codes Maintenance form of RMS. Tax details have to be inserted manually in the ORFM FM_TAX_CODES table, because there is no user interface in the ORFM application for inserting the tax details. The value added tax (VAT) codes in ORFM must be similar to the data set up in VAT codes in RMS. Also, ensure that the tax code/VAT codes set up in RMS and ORFM are as available in TaxWeb Tax Rules.

13. Fiscal Document Numbering Sequence setup – Set up the fiscal document number sequencing details through the Location Fiscal Number form in ORFM. Fiscal document number sequence set up enables you to capture details on sequence series for each supported requisition type. During NF generation, ORFM generates and associates a unique sequence number to the NF based on its requisition type and related sequence series.

14. Brazil Localization setup – Verify that the L10N_PKG_CONFIG table for decoupling contains entries for Brazil before any transactions happen within ORFM. These entries enable Brazil localization related features.

15. SPED Configuration setup – For SPED, you have to interface with third-party system. You have to share the database, and the ports have to be opened in order to establish network connectivity. You can decide to either host the SPED interfacing application (Interdados) within their environment or host it in fiscal partner's environment. For security considerations, a separate schema has to be created that contains only synonyms to as many objects required by the fiscal partner to generate the SPED information. Only the 'select' privileges has to be granted on these synonyms. No insert/update/delete should be allowed. This integrated environment is accessed through DB Link by third party systems.

16. Ensure access to the following forms (accessed through RMS forms) are denied through Security settings for Brazil localized users when Brazil localization is ON:

    – Receiver Cost Adjustment (RECCTADJ)

    – Receiver Unit Adjustment (RECUTADJ)

- – Average Cost Adjustment (AVCSTADJ)
- – Inventory Adjustment by Item (INVADJSK)
- – Inventory Adjustment by Location (INVADJST)

> **Note:** Only RMS supports the multi-currency functionality.

3

Setting up RMS for ORFM and Brazil Localization

This chapter describes the processes you need to prepare RMS to use ORFM and Brazil localization.

Decoupling RMS and ORFM Batch

Identical processes that are performed differently in RMS and ORFM have been separated using a process called decoupling. This allows installation of RMS without installing any (or a minimal portion) of the ORFM modules. This means that direct reads and writes to the ORFM tables and views from RMS are avoided.

All processes involving localization are encapsulated within a localization wrapper package. The decoupling between RMS and ORFM is controlled in that wrapper package.

Dynamic Function Calls

ORFM and RMS utilize dynamic function calls to decouple. This method uses a configuration table to determine which function or procedure, either base or localized code, is dynamically executed.

Figure 3–1   Dynamically Executed Decoupling
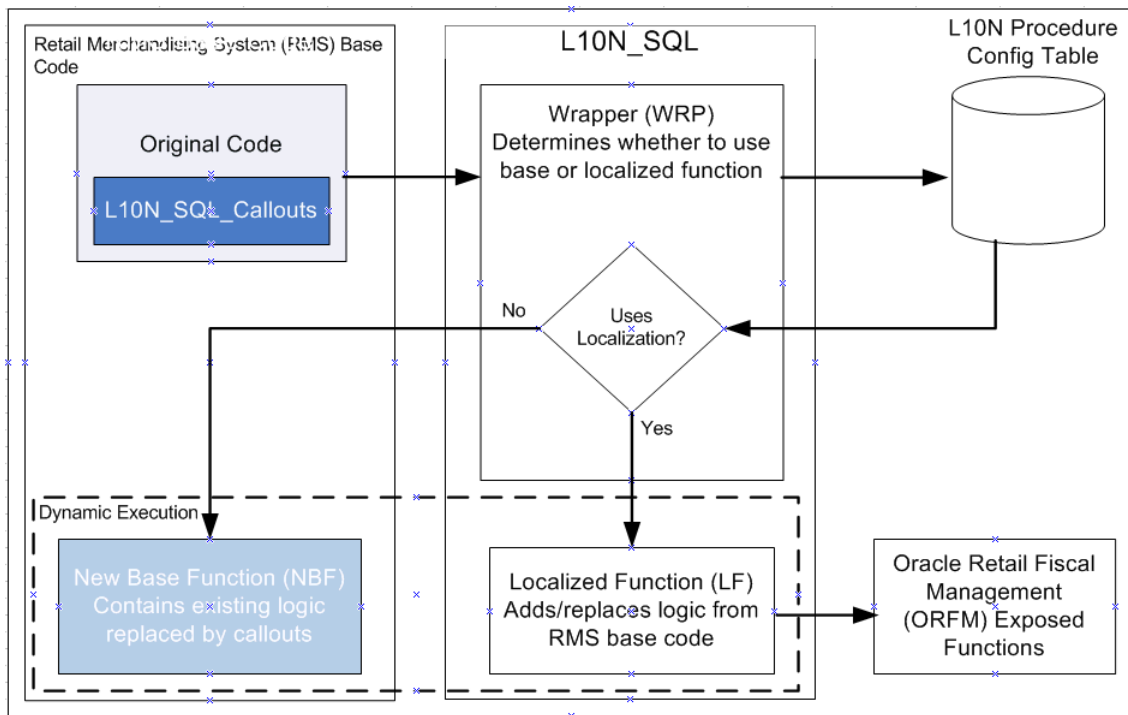
There are five components in the dynamic call approach:

1. **The original code –** This is the base code that does the callout to the localization layer.

2. **L10N_SQL wrapper (WRP)** – This function decides whether to use the base code or the localized version of the code.

3. **L10N configuration table** – This table contains the procedure name to be executed per localized instance.

4. **New base function (NBF) –** This function contains the existing logic from the original code replaced by callouts to the localization layer.

5. **Localized function (LF) –** This contains the necessary procedures to carry out the localization tasks. This may contain actual calls to ORFM functions.

PL/SQL Package Calls

Callouts are made in the original base code where localized codes are introduced. There can be more than one callout in a base function, and every callout points to the localization layer.

In scenarios in which code needs to be replaced, or the base code block is not applicable when localization is enabled, a New Base Function (NBF) exists to contain this code. The localized functionality is moved to a Localized Function (LF).

The localization layer contains the WRAPPER function (WRP) - EXEC_FUNCTION, which is called by the original base code. This wrapper uses a L10N configuration table to determine if the New Base Function (NBF) or a Localized Function (LF) needs to be executed. The call to these functions is done dynamically. The configuration table may contain different function references for different localizations.

Because the calls to the functions are dynamic and the parameters passed in and out may vary depending on the function called, the PL/SQL Type Inheritance approach is used to accomplish this.

PL/SQL Type Inheritance

In this approach, the in/out parameters to be passed to functions should be encapsulated in object types. The object types follow PL/SQL Type Inheritance (a supertype is created from which several subtypes will be derived). The supertype is a generic object that can be used across functions. Derived subtypes contain all the attributes of the parent type (or supertype). The subtypes can also contain additional attributes.

Localization Layer

The localization layer is made up of the following components:

- The L10N configuration table
- The L10N_SQL package
- Country-specific localization packages (L10N_<CN>_SQL)

Figure 3–2   Localization Layer
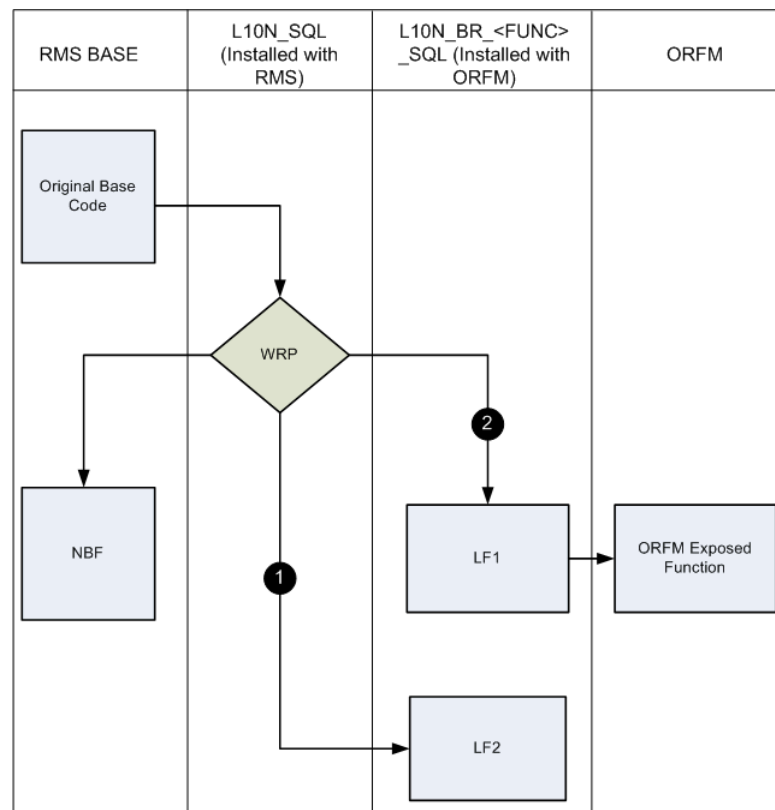
L10N_SQL contains only the wrapper and other functions, which are generic to localized countries.

L10N_BR_<FUNC>_SQL contains only the Brazil-specific functions under <FUNC> functionality area that are called by the L10N wrapper. These functions may (flow line

(2) in the diagram) or may not (flow line (1) in the diagram) call ORFM functions directly.

Accomplishing Decoupling

BASE INSTALL: RMS has a dependency on L10N_SQL because it calls the wrapper directly through a callout. When RMS is installed, an L10N configuration table is pre populated with references to the New Base Functions (NBF). Because these NBF functions have no dependency on ORFM, RMS does not need the Localized Functions (LF) in the base.

LOCALIZATION INSTALL: After ORFM and country-specific localization packs (L10N_BR_<FUNC>_SQL, L10N_PE_<FUNC>_SQL) are installed, the L10N configuration table is pre populated with the localized functions for the country (BR, PE). The wrapper now calls the functions from the country-specific packages.

> **Note:** Dependency is created only at run time because the functions are executed dynamically.

Decoupling Batch Programs

In a batch program, decoupling is accomplished in a way that, if a base and localized process must occur (forking), the base logic will reside in the NBF and the localized process will be in the LF. A wrapper is used by the main batch program to call the LF or NBF. Only one version of the function is executed in an instance of the function call.

To decouple the functions, there are separate localized libraries for LFs per country. NBFs are declared in the main batch program.

A utility library contains all the utility functions including the wrapper function. The utility function is used by the main batch program to find and retrieve the LF or NBF it has to execute. The LF pointers are organized based on the batch program that uses them, so one batch program can only use its set of LF pointers.

The NBF pointers are declared in the main batch program.

To make the parameters of the NBFs and LFs generic, a struct is used. This struct, which is called the parent struct, resides in an object library. This is the only parameter for the NBF and LFs. If additional fields are needed, the library must be modified to create a struct within the parent struct (the child struct). These structs must be populated first before passing it to the wrapper class to call the NBFs or LFs.

Setup Fiscal Download

Fiscal Download holds data such as the Nature of Operation (NOP), ncm_codes (Nomenclatura Comum do Mercosul classifies items according to Mercosul regulations), pauta_codes and NCM exception codes. This data is obtained from TaxWeb Tax Rules, and is maintained in RMS. This is mandatory information for Item and NF creation for the tax retrieval from TaxWeb Tax Rules.

No front-end navigation is available for this. The l10nbrfisdnld batch downloads the data. These fiscal attributes are utilized while making tax calls to TaxWeb.

> **Note:** Fiscal download accepts any one of the parameters for instance, NCM, NCMCHAR and so on. Or, choose the parameter as 'ALL'.

Setup Item Fiscal Attributes

Item fiscal attributes are mandatory for any item creation. These attributes are referenced when the item is used for any kind of transaction data creation like Purchase Orders and Transfers. Item attributes include NCM code, service ind (Y/N), NCM characteristic code, ex ipi, pauta code, service code, origin code, federal service code, state_of_manufacture, and pharma_list_type.

Navigate: From the **Main** menu, select **Items > Country > Fiscal Attributes**.

Setup Fiscal Attributes for other Entities

Fiscal attributes for other entities hold information such as address, company code, and tax contributor indicators. These are necessary for tax calls and are used by TaxWeb Tax Rules to apply appropriate rules and return the correct taxes.

Navigate: From the **Options** menu, select **Fiscal Attributes**. You can also navigate Fiscal attributes from the **Options** menu of Warehouses, Stores, Outside Locations, Partners, Supplier, Transfer Entity, and Set of Books.

For more information, see the *Oracle Retail Merchandising System User Guide.*

Setup Utilization

All merchandise that is being fiscally accepted using ORFM must have the fiscal utilization, which determines the type of the business operation that the fiscal document supports.

The fiscal utilization determines the appropriate taxes involved by the retail operation, the impact on stocks and costs, and the type of information to be sent to other systems.

Utilization is associated with a Requisition Type and a Nature of Operation. This identifies the type of transaction for TaxWeb Tax Rules to calculate appropriate taxes.

Utilization also can be associated with a particular Document Type. Different attributes are available to be configured for utilization which gives more granular information on the kind of document and action to be applied.

The following are the various parameters you can configure for an utilization code:

- Complementary NF
- Complementary NF for Triangulation
- Complementary NF for Cost/Tax Correction
- Complementary NF for Freight
- ICMS-ST recovery
- Automatic NF Approval
- Allow Receiving
- Choose NF
- Item Utilization

Navigate: From the **Main** menu, select **Fiscal Management > Fiscal Configuration > Fiscal Utilization > Edit**. The Fiscal Utilization Setup window opens.

Setup Document Types

You must set up at least one fiscal document type prior to using ORFM. Each Fiscal Document Type needs to be associated with a fiscal utilization. Multiple Utilization identifiers can be associated to one document type.

Navigate: From the **Main** menu, select **Fiscal Management > Fiscal Configuration > Fiscal Document Types > Edit**. The Fiscal Document Type window opens.

Setup System Options

The `fm_system_options` table is the control table where several parameters are stored. The system options table provides various settings such as environment type (Brazil/Non-Brazil), supplier site is on or off, and if multiple set of books are in use.

System options can be setup using the user interface or supplied through direct entries using the database . Usually system options are setup during RMS installation as part of seed data. Use the RMS user interface to update system options post installation.

You will need to setup ORFM system parameters, including the Default Utilization codes. The configuration settings control system behavior based on the values entered.

Navigate: From the **Main** menu, select **Fiscal Management > System Setup > System Options > Edit**. The System Options window opens.

Setup Reason Codes

You must set up reason codes for overage and damaged in the RMS reason code master table (inv_adj_reason) prior to using ORFM. These reason codes are interfaced to ORFM from RWMS and SIM. If these reason codes are not set up correctly, the integration to ORFM for receipt verification fails.

Setup Partner

To use ORFM, the default partner and partner type in fm_system_options must be setup in RMS.

Set Primary Country

To use ORFM, the Primary Country must be set to **BR**. For more information on how this can be accomplished during RMS installation, see the *Oracle Retail Merchandising System Installation Guide*.

Set VAT Indicator

To use ORFM, VAT_IND must be set to **Y**. For more information on how this can be accomplished during RMS installation, see the *Oracle Retail Merchandising System Installation Guide*.

Set Default Tax Type

To use ORFM, the Default Tax Type must be set to **GTAX**. For more information on how this can be accomplished during RMS installation, see the *Oracle Retail Merchandising System Installation Guide*.

Verify Localized Indicator

The Localized indicator check box indicates whether the Brazil patch is installed. It is auto-checked by the system if the Brazil patch is installed.

Use the following procedure to verify the Localized Indicator:

1. From the **RMS Start** menu, select **Control > Setup**. In the Content of Setup window, select **Country > Edit**. The Country Maintenance window appears.

Figure 3–3   Country Maintenance Window
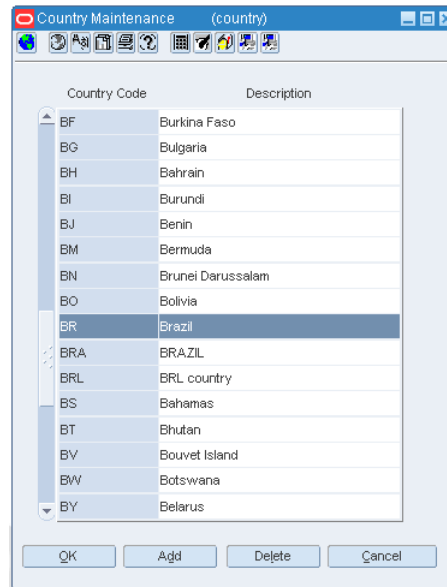
1. Select **BR** and click the **Options** menu.

    From the **Options** menu, select **Attributes**. The Country Attributes window opens.

Figure 3–4   Country Attributes Window
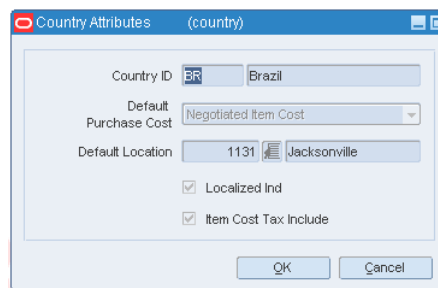
2. Verify that the **Localized Ind** check box is selected and click **OK**. You are returned to the Country Maintenance window.

3. Click **OK** to save your changes and close the window.

Setup Tax Codes

Tax codes must be set up in the FM_TAX_CODES table. This table holds entries for all possible tax codes that can appear on a Nota Fiscal which would be applicable on any

given transaction. It also holds the matching_ind (Y/N) identifier which drives the tax discrepancy identification functionality of ORFM.

This table can only be populated by entering values directly from the back end.

Setup Flex Field Attribute

The fiscal code of the country must be set up before using ORFM.

Navigate: From the Main menu, select Control > Setup > Country > Edit. The Country Maintenance window opens.

Figure 3–5   Country Maintenance Window
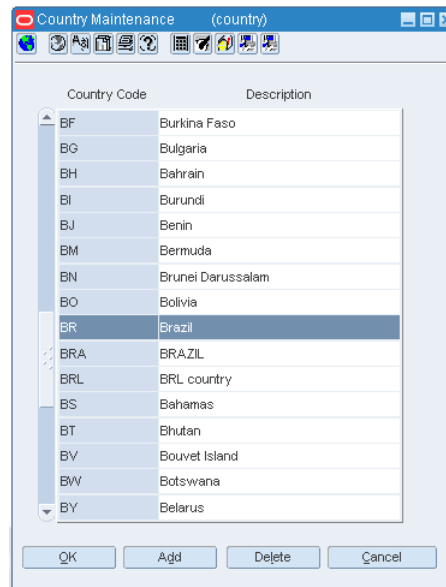
1. Select **BR** and click the **Options** menu.

2. From the **Options** menu, select **Fiscal Attributes**. The Localization Flexible Attributes window opens.

Figure 3–6   Localization Flexible Attributes Window

3. Set the Fiscal Code to 105 and click **OK**. You are returned to the Country Maintenance window.

4. Click **OK** to save your changes and close the window.

Setup Fiscal Document Chunk Size

The L10N_TAX_OBJECT_CONFIG.THREAD_ITEM_LOC_COUNT column in The `L10N_BR_TAX_CALL_STAGE_RMS` table specifies the maximum number of item/locations that are contained in a Fiscal Document object. Location is defined as the 'From Entity' in the ORFM context. Logic in the ORFM integration package (L10N_BR_INT_SQL) uses this value to split fiscal data into logical unit of work packets or 'chunks' that are processed concurrently by RTIL.

Chunks as used in RTIL are further discussed in Chapter 7, "Integrating with TaxWeb Tax Rules – Retail Tax Integration Layer" in section Concurrent Processing in RTIL.

The value in the L10N_TAX_OBJECT_CONFIG.THREAD_ITEM_LOC_COUNT column defaults to 1000 through a script upon installation. Oracle Retail recommends that you set this value in conjunction with determining the configuration for RTIL's Maximum Thread Constraint and Capacity Constraint as they are intrinsically related.

Setup FM Tran Code

Tran codes in ORFM table `fm_tran_codes` are set up as a part of installation.

Setup Set Of Books

You must set up Set of Books in `FM_SOB_SETUP` table.

Setup Chart of Account

You must set up Chart of Accounts in `FM_COA_SETUP` table.

Setup GL Options

You must set up the entities that are dynamic in nature in `FM_GL_OPTIONS` table.

Dynamic Segments Setup for Chart of Accounts

You must set up dynamic segments for chart of accounts in `FM_DYNAMIC_SEGMENT_SETUP` table. You can also achieve this through the Dynamic Segment Configuration form.

Account Setup for Chart of Account

You must set up account setup for chart of account in `FM_ACCOUNT_SETUP` table.

Setup GL Cross Reference

You must set up GL cross reference in `FM_GL_CROSS_REF` table. You can also achieve this through the GL Cross reference form.

4

Integrating ORFM

ORFM is integrated with other Oracle Retail applications including RMS, RWMS, and SIM. Because RMS and ORFM share the same instance and same database schema, ORFM can look up the RMS database tables and vice versa. A purchase order (PO) or transfer created in RMS is directly accessible to ORFM. Supplier, location or item level fiscal attributes that are set in RMS can be accessed by ORFM through direct database look up.

ORFM integrates with SIM and RWMS through RIB (for real-time integration). If Brazil localization is enabled, SIM or RWMS should not be allowed to physically receive any inventory before fiscal receiving (inbound NF Processing) is completed in ORFM. After physical receiving in SIM or RWMS, the inventory in RMS should not be updated until fiscal receipt and physical receipt comparison is done in ORFM and the discrepancy (if any) is resolved as explained above.

If Brazil localization is enabled, RMS inventory is not updated immediately when any inventory is shipped out from RWMS or SIM. The inventory movement information should be sent to ORFM for the generation of outbound NF. After approval of the outbound NF, ORFM should send the inventory updates to RMS. For more information, see the Outbound Operations Process Flow diagram in the "Overall Solution Landscape" section.

In the subsections below, each transaction with a different integration flow (other than the base application) for Brazil is detailed.

This chapter covers the following topics:

- Purchase Order Receiving
- Transfer Shipment and Receiving
- Retail Merchandise Authorization
- Inventory Adjustment
- Finishers/Repairing Flow (Two-Legged Transfer)
- RIB-Based Integration
- HTTP-Based Integration
- Nota Fiscal – Receiving and Issuing
- ORFM and RWMS
- ORFM and SIM
- ORFM Integration with Financial Application

ORFM Integration

Purchase Order Receiving

Purchase orders are created in ORMS and communicated to Vendor; Vendor sends items back to retailer based on purchase order and generates invoice to retailer. Based on items received, invoice is adjusted by retailer and payment is made to the vendor. PO can also be created by ORSIM through direct store delivery functionality and communicated to ORMS.

In Brazil, receiving without a valid PO number is not allowed because fiscal receiving must precede physical receiving. Therefore, a direct store delivery without a PO cannot be received in SIM. A PO created in RMS is sent to RWMS and SIM through RIB as per the base process flow. The PO created flows to the TaxWeb Tax Rules through the Retail Taxation Integration Layer (RTIL), along with the fiscal attributes of the suppliers, receiving locations, and the items. The TaxWeb Tax Rules calculates the tax breakdown and sends it back to RMS through RTIL

Figure 4–1 Purchase Order Receiving
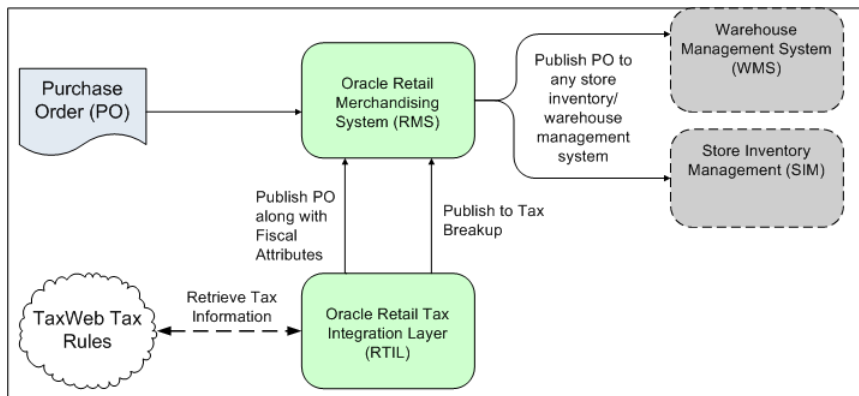
A purchase order goes through the following process and is published to RWMS or SIM:

1. The purchase order is created in RMS.

2. The purchase order, along with the PO's fiscal attributes, are published to RTIL.

3. The Brazil tax information is retrieved and appended to the purchase order and sent back to RTIL.

4. RTIL publishes the tax breakdown of the purchase order.

5. The purchase order is published to SIM or RWMS.

> **Note:** Steps 2, 3 and 4 depicted above are specific to Brazil localization. Steps 1 and 5 are the same as in base RMS.

ORFM has direct access to the PO created in RMS and the related fiscal details. This is required to validate and match the Nota Fiscal.

The following PO details are relevant to ORFM:

- PO header information

- Order supplier master data

- Order supplier site master data

- Order supplier site fiscal attributes

- Delivery supplier master data

- Delivery supplier site master data

- Delivery supplier site fiscal attributes

- Destination locations master data

- Destination location fiscal attributes

- PO detail information

- Non-merch cost details

- Item master data

- Item fiscal attributes

PO Receiving at a Warehouse

For a PO with a warehouse as the receiving location, the NF schedule status is Submitted. The message confirming that fiscal receiving is complete is published to RWMS through the Oracle Retail Integration Bus (RIB).

The RIB message contains the following information:

- Schedule number

- PO number

- Receiving physical warehouse number

- Distinct item IDs per PO

- Consolidated quantities per item-PO

- UOM

One Nota Fiscal may contain items pertaining to several POs, and one schedule may contain several Nota Fiscal documents. It is likely that the supplier ships the same item on a PO through multiple NFs, and these NFs are part of the same schedule. If so, ORFM consolidates the item quantity by PO number before publishing it to RWMS.

Figure 4–2   PO Receiving at a Warehouse
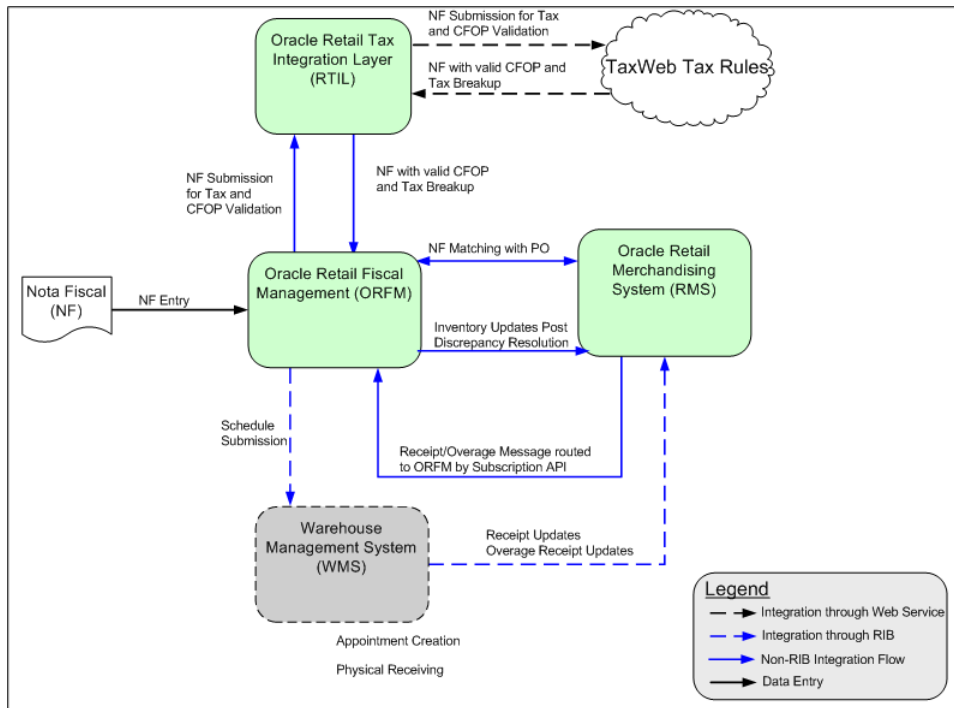
After the automatic appointment creation, the RWMS user can schedule the appointment and proceed with the physical receiving. If Brazil localization is enabled in RWMS, receiving an item that is not in the appointment or receiving any extra quantity than in the appointment is restricted. Any such overage should be received separately in RWMS as an overage receipt.

After completing the receipt, as the appointment is closed, the receipt and the overage receipt are published by RWMS through RIB. RMS consumes these messages and the Subscription API routes the messages to ORFM (in Brazil Environment). When receiving, the same receipt message is used that is used to update receipt information to RMS. Though not all fields in the receipt message are relevant to ORFM, ORFM still consumes the entire message and persists it until the RMS inventory is updated.

The Receipt Overage message is a RIB message from RWMS to ORFM.

The message contains the following information:

- Schedule Number
- ASN Number (Null for Warehouse receipts)
- Item Id
- Quantity Over-received
- UOM
- Reason Code

After ORFM consumes the Receipt and Receipt Overage messages, the discrepancy between physical and fiscal receiving is processed through the discrepancy resolution module. The correction letter and/or Return NF (for an overage receipt) are generated. No inventory updates happen in RMS for the overage receipt.

For the received items, the taxes are stripped from the unit cost to arrive at the tax-exclusive cost, additional cost components (if any) are added to calculate the actual landed cost, and the inventory and WAC updates are sent to RMS along with the NF number. Other than the cost and NF, all other information regarding the shipment is taken from the receipt message received from RWMS.

PO Receiving at a Store

For a PO with Receiving Location as a Store, if the NF schedule status is Submitted, the ASNs (ASNin) are generated and published to SIM. SIM then consumes this message to create the inbound ASN.

One NF can contain items pertaining to several POs, and one schedule can contain several NFs. It is likely that the supplier ships the same item on a PO through multiple NFs, and these NFs are part of the same schedule. If so, ORFM consolidates the item quantity by PO number before publishing it to SIM.

Figure 4–3   PO Receiving at Store
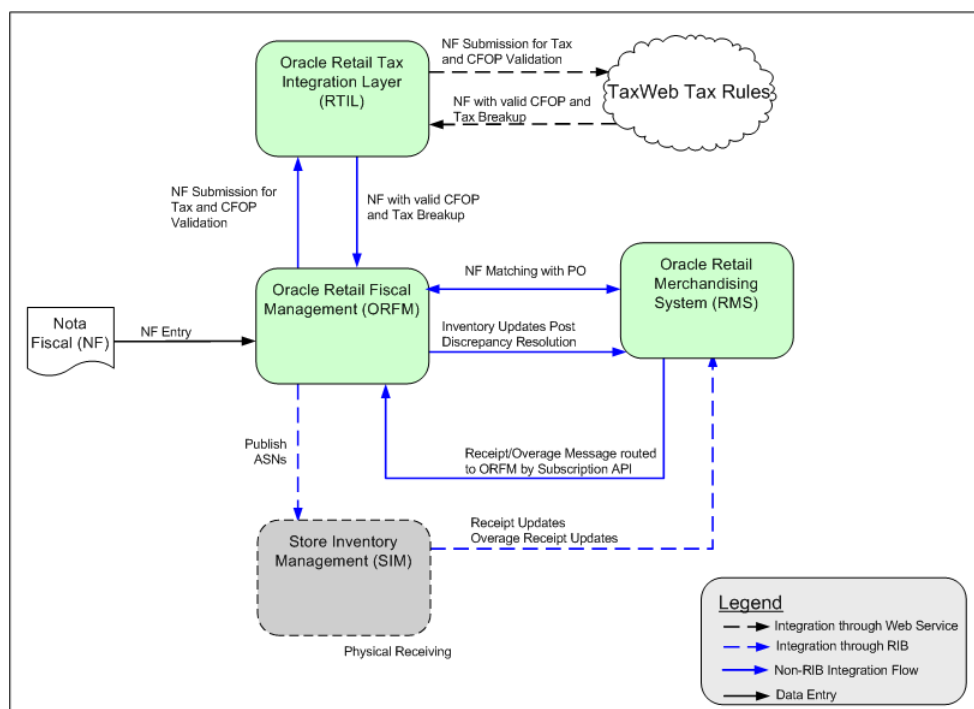
After the ASN message is received, the SIM user can proceed with the physical receiving. For Brazil localization, PO receiving without a valid inbound ASN is not permitted in SIM. In addition, receiving an item that is not in the ASN, or receiving any extra quantity than in the ASN, is restricted.

SIM can enforce this restriction in the following ways:

- The user can continue receiving inventory beyond the expected quantity on the NF, but the extra quantity will be removed from the transaction when confirming the delivery.

- The user is not allowed to receive any units above the expected quantity.

When receipt is completed for each ASN, the receipt and the overage receipt per ASN are published to RMS through RIB. For a receipt, the same receipt message is used that

is used to update receipt information to RMS. The RMS subscription API does not consume the receipt message for the Brazil environment, but it is diverted to ORFM. ORFM stores the information in a staging table until all the ASNs in a schedule are received. The schedule status is changed to Received only when all the ASNs in a schedule are received. Referential integrity between the schedule number and the ASN is maintained in ORFM. SIM has no visibility to the schedule number. Not all the fields in the receipt message are relevant in ORFM, but ORFM still needs to consume the entire message and persist it until RMS inventory is updated.

The Receipt Overage message is a RIB message from SIM to ORFM.

After ORFM receives the Receipt and Receipt Overage messages, the discrepancy between physical and fiscal receiving is processed through the discrepancy resolution module. As required, the correction letter and/or Return NF (for an overage receipt) are generated. No information about the overage receipt flows to RMS from ORFM. For over-receipt, no inventory updates happen in RMS.

For the received items, the taxes are stripped off from the unit cost to arrive at the tax-exclusive cost. Additional cost components (if any) are added to calculate the actual landed cost. ORFM then calls RMS to update inventory and WAC. Other than the cost and the NF number, all other information regarding the shipment is taken from the receipt message that SIM has published.

Transfer Shipment and Receiving

A transfer (inter-company as well as intra-company) or allocation created in RMS is sent to RWMS and SIM through RIB.

ORFM has direct access to the transfer and the related fiscal details.

Transfers

Transfers are created to move items from one location (source) to other (destination). Location could be store or warehouse.

> **Note:** The following section uses warehouse-to-warehouse transfers to illustrate the transfer process. Other transfer processes, including warehouse-to-store, and store-to-store are similar from an ORFM perspective.

Transfer Creation

If a warehouse-to-warehouse transfer (or allocation) is initiated in RMS, the transfer request flows to RWMS (Sending Location and the Receiving Location) through RIB.

Figure 4–4   Warehouse to Warehouse Transfer Creation
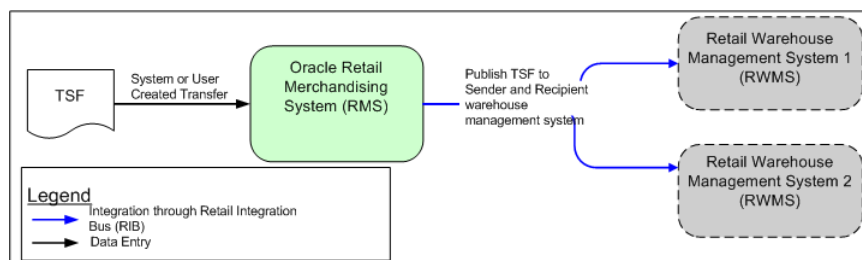
> **Note:** The sender warehouse and the recipient warehouse may run on the same instance or different instances of RWMS. If both warehouses run on the same instance of RWMS, RWMS-1 and RWMS-2 in the above image would be the same.

If a warehouse-to-warehouse transfer is initiated in RWMS, it flows to RMS and the other RWMS through RIB. If the RMS-initiated transfer is modified in RWMS-1, the modifications flow back to RMS and the other RWMS.

Figure 4–5   Warehouse to Warehouse Transfer
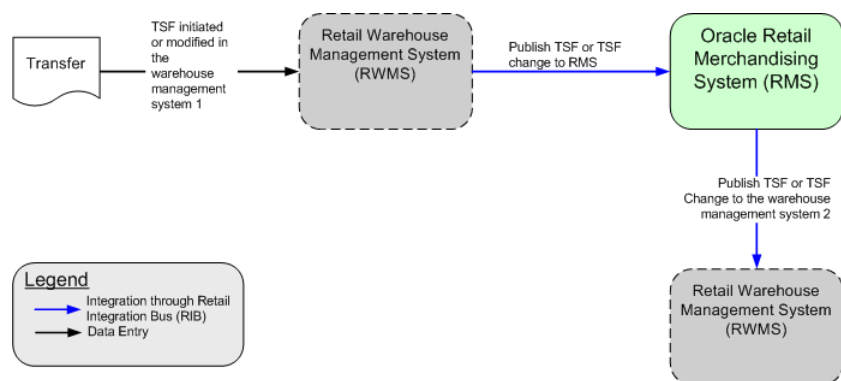
Transfer Shipment at Source

In Brazil, each inventory movement (shipment) should be accompanied by a fiscal document (NF). When the shipment goes out of the source warehouse, an NF is generated and approved. When RWMS ships the transfer out, ORFM is informed about the ASNout details so that the outbound NF can be generated. On approval of the outbound NF, the on-hand inventory of the source warehouse and in-transit inventory of the destination warehouse are updated in RMS.

The ASNout should be sent to the destination warehouse too, so that the destination warehouse generates the appointment and is prepared for the receiving.

Figure 4–6   Transfer Shipment at Source
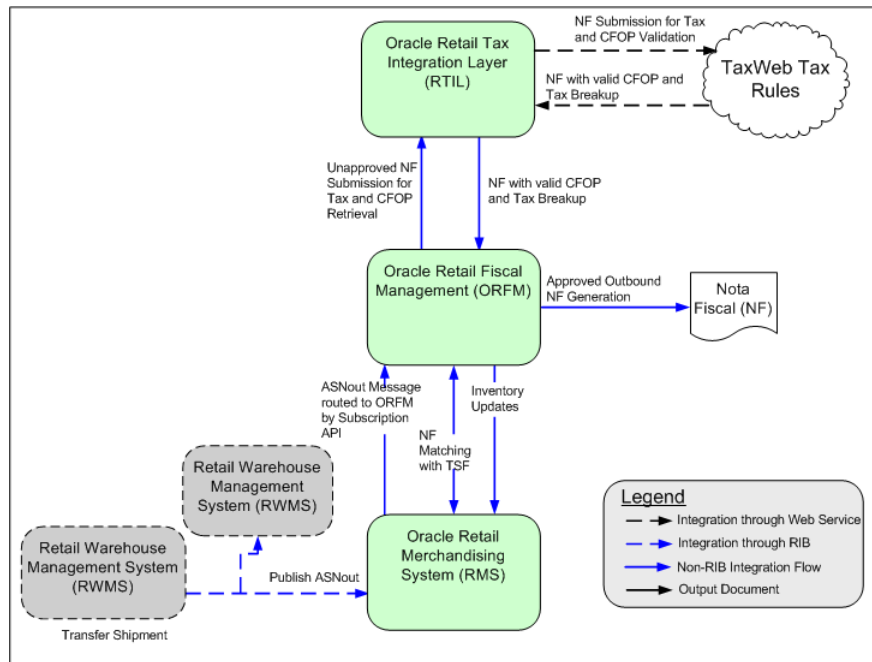
When the transfer is picked and shipped in RWMS-1 at the source warehouse, ASNout is published and is subscribed to by RMS and RWMS of the destination warehouse simultaneously. In the Brazil environment, RMS routes the ASN message to ORFM. On consuming the ASN, ORFM generates the outbound schedules and outbound Nota Fiscal documents using the ASNout information. The tax information and the CFOPs are published on the NFs. NFs are validated before approval. On NF approval, the ASNout information is used to update the RMS inventory subsequently. The on-hand inventory of the source warehouse is decreased and the in-transit inventory for the destination warehouse is increased.

Transfer Receiving at Destination

When the shipment arrives at the destination warehouse, the user is expected to create an inbound schedule using the outbound schedule. There should be an inbound NF for each outbound NF in the schedule, and a one-to-one relationship must exist between the outbound schedule and inbound schedule, as well as outbound NFs and inbound NFs. As the schedule is entered and validated (retrieve appropriate inbound CFOPs through TaxWeb Tax Rules), the schedule can be submitted to RWMS-2 for receiving. Because RWMS-2 already has the ASN information, it is enough that the message published to RWMS-2 contains just the schedule number and the ASN numbers grouped under the same schedule. To avoid introducing a new RIB message, the schedule is submitted with the entire ASN (referred to as secondary ASN). RWMS can ignore the other information in the message and can just refer to the ASN numbers. The schedule number can be null, as in the case of publishing an ASN to SIM.

RWMS should create one appointment per schedule and combine all the ASNs for that schedule under the same appointment. After scheduling the appointment, the RWMS user can proceed with the physical receiving. It is possible that the user finds a discrepancy between the NF and the physical receipt. Any overage is consumed by ORFM, but no discrepancy resolution against overages is performed. In case of under-receipt, the receipt message coming from RWMS is the same as that of the ASN.

In case of under-receipt in the destination warehouse, the inventory mismatch in the source warehouse should be handled manually by doing an inventory adjustment. In the Brazil environment, the RMS subscription API routes the receipt, overage, and inventory adjustment messages to ORFM. If the inventory adjustment reason code is associated with an utilization code, the NF should be generated in ORFM.

On receiving the receipt message from RWMS-2, the inventory updates are done in RMS. The in-transit inventory at the destination location is reduced and on-hand inventory is increased. WAC is recalculated for the destination location. Subsequent to the inventory updates for receipts, the inventory is updated for the inventory adjustment for any under-receipt.

After physical receiving is done at RWMS-2, RWMS-2 publishes the receipt and overage receipt (if any) message to ORFM through the RMS subscription API. Though ORFM may consume this message, ORFM does not use the information for updating the inventory. Any over-receipt at RWMS is handled outside the system.

Return to Vendor (RTV) Shipments

An RTV request can be initiated in RMS (for returns from store), or in SIM and RWMS. An RTV request initiated in RMS should flow to SIM through RIB as per the existing flow.

RTV Shipped from Warehouse

When the RTV is picked and shipped from the warehouse, an RTV Shipment message should flow from RWMS to RMS through RIB. In the Brazil flow, after consuming the message, RMS routes it to ORFM; ORFM consumes the message from RMS.

Using the information in RTV Shipment message, ORFM should generate the outbound RTV NF. If there is reference to the inbound NF (through which the stock was received), the tax information is also retrieved from the inbound NF. If no inbound NF reference is required, the system should get the tax information through the TaxWeb Tax Rules. The TaxWeb Tax Rules should return appropriate CFOPs too. The NF should then be auto-approved. The shipment can leave the warehouse on printing of the approved NF.

The inventory updates are done in RMS and on-hand inventory is reduced. After the RMS updates are successful, the NF is approved. The RTV cost communicated to RMS is the tax-exclusive cost of the inventory returned. While making the transaction data entry, RMS should compare the RTV cost with the WAC, and if there is a mismatch, RMS should post a cost variance appropriately.

In Brazil, any outbound shipment should have a valid approved outbound NF accompanying it. Neither RWMS nor ORFM can control the outbound trailer leaving the warehouse or store without a valid NF. Business discipline must be followed to ensure this.

If for any reason the outbound NF cannot be generated or approved, the shipment cannot leave the source location. The RTV needs to be cancelled in that case. The system currently does not handle the cancellation of an RTV post-shipment from RWMS. It needs to be handled through manual inventory adjustments in RWMS and RMS simultaneously.

Figure 4–7   RTV Shipped from Warehouse
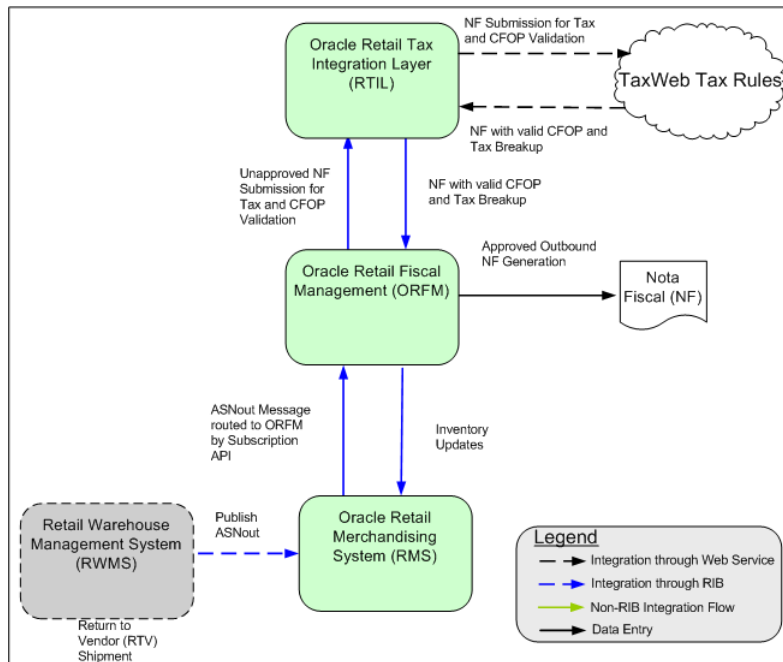
RTV Shipped from Store

From an ORFM perspective, RTV shipments from stores are similar to RTV shipments from RWMS. When the RTV is shipped from the store, the RTV Shipment message flows from SIM to RMS through RIB. In a Brazil environment, RMS should direct the message to ORFM.

Using the information in the RTV shipments message, ORFM generates the outbound RTV NF, gets the tax information through the TaxWeb Tax Rules, matches the NF with the requisition (transfer request in RMS, if available), and approves the NF. The shipment can leave the warehouse on printing of the approved NF.

During NF approval, the inventory updates and transaction data posting API in RMS is called. After RMS updates are successful, the NF is approved.

Figure 4–8   RTV Shipped from Store
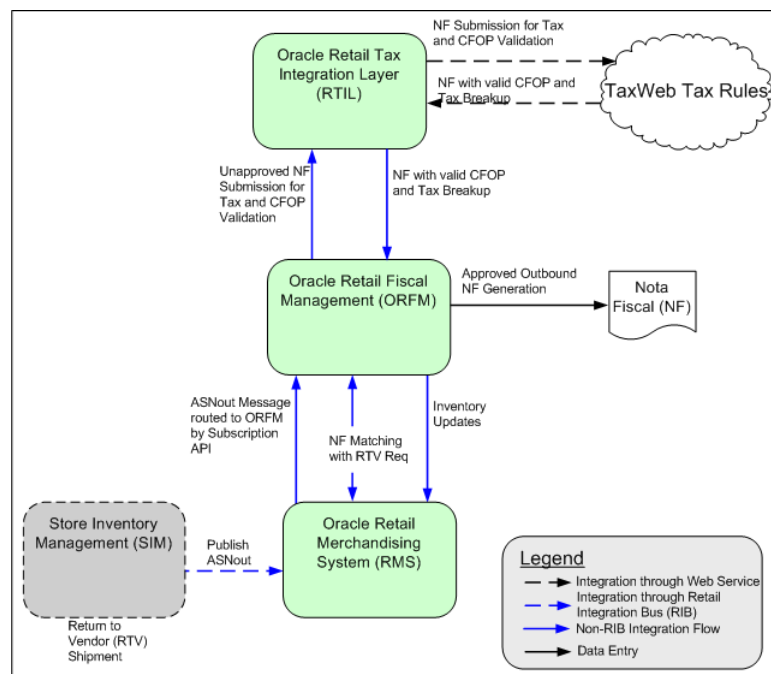
Retail Merchandise Authorization

The third party management system sends Retail Merchandise Authorization (RMA) information to ORFM. ORFM consumes this information and creates an inbound NF in Pending for Receiving status which informs the user that NF is created and waiting for the goods to be received.

Once the goods are received at warehouse, WMS sends the returns receipt message to the third party. The third party sends the Processed RMA information to ORFM and at the same time it sends the RTLOG file to ReSA for inventory updates.

The user can cancel the NF in ORFM, if the goods are not received at WMS.

Following are the assumptions:

- RMA is supported only in WMS since Return Receiving process is not supported by SIM

- You need to attach Virtual Store to the Warehouse in RMS so that RTLOG files can be created for that virtual store

- RTLOG files is generated by Third party order management system and sent directly to ReSA after the completion of RMA

- You need to create RMA manually in RWMS during the physical receiving with NF, since there is no integration between ORFM and RWMS for RMAs

- Third party order management application sends the costing information of the items related to corresponding RMA to ORFM

- Third party Order Management application validates the customer ID along with its fiscal details before sending to ORFM

- Only the processed status is sent, when the RMA processing is completed in Third party Order Management application

- There is no Discrepancy Identification and Resolution process for RMA

Inventory Adjustment

In Brazil, the inventory adjustment may also require NF generation, depending on the reason for which the inventory was adjusted. The same inventory adjustment reason codes defined in RMS are available in RWMS and SIM. In ORFM, the utilization codes are associated with the reason codes. It is not mandatory that each reason code is associated with some utilization code, but if the reason code is associated with the utilization code, an outbound NF is generated.

Because the inventory adjustment is done in SIM or RWMS, an Inventory Adjustment message should be published by SIM or RWMS. This message requires no changes to support Brazil specific flows. The message is consumed by RMS, and the subscription API routes it to ORFM in the Brazil environment. If the reason code associated with the inventory adjustment requires NF generation, an outbound NF is generated. After the NF validation and approval, the inventory in RMS is updated. If the reason code associated with the inventory adjustment does not require NF generation, using the same information received in the Inventory Adjustment message, RMS inventory is updated.

> **Note:** In Brazil, a positive inventory adjustment is not legal. Therefore it is assumed that the user will perform a negative inventory adjustment each time. The NF generated will then be an outbound NF.

Figure 4–9   Inventory Adjustment
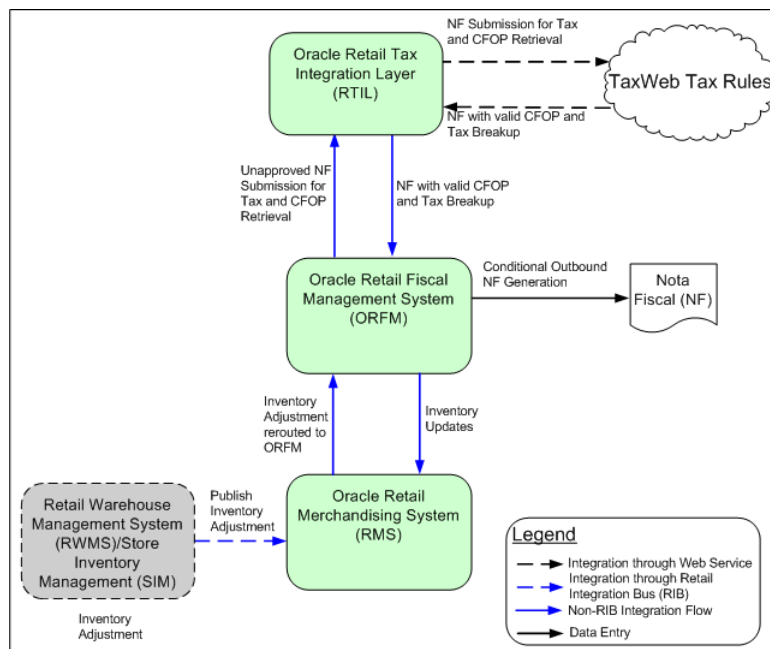
Finishers/Repairing Flow (Two-Legged Transfer)

A retailer can send goods out to an external finisher or supplier for either finishing work such as printing, dying, or embroidery, or repair in the case of damaged goods.

After the job is done, the finisher or supplier returns the goods to the same or a different location. While the goods are out of warehouse or store for repair or finishing, RMS needs to account for the inventory as the retailer's inventory.

This flow needs to be handled through two-legged transfers. The first leg of the transfer corresponds to the inventory being sent from the retailer's location (warehouse or store) to the finisher's location. The second leg corresponds to the receiving the finished goods (same or different SKU) or repaired goods back from the finisher at the retailer's location. Because the finisher's location is an outside location, the retailer is not responsible for the receiving or shipping at those locations.

Two-Legged Transfer Shipping from RWMS

When goods need to be sent to external finishers for finishing work, a two-legged transfer must be initiated in RMS, with transfer type as manual requisition. The first leg of the transfer (warehouse to finisher) flows to RWMS through RIB. RWMS persists it as a regular stock order. RWMS needs to do no special processing for such stock orders.

When goods need to be sent to external finishers or suppliers for repair, a two-legged transfer can be initiated in RMS, with the transfer type as manual requisition and the context type as repair. The first leg of the transfer (warehouse to finisher) flows to RWMS through RIB. RWMS stores it as a stock order with the type set to Repairing. All Repairing types of stock orders need special processing in RWMS. A regular wave should not select these stock orders for picking. These stock orders are picked manually, and only the inventory associated with appropriate trouble code can be picked up for repairing.

RWMS, as the physical owner of the inventory, has better visibility to the goods that need to be repaired. Therefore, a repairing stock order can be initiated in RWMS as well. As a stock order with the type of Repairing is created in RWMS, it flows to RMS through RIB. RMS, on consuming the message, creates a two-legged transfer with a context type of Repairing with the final location as the same origin warehouse.

As the required stock is picked and shipped at RWMS (for the external finisher's location), the ASNout message is published. ORFM consumes this message through the RMS subscription API and generates an outbound NF with Repairing as the requisition type. During NF validation, tax information is retrieved from TaxWeb Tax Rules, and NF matching is done with the requisition document. RMS inventory is updated, and then the NF is approved. On NF approval, the shipment is dispatched from the warehouse. RMS publishes the information of the second leg to RWMS through RIB subsequent to the inventory updates. RWMS does not maintain any link between the first and the second legs of the transfer. For RWMS, the two legs of the transfer should be two independent transactions. The second leg does not need to have context type as Repairing. If it is, RWMS ignores it if the recipient is the warehouse.

If the Auto Receive Stock indicator is enabled for the supplier in RMS, the on-hand inventory at the finisher's location is increased immediately as the stock is shipped from the retailer's location. If the indicator is disabled, on shipment of stock from retailer's location, in-transit inventory is increased at the external location. The on-hand inventory is updated only after RMS receives a receipt message from the finisher.

As the shipment leaves the warehouse, the first leg of the transfer is concluded from an ORFM perspective.

Figure 4–10   RWMS Two-Legged Transfer - First Leg (Shipping)
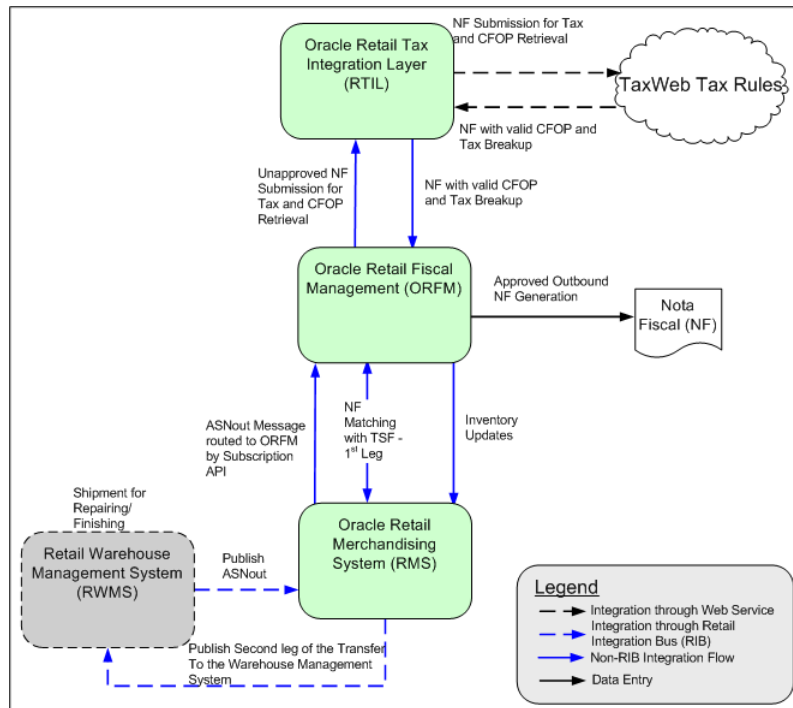
Two-Legged Transfer Receiving at RWMS

Receiving a two-legged transfer (second leg) is similar to transfer receiving or WO return receiving. The only difference is that the primary ASN is not already available in RWMS when the NF and schedule is entered in ORFM. The finisher ships the finished or repaired items (same or different SKUs) back to the warehouse along with the NF. As the shipment reaches the warehouse, a schedule is created and the inbound NF is entered into the system. The requisition number for the NF should be the main transfer number (two-legged). Each time, the requisition number on the NF should be the one in RMS. As the NF and schedule go through the tax validation and matching process, the schedule is submitted to RWMS for physical receiving. The message that flows from ORFM to RWMS through RIB should be equivalent to an ASNout message and should contain the schedule number and all the ASN details. The message should be detailed and should contain following fields:

■   ASN Number (ORFM should generate the ASN# and maintain the next up sequence number for it).

■   Requisition Number (the secondary transfer number)

■   Receiving Warehouse Number

■   Distinct Item Ids per transfer leg two

■   Consolidated Quantities per item-requisition

■   UOM

■   Schedule number

■   Container ID (ORFM should generate the container number and should maintain the next up sequence)

As RWMS receives this message, an automatic appointment (per schedule) is created and the ASN is linked to the appointment. After scheduling the appointment, physical receiving takes place. The user may find discrepancies between the NF and the physical receipt. In case of an overage, RWMS sends a separate overage message. Any over-receipt at RWMS is handled out of the system. In case of under-receipt, the receipt message coming from RWMS contains the same information as the ASN (items and quantities). Any inventory received short should flow to ORFM (through RIB and RMS subscription APIs) as a separate inventory adjustment message.

After ORFM consumes the receipt message, the inventory updates are done in RMS. The in-transit inventory at the destination location is reduced and on-hand inventory is increased. WAC is recalculated for the destination location. Subsequent to the inventory updates for receipts, the inventory should be updated for the inventory adjustment for any under-receipt.

Figure 4–11   RWMS Two-Legged Transfer - Second Leg (Receiving)
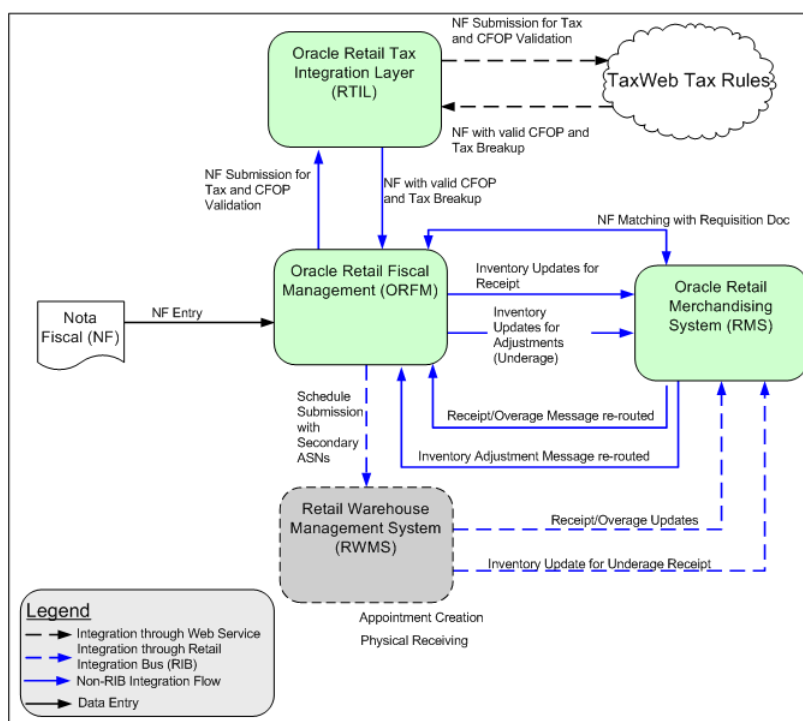
Two-Legged Transfer Shipping from SIM

When goods need to be sent from a store to an external finishers for finishing work, a two-legged transfer needs to be initiated in RMS, with the transfer type set as manual requisition and no context type. The first leg of the transfer (store to finisher) should flow to SIM through RIB. SIM should persist this as a return to warehouse (RTW) request.

When goods need to be sent out to external finishers or suppliers for repair, a two-legged transfer can be initiated in RMS, with the transfer type set to manual requisition and the context type to Repairing. The first leg of the transfer (warehouse to finisher) flows to SIM through RIB. SIM persists this as a return to warehouse request.

Because SIM is the physical owner of the inventory, it has better visibility to the goods that need to be repaired. Therefore, a repairing RTW can be initiated in SIM as well. As a RTW with the type set to Repairing is created in SIM, it should flow to RMS through RIB. RMS, on consuming the message, creates a two-legged transfer, with the context type set to the value from the transfer created in SIM, and the final location as the same origin store.

As the required stock is shipped from the store (for the external finisher's location), the ASNout message is published. RMS consumes this message and the subscription API directs it to ORFM. ORFM generates an outbound NF with Repairing as the requisition type. During NF validation, tax information is retrieved from TaxWeb Tax Rules. NF matching is done with the requisition document and on NF approval, the shipment is dispatched from the warehouse. RMS inventory is updated on NF approval. RMS publishes the information of the second leg to SIM through RIB after the inventory updates. SIM needs to maintain a link between the first and the second legs of the transfer because of serial-numbering requirements. RMS, while publishing the second leg to SIM, references the original transfer number. The second leg does not need to have the context type set to Repairing. SIM persists this transfer as a warehouse delivery.

If the Auto Receive Stock indicator is enabled for the supplier in RMS, the on-hand inventory at the finisher's location is increased immediately as the stock is shipped from the retailer's location. If the indicator is disabled, on shipment of stock from retailer's location, in-transit inventory is increased at the external location. The on-hand inventory is updated only after RMS receives a receipt message from the finisher.

As the shipment leaves the store, the first leg of the transfer is concluded from the ORFM perspective.

Figure 4–12   SIM Two-Legged Transfer - Shipping
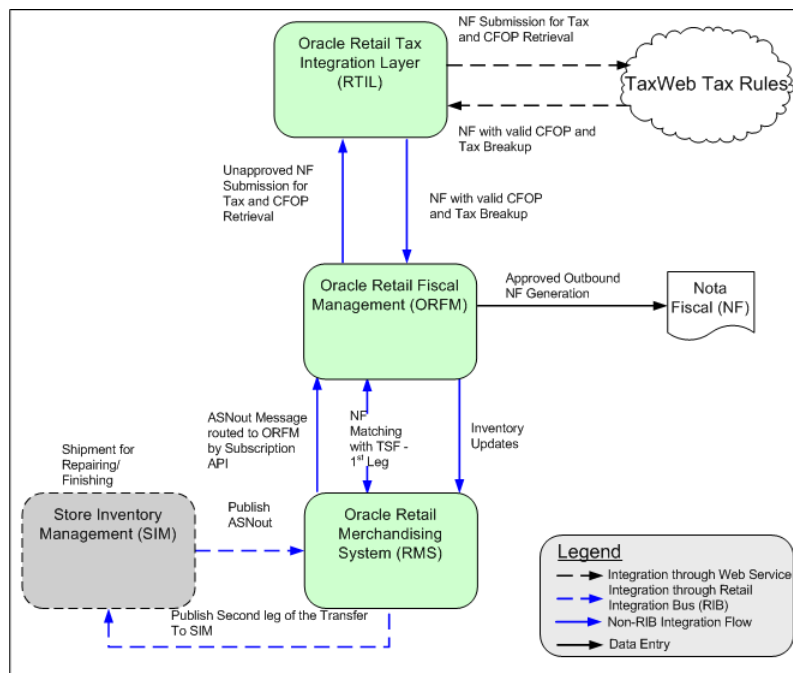
Two-Legged Transfer Receiving at SIM

Receiving a two-legged transfer (second leg) is similar to a transfer receiving at a store. The only difference is that the primary ASN is not already available in SIM when the NF and schedule is entered in ORFM. The finisher ships the finished or repaired items (same or different SKUs) back to the warehouse along with the NF. As the shipment reaches the store, a schedule is created and the inbound NF is entered into the system. The requisition number for the NF should be the main transfer number (two-legged). As the NF and schedule go through the tax validation and matching process, the ASN is published to SIM for physical receiving. The message that flows from ORFM to SIM through RIB should be the equivalent of an ASNout message. The auto-receiving indicator should be 'Y' for this ASN. The detailed message should contain the following information:

- ASN Number (ORFM should generate the ASN number and maintain the next up sequence number for it)

- Requisition Number (the secondary transfer number)

- Receiving store ID

- Distinct Item IDs per transfer leg-two

- Consolidated quantities per item-requisition

- UOM

- Container ID (ORFM should generate the container number and should maintain the next-up sequence)

- Auto-receiving indicator (value should be 'Y')

As SIM receives this message, the ASN is automatically received and a receipt message is sent to ORFM. On receipt of the receipt message from SIM, the inventory updates are done in RMS. The in-transit inventory at the destination location is reduced and on-hand inventory is increased. WAC is recalculated for the destination location.

Figure 4–13   SIM Two-Legged Transfer - Receive
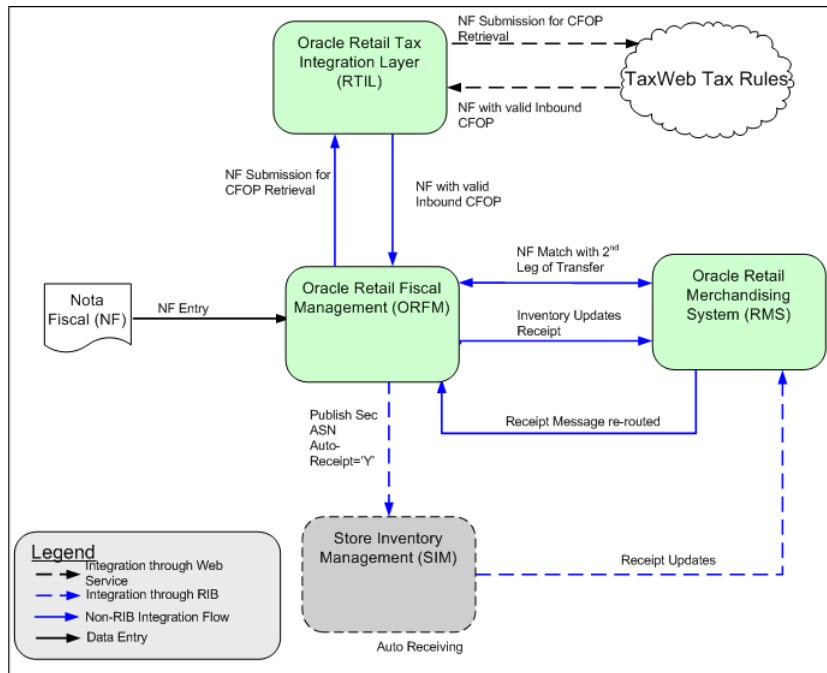
RIB-Based Integration

ORFM can integrate with other Oracle Retail products, such as SIM and RWMS, through Oracle Retail Integration Bus (RIB). RIB utilizes a publish and subscribe (pub/sub) messaging paradigm with some guarantee of delivery for a message. In a pub/sub messaging system, an adapter publishes a message to the integration bus that is then forwarded to one or more subscribers. The publishing adapter does not know or care how many subscribers are waiting for the message, what types of adapters the subscribers are, what the subscribers current states are (running/down), or where the subscribers are located. Delivering the message to all subscribing adapters is the responsibility of the integration bus.

See the *Oracle Retail Integration Bus Operations Guide* and other RIB documentation for additional information.

HTTP-Based Integration

ORFM integrates with RTIL through a HTTP based interfaces. See "RTIL Architecture" on page 7-4 for additional information.

Nota Fiscal – Receiving and Issuing

The Nota Fiscal receiving (inbound) and issuing (outbound) are controlled within ORFM and all integration with RMS, RWMS, and SIM is based on the physical movement of the products.

Nota Fiscal Receiving

When ORFM is enabled, SIM and RWMS are not allowed to physically receive any inventory prior to fiscal receiving (that is, inbound Nota Fiscal processing) is completed in ORFM. After physical receiving in SIM or RWMS, the inventory in RMS

is not updated until the fiscal receipt and physical receipt comparison is completed in ORFM and any discrepancy is resolved.

Figure 4-14 outlines the ORFM business process for inbound operations.

Figure 4–14   Inbound Process Flow
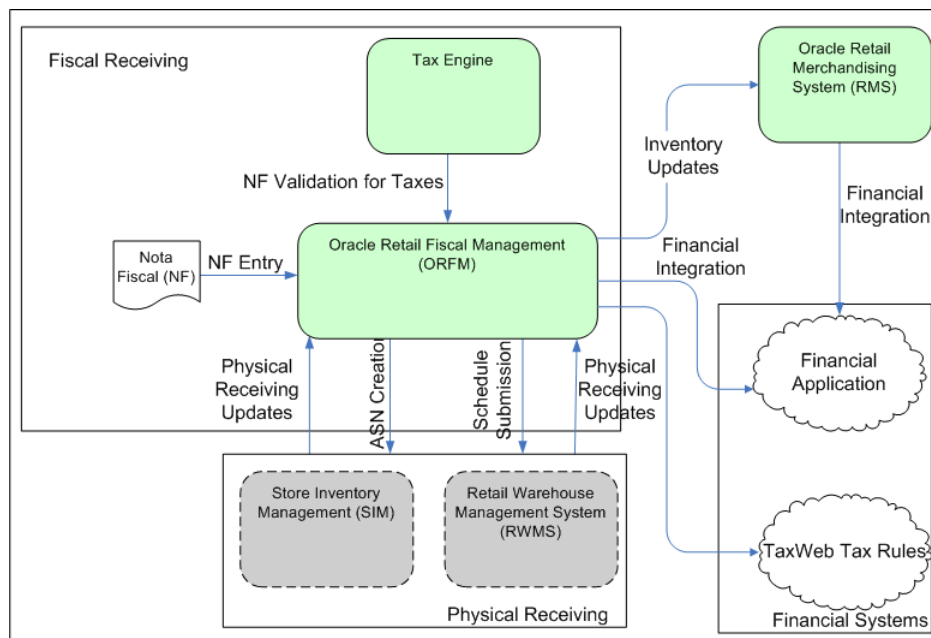
Inbound Process Flow

The inbound process flow is as follows:

1. Enter the Nota Fiscal (NF) in the Oracle Retail Fiscal Management (ORFM) module.

2. When the shipment arrives at the warehouse or the store, create a schedule and enter the NFs received.

   > **Note:** You can link more than one NF to a schedule.

3. After NF entry, validate the NFs. In the validation process, the application checks for data integrity. Match the NF with the requisition documents in Retail Merchandising System (RMS). This process is called as Fiscal Receiving.

4. If the NF and the PO does not match, the NF is in discrepancy. You can identify the following discrepancies with the ORFM application:

   - Unit Cost Discrepancy for each item in the NF

   - Quantity Discrepancy for each item in the NF

   - Tax Discrepancy at aggregate level (such as NF header level or individual item level) for all the applicable taxes

5. After validation, send the schedule to the Warehouse Management System (RWMS), and Store Inventory Management (SIM). After physical receiving,

RWMS and SIM publish the receipt updates to ORFM. This completes the NF processing.

6. After NF processing is complete, ORFM calls RMS to update inventory and WAC.

7. Send the transaction data in RMS and ORFM to a financial application. For fiscal reporting purposes, send the NF data in ORFM to the fiscal reporting system like Mastersaf.

Because the Nota Fiscal contains taxation information, ORFM calls the tax engine for calculations during the receiving and issuing of a Nota Fiscal.

Nota Fiscal Issuing

When ORFM is enabled, RMS inventory is not updated immediately when any inventory is shipped out from RWMS or SIM. The inventory movement information is first sent to ORFM for the generation of the outbound Nota Fiscal. After approval of the outbound Nota Fiscal, ORFM sends the inventory updates to RMS.
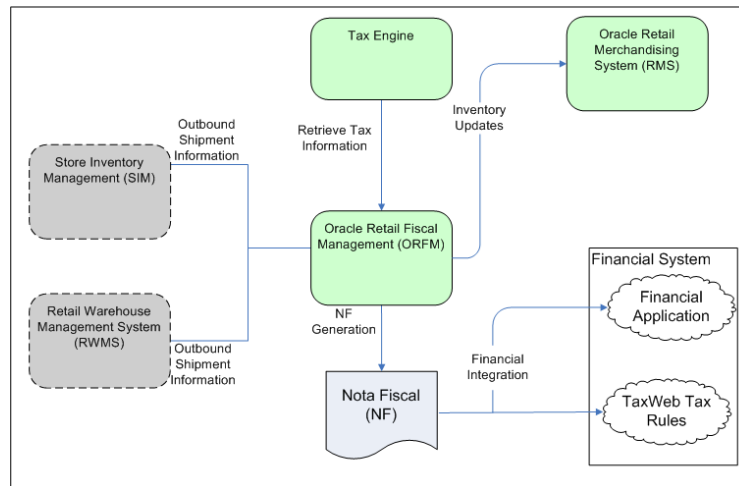
Outbound Process Flow

The outbound process flow is as follows:

1. Enter the outbound shipment information in SIM or RWMS and send it to ORFM.

2. ORFM sends the outbound shipment information to the tax engine. The tax engine figures the tax information and returns it to ORFM.

3. Create the Nota Fiscal based on the outbound shipment information and the tax information.

4. After NF processing is complete, ORFM calls RMS to update inventory and WAC.

Triangulation

ORFM supports the triangulation process. Triangulation is a typical purchase transaction where the PO is raised for a supplier (referred to as main/ordered supplier) but the goods are shipped to the retailer by a different supplier (referred to as the delivery supplier). In Brazil, for triangulation transactions, NFs are received from both the main and delivery suppliers. Both the NFs are for the same stock but the details on both the NFs may vary. For example, federal tax such as IPI usually appears on the main supplier's NF, while state taxes such as ICMS and ICMS-ST usually appear on the delivery supplier's NF.

EDI NF

ORFM supports generation of NF through EDI. Supplier will interface the NF details to RFM and is captured in the EDI tables. RFM has a batch which when run fetches data from the EDI tables and generates the NF in Worksheet status. The NFs are generated with the header, detail, informed header taxes, informed detail taxes, payment information, and complementary details fetched from the EDI tables interfaced by the supplier.

ORFM and RWMS

RWMS is integrated with the ORFM through RIB for the below business transactions:

PO Receiving

- ORFM publishes the Appointment Quantity based on the NF details to WMS
- WMS schedules the Appointment based on the above information
- WMS can over-receive/under-receive during physical receipt. The over-receipt is through new overage screen in WMS
- The receipt information with the damaged and overage details are published by WMS to ORFM and ORFM to RMS

Transfer Receiving

- WMS and ORFM subscribes to the ASN information from the sourcing location
- After the ASN is processed in ORFM, a secondary ASN is published to the warehouse receiving the transfer
- The warehouse will receive against the secondary ASN and the receipt information is interfaced to ORFM through RIB

Brazil requirements are strict for receiving exactly what is on the NF. ORFM provides delivery information to RWMS, from which the user creates receiving appointments. The ability to alter appointment quantities is not available based on system parameter configuration. Validation exists to compare the appointment quantities with the quantities supplied by ORFM. Users can split items as needed by altering case pack sizes, but the totals must match with ORFM. The System Configuration Parameter -- appt_update_allowed should be set to N to disallow the appointment quantities being updated during receiving.

Requirements for each appointment are grouped into the same unique schedule number provided by ORFM. That schedule number is added to the appointment table and is searchable by users.

ORFM and SIM

Nota Fiscal (NF) is a Brazilian fiscal legal document that needs to be generated to follow the physical movement of goods. ORFM handles the specific processing.

SIM is impacted by NF in the following ways:

- Shipping to warehouse or vendor
- DSD Receiving
- Transfer receiving
- Inventory Adjustments and stock counts

- Receive Unit Adjustments

The receiving process in SIM for internal (warehouse and store) deliveries does not allow under receiving or over receiving because of the following:

- Transfers cannot be modified in Brazil through the user interface security; they are auto-received.

- No adjustments are allowed in Brazil.

For warehouse deliveries, NF and SIM get the ASN information from the warehouse. After the ASN has been processed in NF, SIM gets a second ASNin message. This message launches a new auto-receive process. Security is used to limit users' access to the warehouse delivery dialog.

The reason for SIM to still subscribe to the ASNin message from RWMS or another store is so that the system knows which items are on their way while they are in transit, for more accurate inventory information.

ORFM Integration with Financial Application

Enterprise Business Suite (EBS) is integrated as financial system with Oracle Retail applications including RMS/ReSA. Financial integration between Oracle Retail and EBS is provided in the Application Integration Architecture (AIA) Process Integration Pack (PIP), **Oracle Financial Operations Control Integration Pack for Oracle Retail Merchandising Suite, and E-Business Suite Financials**. The Brazilian localized extensions that need to be used in conjunction with the AIA PIP are discussed in this section.

As part of the existing AIA PIP Integration, the EBS financial module communicates foundation information for supplier setup, payment terms, exchange rates, and chart of accounts to RMS/ReSA. In base version of Oracle Retail, EBS Financial module also communicates Chart of Accounts (COA) to ReIM. In Brazil localization, chart of accounts is not interfaced, and needs to be manually set up in ORFM.

Transaction information is communicated from RMS/ReSA to EBS Financial General Ledger through Oracle Data Integrator (ODI) . In base version of Oracle Retail, ReIM communicates payment requests to EBS Accounts Payable (AP) through Retail enabled ODI. In Brazil localization, ORFM has built upon similar interface tables (AP Stage head and detail) as ReIM to publish accounts payable data. These tables are used to publish NF payment data from ORFM to EBS.

The following are discussed as a part of EBS integration:

- Supplier Setup and Integration

- ORFM AP Integration Stage Head and Detail

Figure 4–15   Flow of information between Financial System and Oracle Retail (Brazil Localization)
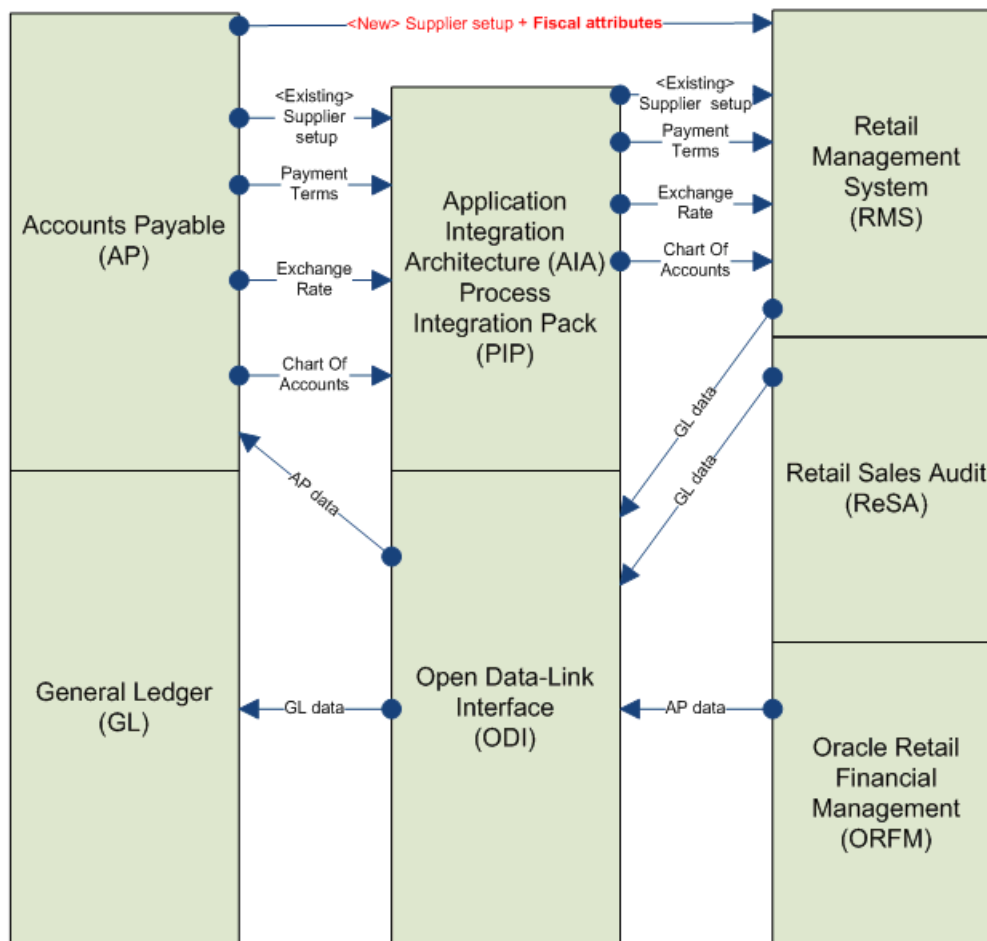
Supplier Setup and Integration

EBS AP allows setup of Brazilian specific fiscal attributes during supplier setup in a Brazil localized environment, for suppliers located in Brazil. These attributes are passed from EBS to RMS. The Vendor subscription API in RMS is enhanced to subscribe to additional fiscal attributes of the supplier.

These fiscal attributes are passed from EBS AP to RMS in both Supplier Create and Supplier Update/Modify messages flows. Fiscal attributes added to new suppliers/sites that are newly created as well as modifications to these attributes will flow through to RMS as part of the integration.

The global attributes are passed from EBS AP to RMS and consumed at RMS. These attributes are captured in V_BR_SUPS of RMS as L10N Attributes (LFAS).

Table 4–1 provides the additional attributes that are passed from EBS AP to RMS and consumed at RMS.

Table 4–1   Additional Brazil Localization attributes passed from EBS AP to RMS

EBS Field	Description	RMS Field	Description
global_attribute9	EBS-Inscription Type (CNPJ or CPF)	If type = "CNPJ" then TAXPAYER_ TYPE="J" if type = "CPF" then TAXPAYER_TYPE = "F"	Mapped to Taxpayer type field
global_attribute10	EBS-Inscription Number	If type = "CNPJ" concatenate fields global_attribute10, global attribute11 and global attribute12 into RMS field CNPJ If type = "CPF" concatenate fields global_attribute10 and global attribute12 into RMS field CPF	Concatenate and record value under CNPJ or CPF field
global_attribute11	EBS-Inscription Branch		
global_attribute12	EBS-Inscription Digit		
global_attribute13	EBS-State Inscription	IE	Mapped to IE field
global_attribute14	EBS-City Inscription	IM	Mapped to IM field
global_attribute17	SUFRAMA	SUFRAMA	Mapped to SUFRAMA
NIT	EBS - NIT (Numero de Inscrição do Trabalhador perante a previdencia social) - only for individual taxpayers	NIT	Mapped to NIT (Numero de Inscrição do Trabalhador perante a previdencia social) - only for individual taxpayers
ISS_CONTRIB_IND	EBS - ISS contributor indicator	ISS_CONTRIB_ID	Mapped to ISS contributor indicator
SIMPLES_IND	EBS - SIMPLES contributor indicator	SIMPLES_CONTRIB_ ID	Mapped to SIMPLES contributor indicator
ST_CONTRIB_IND	EBS - ST contributor indicator	ST_CONTRIB_ID	Mapped to ST contributor indicator
RURAL_PROD_IND	EBS - Rural producer	RURAL_PROD_IND	Mapped to Rural producer
ICMS_CONTRIB_ IND	EBS - ICMS contributor indicator	ICMS_CONTRIB_ IND	Mapped to ICMS contributor indicator
IPI_IND	EBS - IPI contributor indicator	IPI_IND	Mapped to IPI contributor indicator
PIS_CONTRIB_IND	EBS - PIS contributor indicator	PIS_CONTRIB_IND	Mapped to PIS contributor indicator
COFINS_CONTRIB_ IND	EBS - COFINS contributor indicator	COFINS_CONTRIB_ IND	Mapped to COFINS contributor indicator
IS_INCOME_ RANGE_ELIGIBLE	EBS - is income rage eligible indicator	IS_INCOME_ RANGE_ELIGIBLE	Mapped to Is Income rage eligible indicator
IS_DISTR_A_ MANUFACTURER	EBS - is the distributor a manufacturer indicator	IS_DISTR_A_ MANUFACTURER	Mapped to Is the distributor a manufacturer indicator

Table 4–1   Additional Brazil Localization attributes passed from EBS AP to RMS

EBS Field	Description	RMS Field	Description
ICMS_SIMPLES_RATE	EBS - ICMS SIMPLES rate	ICMS_SIMPLES_RATE	Mapped to ICMS SIMPLES rate
Address_line4	EBS - Neighborhood	ICMS_SIMPLES_RATE	Mapped to ICMS SIMPLES rate

ORFM AP Integration Stage Head and Detail

As ReIM is not part of the Brazil solution, the ReIM to AP flow is replaced with an ORFM to AP flow.

Transaction data in ORFM maintains NF information so that it can be sent to the financial application.

Different transaction codes are used for tax-exclusive cost (also referred as BC), cost components (such as freight or insurance), taxes that are input creditable, and taxes that affect cost (non-input credit type).

For tax-related transaction data, the user can configure the system to make a separate entry for each tax code. By doing so, the user can map the same transaction code having different tax codes in different accounts.

Following tables describe the cross mapping between financial staging tables in ORFM and AP:

Table 4–2   Header level mappings between financial staging tables in ORFM and AP

Product: E-Business Suite		Product: Retail RFM		
Version: 12.1.1		Version: 13.0.2		
Table: AP_INVOICE_LINES_INTERFACE (Target		View: IM_AP_STAGE_HEAD (Source)		

Column Name	Data Type	Column Name	Data Type	Comments
INVOICE_ID	NUMBER	FISCAL_DOC_ID	NUMBER	Populated with ORFM internal sequencing ID
INVOICE_NUM	VARCHAR2	FISCAL_DOC_NO	VARCHAR2	Populated with the NF fiscal doc ID
INVOICE_TYPE_LOOKUP_CODE	VARCHAR2	FISCAL_TYPE_LOOKUP_CODE	VARCHAR2	Type of Invoice (can be Standard, Credit or Prepayment)
INVOICE_DATE	DATE	TRAN_DATE	DATE	Populated with the NF issue date
VENDOR_ID	NUMBER	VENDOR	NUMBER	Supplier corresponding to the Supplier Site on the NF
VENDOR_SITE_ID	NUMBER	ORACLE_SITE_ID	NUMBER	Supplier site ID of the 'Remit to site' corresponding to the Supplier on the NF
INVOICE_AMOUNT	NUMBER	AMOUNT	NUMBER	

Table 4–2   (Cont.)   Header level mappings between financial staging tables in ORFM and

Product: E-Business Suite	Product: Retail RFM
Version: 12.1.1	Version: 13.0.2
Table: AP_INVOICE_LINES_ INTERFACE (Target	View: IM_AP_STAGE_HEAD (Source)

Column Name	Data Type	Column Name	Data Type	Comments
INVOICE_ CURRENCY _CODE	VARCHAR2	CURRENCY _CODE	VARCHAR2	Currency based on System Option Setup
EXCHANGE _RATE	NUMBER	EXCHANGE _RATE	NUMBER	
EXCHANGE _RATE_ TYPE	VARCHAR2	EXCHANGE _RATE_ TYPE	VARCHAR2	
TERMS_ID	NUMBER	BEST_ TERMS	VARCHAR2	Populated with best terms id picked based on 'Best Terms calculation' logic that uses terms on the Supplier/ POs that are present on the NF
SOURCE	VARCHAR2	RFM		Populated with value ′RFM′
ORG_ID	NUMBER	ORG_UNIT	NUMBER	
TERMS_ DATE	DATE	BEST_ TERMS_ DATE	DATE	

Table 4–3   Detail/Line Level mappings between financial staging tables in ORFM and AP

Product: E-Business Suite	Product: Retail RFM
Version: 12.1.1	Version: 13.0.2
Table: AP_INVOICE_LINES_ INTERFACE (Target	View: IM_AP_STAGE_ DETAIL (Source)

Column Name	Data Type	Column Name	Data Type	Comments
INVOICE_ID	NUMBER	FISCAL_ DOC_ID	NUMBER	Populated with ORFM internal sequencing ID
LINE_TYPE_ LOOKUP_ CODE	VARCHAR2	LINE_TYPE_ LOOKUP_ CODE	VARCHAR2	Type of invoice line (Item, Tax, Miscellaneous)
AMOUNT	NUMBER	AMOUNT	NUMBER	
ACCOUNTI NG_DATE	DATE	CREATE_ DATE_TIME	DATE	
DIST_ CODE_ CONCATEN ATED	VARCHAR2	SEGMENT1 - SEGMENT20	VARCHAR2	
ORG_ID	NUMBER	ORG_UNIT	NUMBER	

5

Publication and Subscription API Designs

This chapter includes publication designs that describe, on a technical level, how ORFM publishes messages to the Oracle Retail Integration Bus (RIB).

The following topics are covered in this chapter:

- Secondary ASNOut Publication API
- POSchedule Publication API

Secondary ASNOut Publication API

The business overview, functionality checklist, and package impact of the Secondary ASNOut Publication API is covered in the subsequent sections.

Business Overview

ORFM publishes this message while submitting a schedule based on PO to SIM for receiving or transfer to RWMS/SIM for receiving. RWMS/SIM subscribes to this message to do the receiving.

This RIB message is triggered by submitting a schedule in ORFM. The output message is in hierarchical structure, with ASN information in the Description section, distro, item, and carton information in the detail sections.

To facilitate the routing of data, the header level of routing information contains the to_phys_loc with the value of location id and 'to_phys_loc_type' with the value W/S and from_phys_loc and from_phys_loc_type information. Detail level routing information contains the source_app with the value of 'ORFM'. This allows RIB to route the message from ORFM to RWMS/SIM.

Functionality Checklist

Table 5–1

Description	ORFM	RIB
ORFM must publish ASNOut information		
Create new publisher	X	X

Form Impact

None

Business Object Records

None

Package Impact

This section covers the impact of the package – FM_SECONDARY_ASNOUT.

Business Object ID

The business object ID for ASNOut publisher is recv_no.

Package Name: FM_SECONDARY_ASNOUT

Spec File Name: fm_secondary_asnout_sql_s.pls

Body File Name: fm_secondary_asnout_sql_b.pls

Package Specification – Global Variables

L_FAMILY constant varchar2(25) := asnout

Function Level Description

ADDTOQ – The inbound Schedule for a transfer is published from ORFM as one ASNout message to SIM/RWMS and PO published to SIM.

This function adds a record in the `FM_RIB_RECEIVING_MFQUEUE` table with asnoutmod message_type for transfer, asnoutcre for PO/Two-legged transfer/Repairing and ASNOut family for each inbound schedule/ASN. The published flag should be set to U.

GETNXT – The RIB calls GETNXT function to get ASNOut messages. It performs a cursor loop on the unpublished records with PUB_STATUS = U and FAMILY=asnout on the `FM_RIB_RECEIVING_MFQUEUE` table. For each record retrieved, GETNXT gets the following:

- A lock on the queue `FM_RIB_RECEIVING_MFQUEUE` table for the records which ORFM is going to publish. The lock is obtained by calling the function LOCK_THE_BLOCK.

- The information from the `FM_RIB_RECEIVING_MFQUEUE` table is passed to PROCESS_QUEUE_RECORD function. This function will build the Oracle Object message for RIB.

  - If PROCESS_QUEUE_RECORD does not run successfully, GETNXT raises an exception call to HANDLE_ERRORS function that updates the PUB_STATUS to 'H'.

PUB_RETRY – It performs a cursor loop on the records with PUB_STATUS = H and FAMILY = asnoutmod. For each record retrieved, PUB_RETRY gets the following:

- A lock on the queue `FM_RIB_RECEIVING_MFQUEUE` table for the records which ORFM is going to publish. The lock is obtained by calling the function LOCK_THE_BLOCK.

- The information from the `FM_RIB_RECEIVING_MFQUEUE` table is passed to PROCESS_QUEUE_RECORD function. This function will build the Oracle Object message for RIB.

- If PROCESS_QUEUE_RECORD does not run successfully, PUB_RETRY raises an exception call to HANDLE_ERRORS function that updates the PUB_STATUS to 'H'.

PROCESS_QUEUE_RECORD (local) – This function is called from GETNXT and PUB_RETRY functions for those Inbound schedule records that are in PUB_STATUS = 'U' and 'H' respectively. This function calls API_LIBRARY.GET_RIB_SETTINGS to get the RIB settings for the ASNOut message family. It also calls BUILD_HEADER_OBJECT function to build the header and detail sections of secondary ASNOut message. Once the Oracle object is successfully formulated, this function deletes the current record from the queue (that is, `FM_RIB_RECEIVING_MFQUEUE` table) by calling DELETE_QUEUE_REC function.

BUILD_HEADER_OBJECT (local) – This function will take necessary data from `FM_RIB_STG_RECEIVING_HEADER` and `FM_RIB_STG_RECEIVING_DETAIL` tables for the current schedule_nbr/ASN and prepare the Oracle object for the ASNOut message. The structure of ASNOut is same as base payloads with an additional attribute containing schedule_nbr in the RIB_ASNOutDesc_REC section for RWMS.

The same function will build ASNOut message structure with an additional attribute of auto_receipt flag set to Y in the RIB_ASNOutDesc_REC section for SIM. This flag will otherwise be NULL in case of RWMS.

While publishing the secondary ASNOut to RWMS and SIM, ORFM will be sending three routing information through Oracle Object - RIB_ROUTINGINFO_REC.

1. Sending location

2. Receiving location

3. A hard-coded string depicting source application - ORFM

This is required to indicate the source of ASNOut message is ORFM and hence, it is meant only for RWMS and SIM applications to consume it. However, RMS also subscribes to ASNOut message, but in this case as the source is ORFM - RMS subscriber package will discard the message.

Call the BUILD_DETAIL_OBJECTS to get the details of the current schedule record. The container_qty is a required field on the RIB object. So, ORFM sends it as 1 instead of NULL.

This function will also be modified to publish the following additional fields in the publishing message to RWMS/SIM:

1. Schedule number (will be NULL for SIM)

2. Auto_receive flag (will be NULL for RWMS and Y for SIM)

BUILD_DETAIL_OBJECTS (local) – This function is responsible for building the detail section of ASNOut message. It builds as many detail Oracle Object as many items present in a transfer. It will fetch the detail records from FM_RIB_STG_RECEIVING_HEADER, FM_RIB_STG_RECEIVING_DETAIL for the given schedule number and will assign the above details into RIB_ASNOutItem_REC, RIB_ASNOutCtn_REC and RIB_ASNOutDistro_REC record groups.

This package will be modified to generate the ASN# and CTN# and will maintain the next up sequence number for it in case of PO to SIM and two-legged transfer/repairing publishing to RWMS/SIM.

LOCK_THE_BLOCK (local) – This function locks all queue records for the current business object. This is to ensure that GETNXT does not wait on any business processes that currently have the queue table locked and have not committed.

HANDLE_ERRORS (local) – HANDLE_ERRORS is called from GETNXT and PUB_ RETRY when an exception is raised. If the error is a non-fatal error, GETNXT passes the sequence number of the driving FM_RIB_RECEIVING_MFQUEUE record back to the RIB in the ROUTING_INFO. It sends back a status of 'H'ospital to the RIB as well. It then updates the status of the queue record to 'H'ospital, so that it will not get picked up again by the driving cursor in GETNXT.

DELETE_QUEUE_REC (local) – DELETE_QUEUE_REC is called from PROCESS_ QUEUE_RECORD once a queue record is formed from `FM_RIB_RECEIVING_ MFQUEUE`, `FM_RIB_STG_RECEIVING_DETAIL`, and `FM_RIB_STG_RECEIVING_ HEADER` tables. This is just to not pick the same record again.

Trigger Impact

Trigger Name
NA

Trigger File Name
NA

Table
NA

Message XSD

The file name that corresponds with the message type. Refer the RIB documentation for this message type in order to get a detailed picture of the composition of the message.

Table 5–2

Message Types	Message Type Description	XML Schema Definition (XSD)
ASNOutCre	ASNOut Create message	ASNOutDesc.xsd
ASNOutMod	ASNOut Modify message	ASNOutDesc.xsd

Table Impact

Table 5–3

TABLE	SELECT	INSERT	UPDATE	DELETE
FM_RIB_RECEIVING_MFQUEUE	Y	Y	Y	Y
FM_RIB_STG_RECEIVING_ HEADER	Y	Y	N	Y
FM_RIB_STG_RECEIVING_ DETAIL	Y	Y	N	Y
FM_RECEIVING_HEADER	Y	N	N	N
FM_RECEIVING_DETAIL	Y	N	N	N

Design Assumptions

This message is applicable only if ORFM exists in case of Brazil countries. It is applicable for PO/Schedule publication to SIM and transfer/schedule publishing to RWMS/SIM.

POSchedule Publication API

The business overview, functionality checklist, and package impact of the POSchedule Publication API is covered in the subsequent sections.

Business Overview

ORFM publishes this message while submitting a schedule based on 'PO' to RWMS for receiving. RWMS subscribes to this message it order to create schedule based appointment and does the receiving.

This RIB message is triggered by submitting of PO based Schedule in ORFM. The output message is in hierarchical structure, with Schedule no in the Description section, PO information in the Header and Item details in the detail section.

To facilitate the routing of data, the header level of routing info contains the to_phys_loc with the value of location id and to_phys_loc_type with the value W. Detail level routing info contains the source_app with the value of ORFM. This allows the RIB to route the message from ORFM to RWMS.

Functionality Checklist

Table 5–4

Description	ORFM	RIB
ORFM must publish POSchedule information		
Create new publisher	X	X

Form Impact

None

Business Object Records

None

Package Impact

This section covers the impact of the package – FM_SCHED_SUBMIT.

Business Object ID

The business object id for POSchedule publisher is recv_no that is, schedule no.

Package Name: FM_SCHED_SUBMIT

Spec File Name: fm_sched_submit_sql_s.pls

Body File Name: fm_sched_submit_sql_b.pls

Package Specification - Global Variables

FAMILY constant varchar2(25) := poschedule

LP_cre_type varchar2(15) := poschedulecre.

Function Level Description

ADDTOQ – This function adds the PO Schedule to the `FM_RIB_RECEIVING_ MFQUEUE` table in 'U'npublished status. It stages the recv_no (schedule number) for publishing to the RIB.

GETNXT – This procedure is called from the RIB to get the next 'U'npublished schedule number from `FM_RIB_RECEIVING_MFQUEUE` table for publishing. Based on the seq_no on FM_RIB_RECEIVING_MFQUEUE, it calls the PROCESS_QUEUE_ RECORD procedure.

GETNXT gets the following:

- A lock on the queue `FM_RIB_RECEIVING_MFQUEUE` table for the records which ORFM is going to publish. The lock is obtained by calling the function LOCK_ THE_BLOCK.

- The information from the `FM_RIB_RECEIVING_MFQUEUE` table is passed to PROCESS_QUEUE_RECORD function. This function will build the Oracle Object message for RIB.

  If PROCESS_QUEUE_RECORD does not run successfully, GETNXT raises an exception.

- If any exception is raised in GETNXT, including the exception raised by an unsuccessful call to PROCESS_QUEUE_RECORD, HANDLE_ERRORS is called.

PUB_RETRY – It will process the record for PUB_STATUS = 'H' and MESSAGE_ TYPE = poschedcre. For each record retrieved, PUB_RETRY gets the following:-

- A lock on the queue `FM_RIB_RECEIVING_MFQUEUE` table for the records which ORFM is going to publish. The lock is obtained by calling the function LOCK_ THE_BLOCK.

- The information from the `FM_RIB_RECEIVING_MFQUEUE` table is passed to PROCESS_QUEUE_RECORD function. This function will build the Oracle Object message for RIB.

- If PROCESS_QUEUE_RECORD does not run successfully, PUB_RETRY raises an exception. Call to HANDLE_ERRORS function that updates the PUB_STATUS to 'H'.

- If any exception is raised in GETNXT, including the exception raised by an unsuccessful call to PROCESS_QUEUE_RECORD, HANDLE_ERRORS is called.

PROCESS_QUEUE_RECORD (local) – This function is called from GETNXT and PUB_RETRY functions for those Inbound schedule records that are in PUB_STATUS = U and H respectively. This function calls API_LIBRARY.GET_RIB_SETTINGS to get the RIB settings for its family (posched). It also calls BUILD_HEADER_OBJECT function to build header and detail sections of the PO Schedule Oracle objects. Once

the Oracle object is successfully formulated, this function deletes the current record from the queue (that is, `FM_RIB_RECEIVING_MFQUEUE` table) by calling DELETE_ QUEUE_REC function.

BUILD_HEADER_OBJECT (local) – Take all necessary data from `FM_RIB_STG_ RECEIVING_HEADER` and `FM_RIB_STG_RECEIVING_DETAIL` tables for the current schedule and put it into a RIB_POScheduleDesc_REC object. Two routing information (source Application that is, ORFM and location that is, Warehouse) has to be sent to RIB through RIB_ROUTINGINFO_REC for PO.

This function publishes the following fields in the header section of the POSchedule message to RWMS:

- schedule_nbr

- receiving_location_id

This function will give a call to the BUILD_DETAIL_OBJECTS function.

BUILD_DETAIL_OBJECTS (local) – The function is responsible for building detail level Oracle Objects. It fetches the detail records from FM_RIB_STG_RECEIVING_ DETAIL for the given schedule number and formulates the message structures: RIB_ POSchedule_REC and RIB_POScheduleDtl_REC.

This function publishes the following fields to the above Oracle objects:

- requisition_nbr

- requisition_type

- item_id

- consolidate_qty

LOCK_THE_BLOCK (local) – This function locks all queue records for the current schedule_nbr. This is to ensure that GETNXT does not wait on any business processes that currently have the queue table locked and have not committed.

HANDLE_ERRORS (local) – HANDLE_ERRORS is called from GETNXT and PUB_ RETRY when an exception is raised. If the error is a non-fatal error, GETNXT passes the sequence number of the driving FM_RIB_RECEIVING_MFQUEUE record back to the RIB in the ROUTING_INFO. It sends back a status of 'H'ospital to the RIB as well. It then updates the status of the queue record to 'H'ospital, so that it will not get picked up again by the driving cursor in GETNXT.

If the error is a fatal error, a status of 'E'rror is returned to the RIB.

DELETE_QUEUE_REC (local) – DELETE_QUEUE_REC is called from PROCESS_ QUEUE_RECORD once a queue record is formed from `FM_RIB_RECEIVING_ MFQUEUE`, `FM_RIB_STG_RECEIVING_DETAIL`, `FM_RIB_STG_RECEIVING_HEADER` tables. This is just to not pick the same record again.

Trigger Impact

Trigger Name
NA

Trigger File Name
NA

Table

NA

Message XSD

The file name that corresponds with the message type. Refer to the RIB documentation for this message type in order to get a detailed picture of the composition of the message.

Table 5–5

Message Types	Message Type Description	XML Schema Definition (XSD)
POScheduleCre	POSchedule Create message	POScheduleDesc.xsd

Table Impact

Table 5–6

TABLE	SELECT	INSERT	UPDATE	DELETE
FM_RIB_RECEIVING_MFQUEUE	Y	Y	Y	Y
FM_RIB_STG_RECEIVING_ HEADER	Y	Y	N	Y
FM_RIB_STG_RECEIVING_ DETAIL	Y	Y	N	Y
FM_RECEIVING_HEADER	Y	N	N	N
FM_RECEIVING_DETAIL	Y	N	N	N

Design Assumptions

This message is applicable only if ORFM exists for Brazil localization. It is applicable for PO/Schedule publication to RWMS only.

6

Archiving and Purging Data

In a production environment, the number of transactions increases over a period of time. To keep performance acceptable, data must be purged periodically from the active tables of the application. ORFM purges the data from the active tables and stores the data in history tables.

To enter a Nota Fiscal (for a PO, TO, or RTV), create a schedule in ORFM. After the schedule is in Financial Posted status and passes a certain number of days, all related data with that schedule are purged from the transaction tables. In addition, all the NFs that are inactive for a schedule are purged. An inactive schedule is also purged from the transaction table and the data is stored in the history table.

The history tables are also purged.

Table 6–1   History Table of fm_system_options

Variable	Variable_Type	Description
HISTORY_DAYS_ COMPLETED_NON_POs	NUMBER	Indicates the number of days for completed documents other than POs, when NF status is FP (Financial Posted).
HISTORY_DAYS_ COMPLETED_PO	NUMBER	Indicates the number of days for the completed POs, when NF status is FP (Financial Posted).
HISTORY_DAYS_ DELETED_DOCS	NUMBER	Indicates the number of days for the deleted NF documents, when NF status is I (Inactive).
HISTORY_DAYS_HIST_ TABLES	NUMBER	Indicates the number of days the history table can hold the records.
HISTORY_DAYS_ WORKSHEET_STATUS	NUMBER	Indicates the number of days the NF can be in worksheet status.
HISTORY_DAYS_ PROCESSED_EDI	NUMBER	Indicates the number of days the EDI NFs can exist in the EDI tables before getting purged.

For more information on Purging, see Chapter 11, "Understanding Batch Processes".

7

Integrating with TaxWeb Tax Rules – Retail Tax Integration Layer

Brazil has complex tax and fiscal systems. In order to have a flexible format to fit any kind of tax scenario and also to make the tax rules setup part of a specialist solution, Oracle Retail has integrated its applications with third-party tax engines provided by Oracle partners. The ORFM module is integrated with the TaxWeb Tax Rules solution through the Oracle Retail Tax Integration Layer (RTIL).

RTIL acts as a connector that exposes a Retail Tax Data Model API for calculating tax, which is envisioned to be an abstraction over TaxWeb Tax Rules tax data model. RTIL hosts TaxWeb Tax Rules Adapter that is responsible for converting the Retail tax data model into the native language of a tax service provider, and communicating with a third-party tax solution service. RTIL supports integration of RMS and ORFM with TaxWeb Tax Rules. The ORFM module is integrated with the TaxWeb Tax Rules solution to obtain information about all Brazilian tax legislation, with a high level of exception treatments.

For all flows in Oracle Retail that need to have tax calculation, the integration layer is used to have all taxes applied from the TaxWeb Tax Rules, considering the input parameters.

The following processes in ORFM/RMS use the tax calculation integration:

- Inbound NF validation

- Outbound NF issuance

- PO tax breakdown

- Item creation

- Refresh tax

- Item fiscal reclassification

For each of these processes, the needed input information is sent to TaxWeb Tax Rules through RTIL. RTIL is responsible for transforming the request information from the Retail tax data model format to TaxWeb Tax Rules specific format, and vice versa for the response information.

This chapter covers the following topics:

- Integration Overview

- Retail Tax Data Model

- RTIL Architecture

- RTIL Configuration

- Input Parameters Data Mapping and Expected Output Results

- Concurrent Processing in RTIL

- Tax. Calculation Scenarios

Integration Overview

RTIL is responsible for the TaxWeb Tax Rules API call with the necessary formatting. The TaxWeb Tax Rules API exposed has a native Nota Fiscal format, and formatting must be sent in that way for tax calculation in TaxWeb Tax Rules.

In TaxWeb Tax Rules, an integration layer is used for data completion. Some of the attributes in the API must be determined by Tax Rules (such as CFOP). Only after this step is complete the tax classification is triggered internally in TaxWeb Tax Rules. Once the taxes are classified, the API with the returned values is sent back to the integration layer in Oracle Retail.

The layout of TaxWeb Tax Rules solution is based on the NF layout and has the following structure:

- Entities (foundation data and fiscal attributes of each entity of a NF, such as Issuer, Addressee, and Transporter)

- NF header information

- NF item information

- Tax classification (enquadramento)

Figure 7–1   Oracle Retail/TaxWeb Tax Rules Integration
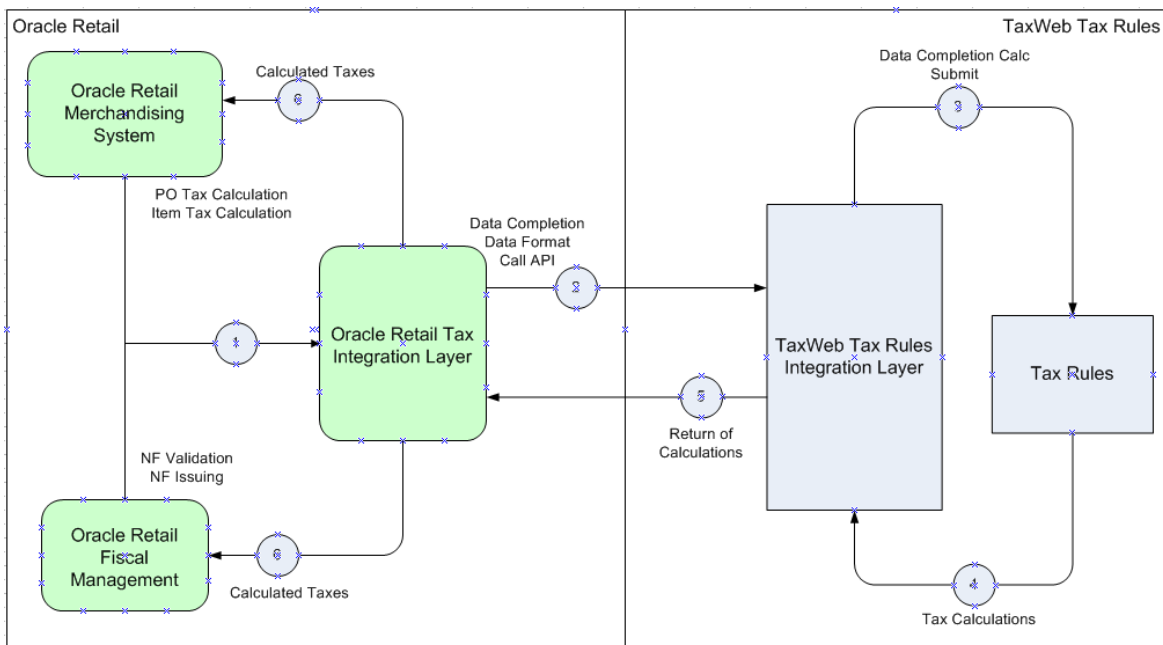
Retail Tax Data Model

The Retail Tax Data model is an abstraction introduced over the TaxWeb Tax Rules Tax Data model with the aim of achieving semantic interoperability. The Retail Tax Data Model aims to decouple Oracle Retail Products (RMS, ORFM) from the vendor

specific data model. This model enables RMS/ORFM to deal with a more generic format than being tightly locked with TaxWeb Tax Rules Vendor data model. This model allows applications to be more loosely coupled.

Figure 7–2   Retail Data Model Structure
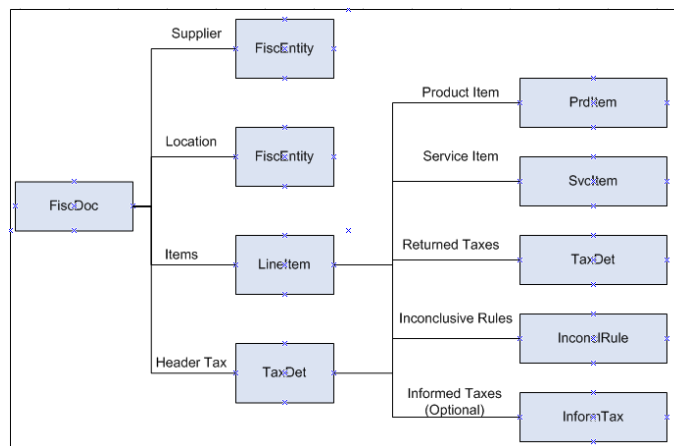
Object Structure Overview

- FiscDoc – This object represents a fiscal document and eventually maps to the Nota Fiscal of TaxWeb Tax Rules in the integration with TaxWeb Tax Rules. It has the information related to the NF header and it has included the other objects that detail each part of the fiscal document as the issuer, addressee, and items.

- FiscEntity – This object has the fiscal and master information related to the entities in a fiscal document. This object will be used to detail the issuer/supplier, addressee/location and facilitator/transporter information.

- LineItem – This object represents the detail of a line item in a fiscal document. It has included the taxes at item level.

- TaxDet – This object represents the details of an individual tax. It can be attached at the Fiscal Document level or at a DocLineItem Level. Each tax applied to an item will have a different object of this type.

- InformTax – This object represents the details of an individual tax informed by the calling application and supplied in the request. This indicates that TaxWeb Tax Rules should not compute this specific tax, but can use this information to derive other taxes if they are dependent on the informed tax. This is an optional entity which is sent in scenarios where in the NF tax is assumed to be the correct one. It can be attached at the Fiscal Document level or at a LineItem level.

- InconclRule – This object has the rules and laws applied to each tax. It has the regulation and the log of the tax classification. It notifies the rules that are not applied to the item and classified as inconclusive rules. This can be used as an information log to the user to show potential tax setup problems.

- ProductItem/ServiceITem – Code and description of items which can be products or services. These objects will have the fiscal attribute details for the items depending on the item type (product or service).

- FiscDocChnk – This is a placeholder object introduced to support a logical unit of work (contains one or more fiscal documents) that can be sent in a single request to TaxWeb Tax Rules. It does not have a direct counterpart in the TaxWeb Tax

Rules data model but aggregates to a collection of DocFiscal objects in the vendor data model. A collection of FiscDocChnk objects helps in enabling RTIL to support splitting a large request into multiple chunks and process them concurrently to enhance performance and throughput. Aggregation of the responses from TaxWeb Tax Rules are handled internally in RTIL.

Retail Tax Data Model to TaxWeb Tax Rules Data Model Object Mapping

The following table illustrates the mapping of the Retail Tax Data Model to TaxWeb Tax Rules at an object level.

Table 7–1   Mapping of Retail Tax Data to TaxWeb Tax Rules

Retail Tax Object	TaxWeb Tax Rules Object
FiscDoc	DocFiscal
FiscEntity	Pessoa
LineItem	ItemDocFiscal
TaxDet	Enquadramento
InconclRule	EnquadramentoItem
ProductItem	Produto
InformTax	Enquadramento
ServiceItem	Servico

RTIL Architecture

RTIL exposes a servlet controller to receive the HTTP request from ORFM/RMS application in a synchronous request/response pattern. Based on the http request from ORFM/RMS, RTIL will fetch the data from RMS/ORFM and form the request for external tax vendor. RTIL will invoke the associated external third party services with native POJO call to get the tax details which will be passed back to the calling application.

Retail Tax Integration Layer will be implemented as Java EE application with the associated tax service provider adapters. This layer will form the conduit between the Oracle Retail Applications and the Tax service provider. The Retail Tax Integration layer is responsible for assembling and disassembling of the vendor specific data model to Retail tax data model based on the configuration. The Retail Tax Integration layer is envisioned to host vendor specific connectors which can communicate to the external third party services. The subscribing application is not aware of the tax service provider, RTIL acts as a bridge between the subscribing application and third party tax service provider. In the current release, RTIL will host a TaxWeb Tax Rules adapter and interacts with TaxWeb Tax Rules using a POJO interface.

High Level Integration View

The integration layer in Oracle Retail does the following.

- Exposes a more abstracted tax data format (Retail Tax Data Model) for RMS/ORFM to pass their transaction details that require taxes to be computed

- Performs the data transformation from canonical to TaxWeb Tax Rules specific format

- Invokes the taxation API of TaxWeb Tax Rules

- Transforms the response containing taxes from TaxWeb Tax Rules specific format back to the canonical format to be consumed by RMS/ORFM

- RTIL is used as the unique tax calculation entry point

Figure 7–3   RTIL High Level Integration (update the SS using MS visio)
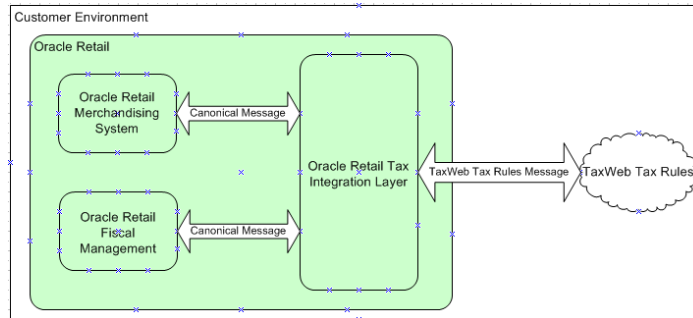
RTIL Integration Architecture

Figure 7–4   RTIL Integration Architecture (update the SS using MS visio)
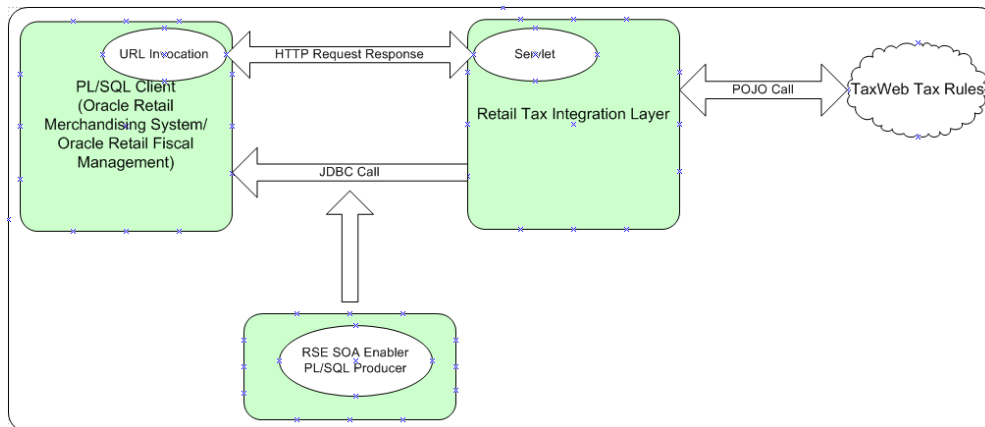
The above diagram illustrates the high level components and its interactions along with the run time and the design time view of the system. The Retail Service Enabler Utility is used to generate the necessary PL/SQL producer and its associated artifacts to fetch/set tax staged data from the database.

The client uses an URL Invoker class signaling RTIL to fetch the tax request data from the client's staged tables and TaxWeb Tax Rules API will be invoked as a POJO (Plain Old Java Object) call from RTIL.

Components Deployment View

Figure 7–5   Components Deployment View
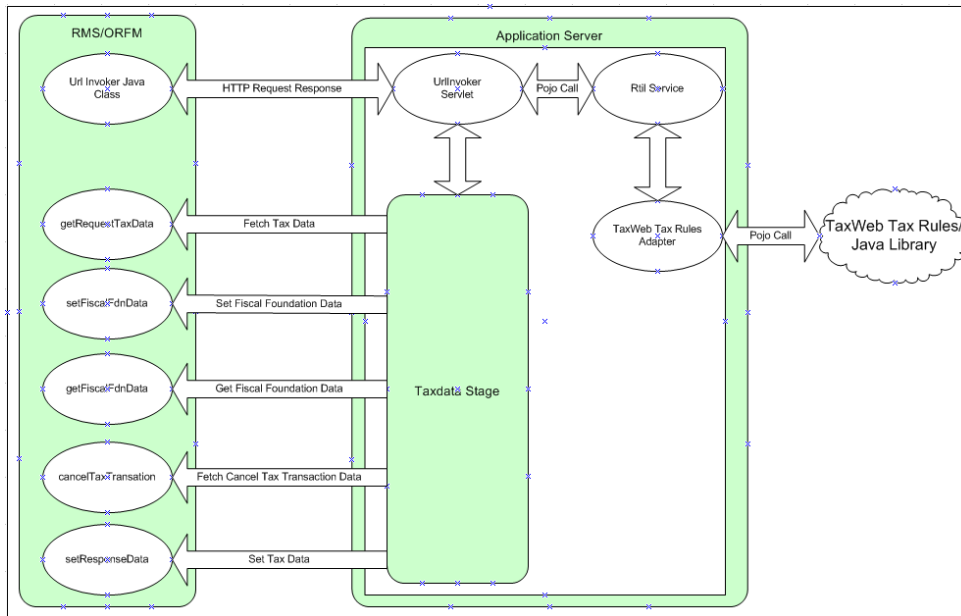
The preceding diagram illustrates the deployment view of the components. The salient features of the architecture are:

■ Retail applications interact only with tax services in an abstracted tax data format using RTIL-supplied client components.

■ RTIL insulates the Oracle Retail applications from tax service providers. RTIL will be deployed as an enterprise application instance (EAR file) in Oracle WebLogic Application Server.

■ RTIL will host the vendor adapters.

■ The adapters interact with the tax service providers in native format.

■ The adapters are responsible for converting the Retail Tax Data Model to the application-specific data model and vice versa.

■ Tax service providers can be deployed on the same or a different application server, based on customer needs.

■ The flow typically involves RTIL being invoked by a HTTP request which contains the qualifying information to invoke the appropriate TaxWeb Tax Rules API.

■ The primary group of operations supported are:

– Fetch Tax Staged Data from RMS/ORFM and invoke TaxWeb Tax Rules calculation API through a POJO call

– Update the Tax Response on to the Staged Tables in RMS/ORFM

– Support invocation of TaxWeb Tax Rules tax fiscal foundation data and updation of foundation response back to RMS/ORFM

– Invocation of a cancel Tax Transaction API on TaxWeb Tax Rules (This is a compensation API to be invoked to eliminate stale data in TaxWeb Tax Rules

in case of failures of retail transactions in RMS/ORFM post TaxWeb Tax Rules data updates).

Client Components

URL Invoker – This is a java class loaded into the database JVM, which triggers RTIL to fetch the Tax request data from the RMS/ORFM staged tables. It's a synchronous call, which is blocked till RTIL updates the tax response from TaxWeb Tax Rules back on to the RMS/ORFM staged tables.

It will be the responsibility of the URL Invoker to send a unique key to RTIL, which will be associated with the tax request and tax response that RTIL has to fetch and set. Also the differentiator parameters for the tax call or to fetch different foundation data or to invoke tax compensation API has to be sent by the invoker to trigger appropriate processing in RTIL.

Retail Tax API – This is a TaxAPI which is in a more abstracted format that is exposed by RTIL to RMS/RFM. Any data enrichment if required should be done prior to the invocation of this API Call. This API will be in the format of the Oracle Retail application implementation.

Since RMS/ORFM are Forms/PLSQL applications, the Oracle Object Types representing abstracted Tax Request and Response are generated from Functional Artifact Generator tool for the given RTIL Tax XSD's. This ensures the API is in the implementation format of the Oracle Retail application.

Based on the request parameters from the URL Invoker, appropriate tax API will be called from RTIL.

TaxData Stage component, PL/SQL packages –This is generated by RSE and provides the runtime support for fetching/setting tax data from/to the database staged tables.

RTIL Configuration

RTIL broadly supports the following configuration features:

- Logging/auditing
- User-friendly exception messages

Logging/Audit

Logging is implemented in a declarative manner using Spring Aspect Oriented Programming (AOP). AOP is a paradigm that addresses separation of concerns, thereby ensuring that logging and business logic are kept separate in the code, avoiding code cluttering. Logging is introduced during run time in a noninvasive manner leveraging spring AOP features. The logging utilities are implemented using the industry-standard Apache commons logging API.

The following are the classes that act as the logging aspect to the application and apply to different methods of the application as configured.

- LoggerAspect
- ObjectLoggerAspect
- TimerLoggerAspect

LoggerAspect

This class logs entry and exit statements for the methods in core RTIL classes. This also logs if there is some exception during the execution of these methods.

ObjectLoggerAspect

This class logs the entire content of request and response objects used in the transactions. It also logs the request object content and response object contents both in the canonical and native TaxWeb Tax Rules format. This is useful for troubleshooting the request and responses.

TimerLoggerAspect

This logs the time taken in milliseconds for execution of the methods present in assembler, Gateway, and adapter classes. This aspect can be used to obtain a timing profile of the method executions.

These aspects are declared by annotations and registered in applicationContext.xml.

The following is a snapshot of the applicationContext.xml configuration file which shows the configuration for registering Aspects.

```
<aop:aspectj-autoproxy />
<bean id="timingLoggerAspect"
class="com.oracle.retail.tax.aspects.TimingLoggerAspect" />
<bean id="loggerAspect" class="com.oracle.retail.tax.aspects.LoggerAspect" />
<bean id="objectLoggerAspect"
class="com.oracle.retail.tax.aspects.ObjectLoggerAspect" />
```

> **Note:** Removal of these entries in this configuration file is equivalent to removing the code which logs these details. Thus, the enablement/disablement of logging is controlled in a declarative manner using Spring AOP.

A second level of control is available in the log4j.properties file. This file controls the following logging features whose configuration can be changed to suit customer needs:

- Logging at different granularities (INFO, ERROR, DEBUG, FATAL)

- File size limit for each log file (log4j.appender.LOGFILE.MaxFileSize)

- Number of log files to be retained during logfile rollover (log4j.appender.LOGFILE.MaxBackupIndex)

The following is a snapshot of the log4j.properties configuration file supplied with default options during deployment time.

```
# Root Level Logger
log4j.rootLogger=INFO,STDOUT

log4j.appender.STDOUT=org.apache.log4j.ConsoleAppender
log4j.appender.STDOUT.layout=org.apache.log4j.PatternLayout
log4j.appender.STDOUT.layout.ConversionPattern=%d %p [%c] - %m%n

log4j.appender.LOGFILE=org.apache.log4j.RollingFileAppender
log4j.appender.LOGFILE.MaxFileSize=5MB
log4j.appender.LOGFILE.File=./log/rtil.log
# Keep ten backup files.
log4j.appender.LOGFILE.MaxBackupIndex=30
```

```
# Pattern to output: date priority [category] - message
log4j.appender.LOGFILE.layout=org.apache.log4j.PatternLayout
log4j.appender.LOGFILE.layout.ConversionPattern=%d %p [%c] - %m%n

#Class Level logger
log4j.category.org.dozer.MappingProcessor=INFO,STDOUT,appender
log4j.category.com.oracle.retail.tax.aspects.TimingLoggerAspect=DEBUG,LOGFILE
log4j.category.com.oracle.retail.tax.aspects.ObjectLoggerAspect=DEBUG,LOGFILE
log4j.category.com.oracle.retail.tax.aspects.LoggerAspect=DEBUG,LOGFILE

log4j.rootCategory= ALL,LOGFILE,STDOUT
```

User Friendly Exception Messages

RTIL provides some flexibility in configuring user-friendly error messages based on the exception conditions encountered. The ExceptionMessage.properties file contains the list of system-generated exception messages in RTIL, which can be further mapped to the user messages required by the customer. The file is a typical properties file containing key value pairs. This file is read by RTIL infrastructure to substitute the system generated message pattern (Key) with the supplied user message (value), and the user message is propagated to the calling infrastructure (ORFM/RMS).

> **Note:** The configuration option provided is only for supplying/altering the user defined messages. This file does not support the addition/modification of system defined messages.

Input Parameters Data Mapping and Expected Output Results

ORFM calls the Tax Rules API for the expected scenario and considering the structure of the API, which is similar to a fiscal document. All the input parameters are formatted in that structure, and the returned values are also sent in the same format. The input parameters do not contain any tax information. Only the fiscal attributes and values are sent. In the output parameters, the tax calculation results are sent back based on the original values in the same fiscal document format.

Figure 7–6   Input x Output Parameters (update the ss with MS visio)
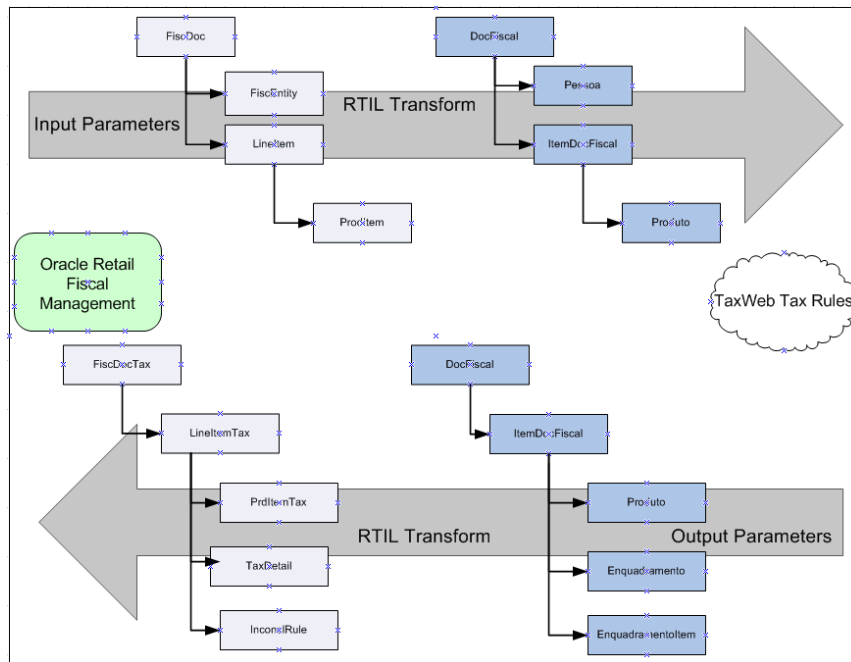
Concurrent Processing in RTIL

RTIL being a J2EE application and deployed in WebLogic application server inherently supports parallel processing of requests, provided its clients (RMS/ORFM) can send the requests concurrently. However, within the context of a single request, the processing is will be from RMS/ORFM to RTIL and then to TaxWeb Tax Rules**. In the event of a single request being very voluminous, this could result in slower throughput as it will be processed sequentially. Typical example of a large request could be a set of items ranged to large number of locations needing a taxation input.

To optimize the handling of large requests, threading is introduced in RTIL to enable concurrent processing. The approach of concurrent processing in RTIL is as follows:

- A large request is partitioned into a set of logical units of work called chunks.

- Each chunk is capable of being independently processed and the chunk is represented as a FisDocChnk object.

- Client (RMS/ORFM) partitions the large request into chunks and invokes RTIL.

- RTIL parses the requests and spawns the necessary threads to process the chunks individually.

- Concurrent requests are sent to TaxWeb Tax Rules with the individual chunks and responses are received**.

- Responses are tagged accordingly to support aggregation and sent back to the calling client (RMS/ORFM).

RMS/ORFM can trigger the concurrent processing in RTIL by setting a boolean flag called "Thread_use" appropriately to Y/N which will be passed in the request. The behavior is controlled at the request level intentionally so that clients have the flexibility of optionally switching on concurrent processing in RTIL only for large request which merit concurrent processing for optimal throughput.

RTIL uses WebLogic WorkManager framework for thread management, which provides the following benefits:

- Provides a concurrent programming API for EJBs/Servlets in a J2EE context.

- Provides a container managed alternative to java Thread API thereby enabling the container full visibility and control over the executing threads.

- Container based thread pooling and management.

- Standards based implementation

The configuration required for the work manager is detailed in the RTIL configuration and operational considerations section.

RTIL Concurrent Flow Initiation

Figure 7–7 depicts how RTIL concurrent processing is triggered based on the Thread_ Use flag value contained in the RMS request.

Figure 7–7   RTIL Concurrent Flow Initiation
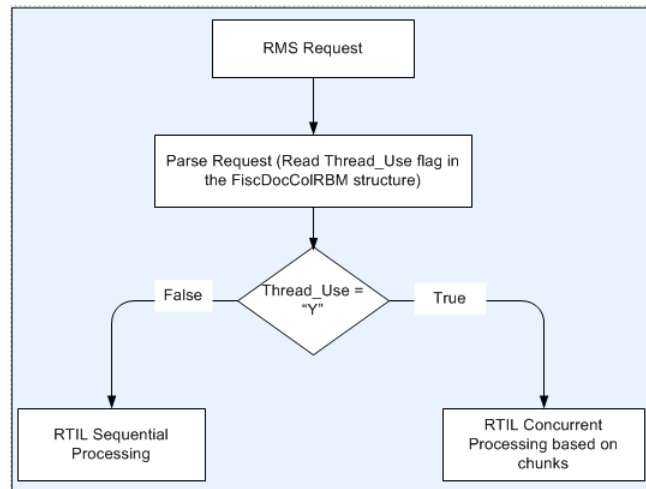
RTIL Concurrent Request Processing Flow

Figure 7–8 depicts handling of a large RMS request in RTIL in a threaded fashion. The steps for handling large RMS request in a threaded fashion is as follows:

1. The RMS request contains the logical chunks (FiscDocChunk[n]).

2. RTIL splits it into individual chunks.

3. Each chunk is fired in parallel as an independent tax calculation request to TaxWeb Tax Rules.

4. The responses are collected from TaxWeb Tax Rules and tagged accordingly to support consolidation of the responses into a single aggregated response in RMS.

Figure 7–8   RTIL Concurrent Request Processing Flow
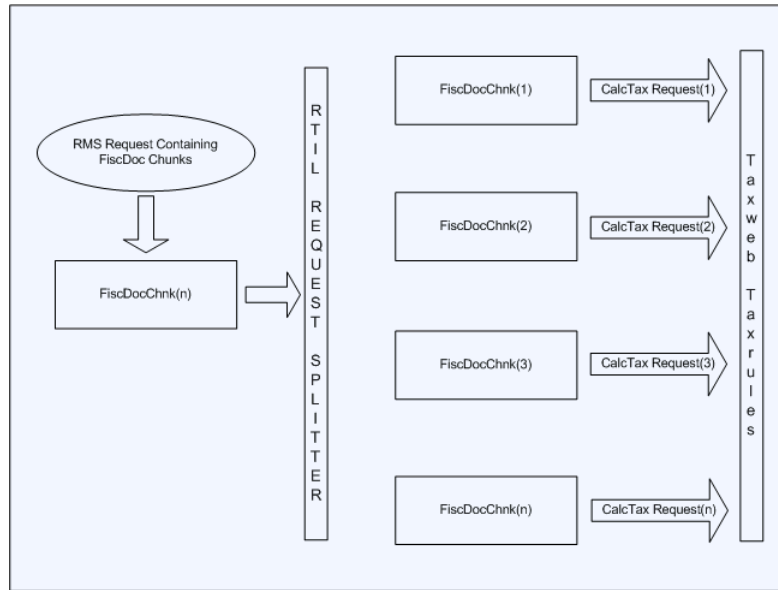
Tax. Calculation Scenarios

For all the scenarios, the input parameters are based on the Retail Tax API objects. Only the tax objects are not used, and the detail data-mapping for the input parameters can change depending on each scenario.

Scenario 1 - Inbound NF through User Entry

Enter the Inbound PO NF through the ORFM user interface. Once the NF is entered, tax call is initiated. The taxes returned by the external tax engine is compared against the informed taxes for any discrepancy.

Scenario 2 - Inbound PO NFs through EDI

NFs from the supplier could be interfaced to RFM through EDI functionality. These NFs once generated also make a tax call through the batch provided. The taxes returned by the external tax engine is compared against the informed taxes for any discrepancy.

Scenario 3 - PO Receiving

Tax calls are made from the RFM application during the receiving process based on the Resolution type on the Discrepancies identified during Validation of a NF.

Scenario 4 - Outbound NF Issuing

For outbound transactions like Transfer, RTV and Inventory Adjustments initiated by the Retailer, ORFM generates the NFs automatically. These NFs generated makes the tax calls and the taxes are printed on the document before approval and inventory movement.

Scenario 4 - PO Tax Calculation

The PO tax breakdown is the calculation of all applicable taxes of a PO. The PO information is similar to a fiscal document, and it is similar to the NF issuing process. The PO information utilizes the API format to have the tax calculated.

Scenario 5 - Item Creation

During the item creation process, the sales taxes applicable to each location where the item is linked are calculated. The item creation process also makes a call to the tax API to get the sales tax rates. The same concept is applied to the default purchase taxes used for margin calculation.

Scenarios such as primary supplier change and fiscal reclassification call the same tax integration defined for original item creation process.

8

Working with RTIL

This chapter discusses the common issues that you have to consider as you progress towards a production environment involving the Oracle Retail Tax Integration Layer (RTIL). It is not a comprehensive list since they are very dependent on the retailer implementation.

Operational Considerations

The operational issues that are to be considered when using RTIL are detailed in the following sections:

- RTIL Log File Monitoring
- Log File Archive and Purge
- Time-out Configurations
- RTIL WorkManager Configurations
- General Troubleshooting for RTIL/TaxWeb Tax Rules
- Recovery in case of RTIL Compensation API failure
- RTIL and TaxWeb Tax Rules Outage Sequence/Start and Stop
- Probable Systemic Failure Scenarios in Tax Flows and Recovery
- Deploying RTIL

RTIL Log File Monitoring

Since RTIL is an integration subsystem that runs with no console, it is important to monitor the application log files that are created. The log files have to be monitored for the content (looking for exceptions), and also the size and growth of the exceptions. RTIL logs are created in the $WEBLOGIC_DOMAIN/log folder in the domain where RTIL is deployed.

The log files that are to be considered are:

- rtil.log – This is the primary log file which captures the RTIL server interactions with external tax engine (TaxWeb Tax Rules). All the exceptions/failures that occur in the tax engine and RTIL interactions are logged in this file.
- taxdatastage_services.log – This log file captures RTIL interactions with RMS/ORFM application. All the exceptions/failures that happen during these interactions are logged in this file.

Log File Archive and Purge

RTIL uses log4j for all of its logging control. It manages the logs size through its control file (log4j.properties), auto archive, and purge. This configuration file is embedded in the rtil-service.ear file in the lib\rtil-config.jar folder. This configuration file controls the behavior of logging in the rtil.log file. The file is configurable through the installer at install time. The file can also be manually modified for any changes that have to be made after the installation.

> **Note:** Refer to the Apache Software Foundation http://logging.apache.org/log4j/1.2/manual.html for more details on log4j.

Below is a snapshot of the RTIL log4j.properties

```
log4j.appender.LOGFILE=org.apache.log4j.RollingFileAppender
log4j.appender.LOGFILE.MaxFileSize=5MB
log4j.appender.LOGFILE.File=./log/rtil.log
# Keep thirty backup files.
log4j.appender.LOGFILE.MaxBackupIndex=30
# Pattern to output: date priority [category] - message
log4j.appender.LOGFILE.layout=org.apache.log4j.PatternLayout
log4j.appender.LOGFILE.layout.ConversionPattern=%d %p [%c] - %m%n
```

The property **log4j.appender.LOGFILE.MaxFileSize** determines the maximum size of the log file before being rolled over. It is recommended to have a optimal size of not more than 5 MB. If the size of this file increases, it may lead to performance issues.

The property **log4j.appender.LOGFILE.MaxBackupIndex** depicts the maximum number of files to be archived in the $WEBLOGIC_DOMAIN/log folder. This value has to be based on the retailers' needs, the archival strategies, and the backup strategies.

Time-out Configurations

Retail tax transactions involves interactions between multiple applications like the Oracle Retail Integration Bus (RIB), RMS, ORFM, any Store Inventory Management system, any Warehouse Management system, RTIL and TaxWeb Tax Rules. Since there are multiple applications involved, they need to be handled very carefully in case of systemic failure in one or more applications.

The time-outs have to be configured diligently based on retailer's needs, volume of data, processing hardware etc. The time-outs have to ensure proper exits in case of slow responsive applications or systemic failures in applications.

A typical example of the retail tax transaction is as follows:

Inventory Adjustment

An inventory adjustment retail transaction comprises of the following interactions between RIB, RMS, ORFM, RTIL and TaxWeb Tax Rules:

- Publication of Inventory Adjustment RIB message from any warehouse management system.

- RMS consumes the RIB message through the subscription API and routes it to ORFM.

- ORFM generates the Nota Fiscal (NF).

- During NF validation, ORFM invokes a synchronous tax call to TaxWeb Tax Rules through RTIL.

- ORFM inventory is updated and the response is sent back to RIB.

Figure 8–1   Inventory Adjustment Business Flow
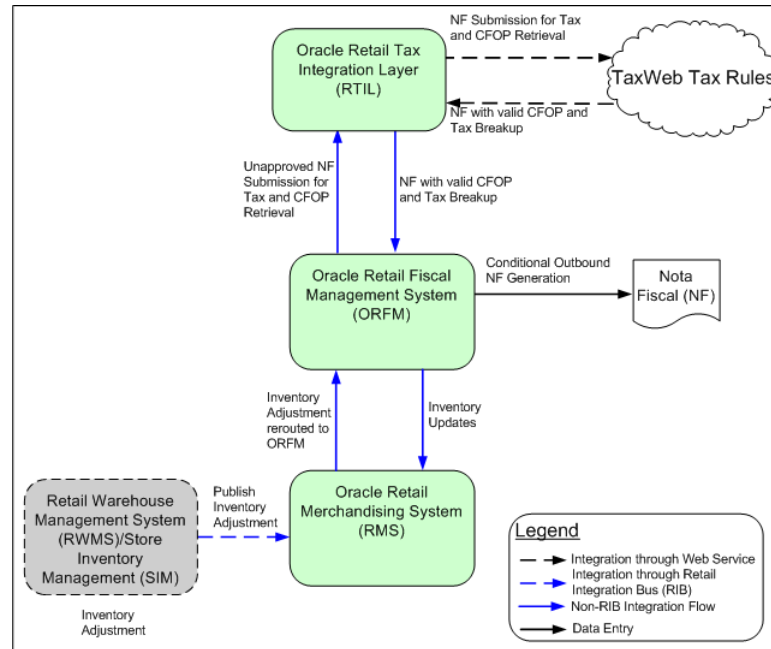
To ensure that the retail transactions are not adversely impacted in case of systemic failures in one or more applications, the time-outs has to be configured appropriately to have a exit path.

The general rule of thumb is that the time out configurations progressively decrease in a call sequence flow as we move from the top layer to the lowest layer, and have to be configured in the following subsystems:

- WebLogic Server

- RMS

- RIB

URLInvoker Time-out

This time-out value is configured in the RMS table (RETAIL_SERVICE_REPORT_ TABLE) for the record (RS_CODE=RTIL) and passed on to the RTIL client component (URLInvoker) when triggering a request to the RTIL server. Note the entry configured in the table is in the units of milliseconds. If the time-out value is not configured, then URLInvoker is set to a default value of 180 seconds.

This time-out caters to RTIL client component and ensures that the client component throws back an error response on expiry of the time-out in RTIL client or if the RTIL server fails to respond within the configured threshold.

RIB Transaction Time-out

The RIB transactions are also configured with a time-out threshold value to ensure that RIB transactions do not wait forever.

While setting up the test environment, the following time-out values are configured:

- URLInvoker time-out – 1000 seconds

- RIB time-out – 1200 seconds

If any of the subsystems are responding slowly outside the range of its configured thresholds, time-outs are triggered. Eventually, these are sent back to RIB with an error status, which causes RIB to rollback the transaction and deliver it to the error hospital or the adapter will shut down based on the nature of the error.

> **Note:** The time-out configurations specified are for indicative purpose only. They have to be customized based on the retailers' platform, environment, transactional volume and hardware.

RTIL WorkManager Configurations

RTIL has a workmanager called "RTILWorkMgr" to support concurrent processing in RTIL for voluminous requests. The concurrent processing behavior is triggered in RTIL based on flag contained in its client's (RMS/ORFM) request.

Since RTIL uses WebLogic WorkManager, the following configurations depicted below can be modified/tuned in the WebLogic administrator console by a WebLogic Administrator:

- Maximum Threads Constraint

- Capacity Constraint

Maximum Threads Constraint

This parameter depicts the maximum number of concurrent threads that can be allocated to execute requests. The default is set to 16. Typically, it is recommended to set this value based on the number of processors available in the system in which RTIL is deployed.

Capacity Constraint

This parameter depicts the maximum limit of the requests that can be queued or executed. The default is set to 20000. After this, the WebLogic Server rejects requests.

> **Note:** The work manager configurations specified are for indicative purpose only. They have to be customized based on the retailers' platform, environment, transactional volume, and hardware.

> **Note:** In case the WebLogic Administrator wants to modify the workmanager configurations, the details are provided in Understanding RTIL WorkManager Settings.

General Troubleshooting for RTIL/TaxWeb Tax Rules

This section discusses some of the errors (systemic or configuration related) that can occur in RTIL and its integration with TaxWeb Tax Rules, and also RTIL integration with RMS/ORFM and consequently RIB. This section provides an overview of the errors, the cause and the corrective action that has to be taken. Some of the errors includes:

- RMS/ORFM is configured with an incorrect RTIL URL (Service Access Point)
- RTIL Service is not responding
- RTIL URL Configuration is correct but insufficient privileges in RMS database for RMS/ORFM to access RTIL
- Incomplete Deployment of TaxWeb Tax Rules libraries
- How do I know if RTIL is installed correctly?

RMS/ORFM is configured with an incorrect RTIL URL (Service Access Point)

Error Message – In case the RTIL URL is incorrectly configured (format is wrong, hostname or port number is incorrect), any RMS/ORFM request to RTIL results in an error in the RTIL client component (URLInvoker hosted in RMS database schema), and the error message returned to the RMS is the incorrect URL format. The Example 8–1 provides the incorrect URL format, with the incorrect part highlighted in **bold**.

Example 8–1   Incorrect URL

```
http://mspdv310.us.oracle.com:17069/rtil-web/invokeApp123?invocationType=CALC_
TAX&invocationKey=1234
```

Cause – The URL returned as depicted in the error message indicates that the RTIL URL is incorrectly configured in RMS.

Resolution – Update the URL column in the retail_service_report_url table with correct RTIL URL configuration as shown in Example 8–2. The correction is made to the portion that is highlighted in bold in Example 8–1.

Example 8–2   Correct URL

```
http://mspdv310.us.oracle.com:17069/rtil-web/invokeApp
```

RTIL Service is not responding

Error Message – RTIL responds with the error message "Connection Refused" for any RMS/ORFM request to RTIL.

Cause – This error is typically thrown when the RTIL service is down.

Resolution – Restart the RTIL application instance from the WebLogic Administration Console. You must have WebLogic Admin privileges for this operation. In addition, an administrator has to investigate the WebLogic admin logs to ascertain the reason for the failure of the service before restarting it.

RTIL URL Configuration is correct but insufficient privileges in RMS database for RMS/ORFM to access RTIL

Error Message – In ORA-00942: table or view does not exist ORA-06512: at "SYS.DBMS_PICKLER", line 18ORA-06512: at "SYS.DBMS_PICKLER", line 58ORA-06512: at line 1

Resolution – Usually occurs because the user used in the connection point does not have sufficient privileges to the Package or Objects being referred to in the application. Either change the user that is being used or make sure proper permissions and synonyms are created in Oracle.

Incomplete Deployment of TaxWeb Tax Rules libraries

Error Message – Unknown Error occurred. Please check the log file for more details

The error in the RTIL logfile is provided in

Example 8–3   RTIL Log File

```
java.lang.ClassNotFoundException: com.mcfox.tax.management.TaxRulesAPI
at
weblogic.utils.classloaders.GenericClassLoader.findLocalClass(GenericClassLoader.j
ava:280)
at
weblogic.utils.classloaders.GenericClassLoader.findClass(GenericClassLoader.java:2
53)
at java.lang.ClassLoader.loadClass(ClassLoader.java:303)
at java.lang.ClassLoader.loadClass(ClassLoader.java:248)
at
weblogic.utils.classloaders.GenericClassLoader.loadClass(GenericClassLoader.java:1
77)
at java.lang.Class.forName0(Native Method)
at java.lang.Class.forName(Class.java:169)
at
com.oracle.retail.tax.servicegateway.MastersafPojoServiceGateway.init(Unknown
Source)
at sun.reflect.NativeMethodAccessorImpl.invoke0(Native Method)
at sun.reflect.NativeMethodAccessorImpl.invoke(NativeMethodAccessorImpl.java:39)
at
sun.reflect.DelegatingMethodAccessorImpl.invoke(DelegatingMethodAccessorImpl.java:
25)
at java.lang.reflect.Method.invoke(Method.java:597)
```

Cause – This error typically occurs if the TaxWeb Tax Rules required libraries are not deployed in the $WEBLOGIC_DOMAN/lib folder of RTIL instance. The error message given is for reference only and will vary based on the missing library.

Correction – Refer to the TaxWeb Tax Rules Installation Guide and ensure that all the libraries (jar files) are made available in the $WEBLOGIC_DOMAIN/lib folder of the deployed RTIL instance. Before transferring the missing files, ensure that the RTIL application instance is shutdown from WebLogic admin console.

How do I know if RTIL is installed correctly?

The following two steps are recommended for a quick sanity test to check if RTIL is deployed correctly and ready to serve requests:

- Technical Ping
- Application Ping

Technical Ping

This step is to ascertain if the RTIL service is running. Open a browser and enter the deployed RTIL URL in the address bar of the browser.

Expected Result – Enter the deployed RTIL URL in address bar of a browser http://<hostname>:<portnum>/rtil-web/invokeApp (For example http://mspdv310.us.oracle.com:17069/rtil-web/invokeApp). If the response in the browser window is "E|invocationKey not present", this implies RTIL service is running.

Application Ping

This step is to ascertain if RTIL is correctly deployed and the configuration to RMS/ORFM is valid.

To sanity test an RTIL deployment, run the stored procedure provided in Example 8–4 in an Oracle database schema in which RMS localized version has been deployed. Ensure that the arguments are replaced in the invokeURL call in the procedure to reflect the URL where RTIL is deployed.

Example 8–4   Procedure

```
Rem
Rem This SQL Script executes the Java Stored Procedure
Rem
SET SERVEROUTPUT ON
CALL dbms_java.set_output(2000);
DECLARE
  o_status  VARCHAR2(255);
  o_error VARCHAR2(255);
BEGIN
--EX : invokeUrl('http://mspdv309.us.oracle.com:7003/rtil-web/invokeApp','CALC_
TAX', '1234', '180000',o_status, o_error);
  invokeUrl('<url>',<invocationType>', '<keyId>', '<timeout>',o_status, o_error);
  DBMS_OUTPUT.PUT_LINE('status ' || o_status);
  DBMS_OUTPUT.PUT_LINE('status ' || o_error);
END;
/
```

url = url where RTIL is deployed

(For example, 'http://mspdv309.us.oracle.com:7003/rtil-web/invokeApp')

invocationType = CALC_TAX(for tax calls)

keyed = 1234 (a dummy key id)

time-out = 1800(a dummy time-out value)

Expected Result – 0L10N_ROUTE_INFO_NOT_FOUND@11234 – This would imply that RTIL is deployed correctly and also the configuration from RTIL to RMS/ORFM is valid.

Recovery in case of RTIL Compensation API failure

Most of the retail tax transactions which have a taxation requirement are plain compute operations in TaxWeb Tax Rules without any data persistence. So failure in RTIL/RMS/ORFM post tax computations does not mandate a data rollback operation within TaxWeb Tax Rules. For certain exclusive ORFM scenarios like triangulation and recovery of ST, TaxWeb Tax Rules performs persistence of data in its tables which mandate a rollback of data in case of failures in post processing.

For handling rolling back of persisted data in TaxWeb Tax Rules tables in the event of a post processing failure in ORFM, it invokes a cancelTax (a compensation API) on to TaxWeb Tax Rules which deals with explicit removal of persisted transaction data. In the rare event of failure of this compensation API, the failure will be logged in RTIL log file along with the data details. In such scenarios, a manual clean up of records is required. The following tables have to be cleaned:

- TAX_DOCFISCALHIST
- TAX_ENQHIST

- TAX_ENQITHIST
- TAX_PESSOAHIST
- TAX_REC_ST_ENTRADA
- TAX_REC_ST_SAIDA

> **Note:** Refer to the TaxWeb Tax Rules documentation for further details on the tables.

RTIL and TaxWeb Tax Rules Outage Sequence/Start and Stop

TaxWeb Tax Rules libraries have to be frequently updated since the content of TaxWeb Tax Rules changes due to incorporation of Brazil tax legislation changes.

RTIL, RMS, ORFM and RIB also would have patch upgrade requirements. To ensure smooth flow of operations at the customer environment, these subsystems have to be stopped and started in a particular order.

The recommended order for stopping the systems during a patch maintenance window is RIB, RMS/ORFM and RTIL and the start up sequence is in the reverse order. RIB is an aggregation of distributed components and refer to RIB Operations Guide, for further details on start and stop of RIB internal components. TaxWeb Tax Rules being a library of RTIL does not have an independent operational start and stop sequence of its own.

Probable Systemic Failure Scenarios in Tax Flows and Recovery

The following table provides the failure scenarios in tax flows:

Table 8–1   Systemic Failure Scenarios in Tax Flows

Event	Scenario	How is it handled
RibForRMS goes down	RTIL, TaxWeb Tax Rules computation is completed, and RMS/ORFM tables are updated	Data will not be read/used as RIB transaction would have rolled back
RTIL failure	RIB transaction has updated RMS tables and RTIL stops responding	URLInvoker in RMS will receive an error as it is a synchronous call, which will result in an "E" status back to RIB. RIB then initiates a rollback of the data updated in RMS
RTIL fails to respond (very long running transaction)	RIB transaction has updated RMS tables and RTIL does not respond	URLInvoker will time-out resulting in an "E" status back to RIB which triggers a rollback of updated data in RMS
TaxWeb Tax Rules business failure	RIB transaction has updated RMS tables and RTIL fails to fetch the tax as TaxWeb Tax Rules is unable to compute the tax due to a data issue in the request	RTIL returns an error to RMS and RMS sends back "E" status to RIB which triggers a rollback of updated data in RMS

Table 8–1 (Cont.) Systemic Failure Scenarios in Tax Flows

Event	Scenario	How is it handled
RTIL failure in updating RMS tables with tax results	RIB transaction has updated RMS tables with request data, RTIL successfully calls TaxWeb Tax Rules and gets the taxes and RTIL updates to RMS tables results in a failure	RTIL returns an error to URLInvoker and this results in RMS sending an "E" to RIB which triggers a rollback of updated data in RMS

TaxWeb Failures and Associated RIB Impacts

Some of the business flows will have a RIB XA transaction spawn across RIB, RMS, ORFM, RTIL, and TaxWeb. In such scenarios, the performance of a RIB transaction is inclusive and dependent on the participating downstream applications like RMS/ORFM/RTIL/TaxWeb. You need to be aware that any adverse behavior of TaxWeb due to poor performance in case of high volume, inherent bugs (if any), or any other issues can potentially result in RIB timeouts/in doubt transactions/RIB failures impacting the RIB adapters.

Deploying RTIL

TaxWeb's lazy load solution is recommended if RTIL is deployed on a memory constrained environment.

The TaxWeb solution has the following two flavors primarily in terms of the memory footprint it occupies:

- Static Load Version

- Lazy Load Version

The Static Load Version loads all the tax rules available in the content at initialization time and has a larger memory footprint. This version requires about 5 GB of Java Heap Size space for it to be functional.

The Lazy Load Version implements an on-demand load feature. This implies only the tax rules needed to satisfy a tax computation request is loaded on the fly into the TaxWeb engine when it receives a tax calculation request.

The Lazy Load version is provided as an alternative to address deployments where in there is a crunch for physical memory.

9

Understanding NF-e and SPED

This chapter provides details about NF-e and SPED and covers the following sections:

- Nota Fiscal Eletrônica (NF-e)

- NF-e Solution

- NF-e Publishing

- NF-e Subscription

- SPED File Structure

- SPED Solution

- Performance Optimization for SPED

> **Note:** ORFM 13.2.5 supports integration with fiscal partners, Mastersaf, and Synchro for NF-e and SPED.

Nota Fiscal Eletrônica (NF-e)

Nota Fiscal Eletrônica (NF-e) or Electronic Fiscal Note is a Brazilian government project with the objective of implementing a national model of electronic fiscal document to substitute the current system of issuing the fiscal documents in paper. The virtual document has juridical validity guaranteed by the digital signature of the issuer. It simplifies the fiscal obligations of the taxpayers and allows the follow-up of the commercial operations by the tax authority.

The NF-e issuer generates an electronic file with all NF information in a more detailed level than the regular NF. This file must be digitally signed to guarantee the integrity of the data and the authorship of the issuer.

The electronic file that corresponds to the NF-e is transmitted through the internet to the SEFAZ (Secretaria da Fazenda - Brazilian Tax Authority) of the origin state of the issuer. The SEFAZ provides a pre-validation of the file and returns a receiving protocol (Authorization for Use), that is necessary to the traffic of the goods.

To follow the goods, a graphic representation of the NF-e is printed. The Documento Auxiliar da Nota Fiscal Eletrônica-Auxiliary Document of the Electronic Invoice (DANFE) is printed in a common paper, one copy that highlights the access key for consultation of the NF-e in the internet and a bi-dimensional bar code which facilitates the capture and confirmation of information of the NF-e by the fiscal units.

The DANFE is not a Nota Fiscal, and does not replace the NF. It is just an auxiliary document for consultation of the NF-e. It has the access code of the NF-e which allows

its owner to confirm the real existence of the NF-e in RFB environment (Receita Federal Brasileira-Brazilian Federal Tax Authority) or the SEFAZ Web site.

NF-e Solution

The overall solution landscape considers that ORFM works with a third party solution for NF-e generation and transmission to the government.

Figure 9–1   NF-e Issuing Solution Approach
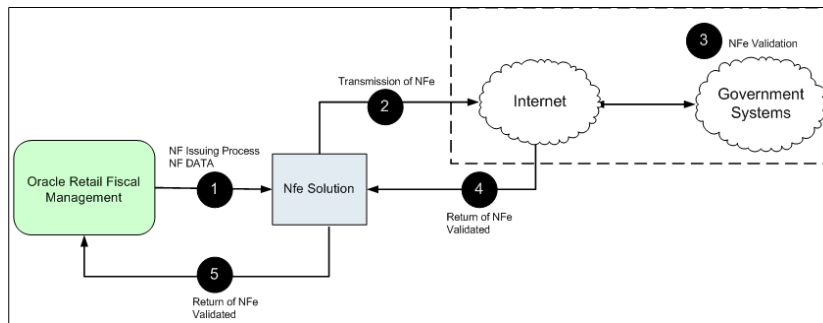
NF-e Options

ORFM allows the generation of the NF-e by location and utilization code. A system option is used to identify the location that issues the NF-e and for which transaction, such as transfers, inventory adjustment, and RTVs.

Considering that the transactions where the NF-e issuing is applicable, like transfers, are initiated in RWMS. Hence the utilization code associated to each location must be the same utilization code set in fm_system_options as default utilization for outbound transactions, such as transfers, intercompany transfers, and RTVs. The default utilization code is used by RWMS to generate the NFS. This behavior is not controlled in the system and must be defined by the user. These default utilization codes should be present in FM_NFE_CONFIG_LOC table for that location to ensure it is going under NF-e processing.

One location may use NF-e for transfers and not for RTVs depending on corresponding location/utilization setup in FM_NFE_CONFIG_LOC table. While NF creation, ORFM considers this parameterization in order to enter in the NF-e flow.

The document type in that case is auto-filled with the 55. That is the defined document type for NF-e. That can also be set up as a system variable, so no hard codes are placed.

NF-e Publishing

The FM_STG_NFE staging table for NF-e contains the fiscal doc identifier and its status for the Java Adapter to fetch those records, which are in NF-e Pending or NF-e Corrected or NF-e Canceled state and submit them to the integrated fiscal partner. The EVENT_ID field contains the sequence in which the NF-e messages are published to fiscal partners. The fiscal_doc_no, series_no, cnpj and justification fields are used to allow fiscal partner's NF-e product to interface the fiscal document information to SEFAZ without scanning through the object source again.

There are eight status codes to capture NF-e flow between ORFM, fiscal partner, and SEFAZ.

- NFe_P - Fresh NF-e document waiting to be picked by fiscal partner.

- NFe_X - Corrected NF-e document waiting to be picked by fiscal partner.

- NFe_C - Canceled NF-e document waiting to be picked by fiscal partner.

- C_A - fiscal partner updates the staging table with this status when NF-e is successfully canceled.

- N_A - fiscal partner updates the staging table with this status when NF-e is successfully nullified.

- NFe_I - Intermediate flag that is updated by fiscal partner while the document has been sent to SEFAZ, just in case of emission. A response to a cancel call is provided by SEFAZ immediately, whereas an emission situation does not provide an immediate call.

- A - Approved NF-e from SEFAZ.

- E - Erroneous NF-e from SEFAZ due to data/transmission errors (in any situation; emission, cancel or nullify).

NF-e Subscription

On the consumption side, ORFM writes the API and grants access to fiscal partner to load acknowledgments (approval or erroneous details) from SEFAZ into the respective ORFM tables.

When the NF-e gets approved successfully or processed with errors, fiscal partner makes a call to the ORFM packaged procedure FM_MS_NFE_SQL.CONSUME to update the NF-e details into the respective ORFM tables. If there are any errors then it inserts the NF-e transaction history table with error_id and error_description. The consume procedure contains the following parameters:

Table 9–1   Parameters

Variable Name	Input/Output Designation	Data Type/Length
O_status_code	IN OUT	NUMBER
O_error_message	IN OUT	VARCHAR2
I_message	IN	OBJ_MS_RFM_NFE_REC

I_message contains the following fields:

Table 9–2   I_Message Fields

OBJ_MS_RFM_NFE_REC	
Name	**Type**
fiscal_doc_id	number10
nfe_access_key	varchar244
nfe_protocol	number15
nfe_danfe_url	varchar21000
status	varchar26
errorDtl_tbl	OBJ_MS_RFM_ErrorDtl_TBL

Table 9–3   Message Fields

OBJ_MS_RFM_ErrorDtl_REC	
Name	**Type**
message_id	number4
message_desc	varchar21000
transaction_date	timestamp

OBJ_MS_RFM_ErrorDtl_REC is used to describe the errors (if any) that occurred in the process of NF-e submission to SEFAZ.

When the NF-e is approved by SEFAZ without any errors, fiscal partner makes a call to this CONSUME procedure with status as A, along with the corresponding values in other fields like NFE_ACCESS_KEY, NFE_PROTOCOL, and NFE_DANFE_URL with appropriate data. In this case, the errorDtl_tbl will be NULL since there are no errors associated with it. The ORFM table is updated based on this input data.

If any errors occurred in NF-e processing, the status will be E. Now the errorDtl_tbl will contain the error details. Here NFE_ACCESS_KEY, NFE_PROTOCOL, NFE_DANFE_URL fields will be NULL.When the fresh document (NFe_P) or corrected document (NFe_X) or canceled document (NFe_C) is picked from the staging table FM_ STG_NFE by fiscal partner's Java integrator monitor for polling, fiscal partner makes a call to this CONSUME procedure with status as NFe_I. All the remaining fields will be NULL:

- nfe_access_key

- nfe_protocol

- nfe_danfe_url

- errorDtl_tbl (PL/SQL table type)

If the CONSUME procedure called by fiscal partner is successful, then O_status_code will be S (Success). If it is unsuccessful, it will be E (Error) with the error referenced in the O_error_message.

There could be some network transmission errors during NF-e flow between ORFM, fiscal partner, and SEFAZ. Such error codes are predefined by SEFAZ and do not require the user to correct anything. fiscal partner has provided two codes for network disruptions that can be resent without manual intervention, 286 and 296. Logic in the FM_MS_NFE_SQL.CONSUMEsubscription API automatically re-sends such rejected NFes without manual intervention.

Sistema Público de Escrituração Digital (SPED)

Sistema Público de Escrituração Digital (SPED) or Public System of Digital Bookkeeping is the result of several efforts from the Brazilian government to modernize and increase the level of control over the fiscal transactions for all companies. It is based on a digital file that is transmitted periodically to the government through the internet. Similar to the NF-e, the file is digitally signed through specific programs that validate its format and content.

The strategy adopted to address this requirement was to keep the transaction features in Oracle Retail Fiscal Module (ORFM) and the interface to the Fiscal Authority in Oracle's fiscal partners.

To support this strategy, views and tables make available all information to the fiscal partner based on the fiscal movements.

SPED File Structure

The SPED file that is generated by the fiscal partners contains a structure organized in blocks with opening and closure registers. The information in each block is as follows:

- Block 0: Opening, Identification and References
- Block C: Fiscal Documents I - Merchandise (ICMS/IPI)
- Block D: Fiscal Documents II - Services (ICMS)
- Block E: Fiscal counting of ICMS and IPI
- Block H: Physical Inventory
- Block 1: Other information
- Block 9: Control and Closing of the Digital File

The file is generic and includes information pertinent to all types of companies. The retail segment is required to fill part of the entire file. In addition, the fiscal partner will be in charge of completing the information that is not provided by Oracle Retail.

SPED Solution

The overall solution landscape is based on the existing views that integrate fiscal and master information with Oracle's fiscal partners.

Figure 9–2   SPED Integration
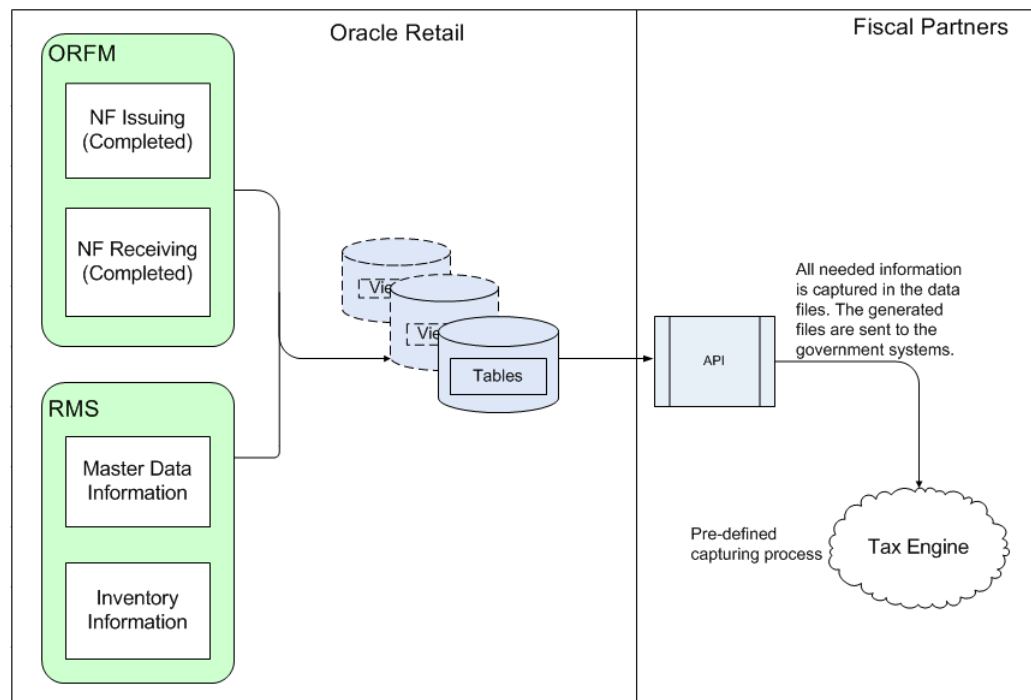
Because the SPED file has several sections corresponding to all types of transactions and fiscal data for a company, the scope of the integration (from the commercial system standpoint) was to make available all data kept within RMS and ORFM.

Because of this, all types of data related to products for resale, and the fiscal transactions related to this type of product, are available in ORFM views and tables.

Products used for consumption, assets, and services, and all transactions related to these types of product, are out of the scope for RMS/ORFM and are not available in the views.

The views and tables created to feed SPED include only data available in RMS/ORFM. The file is generated by the fiscal partner´s solution, and fields (such as file opening and file closing), data related to the version of the SPED program, and all specific data for the file is provided by the fiscal partner. In addition, any field that can be deduced by the fiscal partner should also be provided by them.

SPED interfaces with a third-party system that shares the RMS database and opens the ports to establish network connectivity. It depends on the decision of the client to either host the SPED interfacing application (Interdados) within their environment or host it in a fiscal partner's environment. For security considerations, a separate schema should be created that contains only synonyms to as many objects required by the fiscal partner to generate the SPED information. Only the select privileges should be granted on these synonyms. No insert/update/delete should be allowed.

Performance Optimization for SPED

The anticipated volume of daily operations of a Tier 1 retailer are in the order of a few hundred thousand transactions per day. All these inbound and outbound transactions create NFs that insert records into the ORFM prime tables. The SPED reporting feeds have a monthly frequency that lead to accumulation of large volumes of data. To mitigate the performance concerns for these large volumes, the solution approach has adopted the following strategy:

- Views only use all the foundation/static data tables.

- For the transaction tables that are bound to be voluminous, interface tables are created and populated with the necessary data, required for SPED by a batch job.

- The batch job is run on a daily basis so that the volumes are made manageable by operating on a day window instead of an entire month window.

The primary advantage of this solution is the high impact transaction tables are decoupled from direct access from the external system thereby avoiding the risk of slowing down the system during online operations. The batch job runs during the online shutdown window and the daily job operates on reduced volumes.

10

Maintaining Tax Related Data

There are a number of tables in RMS that contain data provided by (or derived from) an external tax engine (TaxWeb Tax Rules).

This chapter covers the following topics:

- Installing Rules from TaxWeb Tax Rules
- Understanding the Integrated Batch Schedule
- Maintaining External Tax Information (L10N_BR_EXTAX_MAINT)

Installing Rules from TaxWeb Tax Rules

Perform the following steps to install an updated set of rules from TaxWeb Tax Rules tax engine:

1. Stop the WebLogic Server for the domain in which RTIL is deployed.

2. Copy the TaxWeb Tax Rules `.jar` files to `WEBLOGIC_DOMAIN_HOME/lib`.

3. Start the WebLogic Server for the domain in which RTIL is deployed.

4. After a new TaxWeb `.jar` is loaded and the RTIL WebLogic Server runs, set the L10N_BR_EXTAX_REFRESH_CONFIG.REFRESH_NEEDED to **Y**. This triggers updates to the tables in RMS that contain data provided by (or derived from) the external tax engine (like the TaxWeb Tax Rules).

Understanding the Integrated Batch Schedule

The integrated batch schedule process consists of the following:

- Fiscal item reclassification Retail
- Fiscal item reclassification Cost
- Tax Law Changes Retail
- Tax Law Changes Cost

Fiscal item reclassification Retail

This consists of the following:

- `fiscal_reclass_item_extax_setup_retail.ksh` – This module looks at all the valid item/location combinations in RMS for items having their fiscal attributes changed. It determines the unique fiscal attribute groups that cover all

the item/locations combinations and puts the groups on a stage table (`l10n_br_extax_stg_retail`).

- `fiscal_reclass_item_process_retail.ksh` – This module picks the staged groups placed on l10n_br_extax_stg_retail by `fiscal_reclass_item_extax_setup_retail.ksh` and calls the external tax provider with them. The results of these calls are placed on the l10n_br_extax_res_retail and l10n_br_extax_res_retail_det tables.

- `fiscal_reclass_item_extax_finish_retail.ksh` – This module picks the group level tax call results that `fiscal_reclass_item_process_retail.ksh` placed on l10n_br_extax_res_retail and l10n_br_extax_res_retail_det and explodes them back to the item/location level. It then uses the item/location level information to write tax information to GTAX_ITEM_ROLLUP and POS_MODS_TAX_INFO.

Fiscal item reclassification Cost

This consists of the following:

- `fiscal_item_reclass_cost.ksh` – This module create cost changes for all item/supplier/origin country/location combinations that are affected by fiscal item reclassifications. The normal cost change processes update RMS with the new tax law information when necessary. If the default location is not ranged to an item being reclassified, the ITEM_COST_HEAD/DETAIL tables are handled directly.

Item Cost for Non-Brazil Supplier

While computing the item cost, there is no need to make a tax call to tax rules engine if the supplier linked to an item is not a Brazilian Supplier. RMS checks the country of the supplier. RMS does not make a tax call to tax rules engine, and instead populates the item costing tables (Item_Cost_Head and Item_Cost_Detail) without any tax details for non-Brazilian suppliers.

There is no need to navigate to item costing screen in case supplier's country is not Brazil. The Negotiated Item Cost (NIC), Base Cost (BC), Extended Base Cost (EBC), and Inclusive Cost (IC) are calculated with zero tax values. The **Delivery country** is set to **Brazil**, and **primary delivery country** indicator is set to **Y**. The **NIC** static indicator is also set to **Y**. These defaults do not impact the cost calculations as there is no tax in this case.

RMS continues to make tax calls to the tax rules engine for Brazilian suppliers that is, suppliers for which the country on the supplier's primary address is set to Brazil. All values passed to tax rules engine for Cost computation must be in Brazilian Reais (BRL), irrespective of the supplier currency. If the supplier currency is BRL, then the values passed to tax rules engine and retrieved from tax rules engine are in BRL and can be stored in BRL in the RMS tables. In case the supplier operates in a currency other than BRL, then RMS converts the cost values into BRL using the currency conversion rates defined in RMS. These converted values are passed to tax rules engine which return values of cost components and taxes in BRL. These cost and tax values are converted back into the supplier currency before storing them in the database.

Navigate to the item costing screen and click **Calculate Tax**. The data is retrieved with zero taxes.

The data conversion scripts for non-Brazil supplier picks the items in Approved status and inserts the data as it is from dat files to tables.

The item costing will not be made and it is not a correct business scenario, when an item has a non-Brazil delivery country.

Tax Law Changes Retail

This consists of the following:

- `refresh_extax_setup_retail.ksh` – This module gets a list of fiscal attributes that have tax law changes scheduled against them since the last run. It then uses this information to determine what item/locations will be affected by those scheduled tax law changes. It determines the unique fiscal attribute groups that cover those item/locations combinations and puts the groups on a stage table (l10n_br_extax_stg_retail).

- `refresh_extax_process_retail.ksh` – This module picks the staged groups placed on l10n_br_extax_stg_retail by `refresh_extax_setup_retail.ksh` and calls the external tax provider with them. The results of these calls are placed on the l10n_br_extax_res_retail and l10n_br_extax_res_retail_det tables.

- `refresh_extax_finish_retail.ksh` – This module picks the group level tax call results that `refresh_extax_process_retail.ksh` placed on l10n_br_extax_res_retail and l10n_br_extax_res_retail_det and explodes them back to the item/location level. It then uses the item/location level information to write tax information to GTAX_ITEM_ROLLUP and POS_MODS_TAX_INFO.

Tax Law Changes Cost

This consists of the following:

- `refresh_extax_future_cost.ksh` – This module gets a list of fiscal attributes that have tax law changes scheduled against them since its last run. It then uses this information to determine what item/supplier/origin country/locations will be affected by those scheduled tax law changes. These item/supplier/origin country/location combinations are then used to create new cost changes on the date of their tax law changes. The normal cost change processes then take care of updating RMS with the new tax law information when necessary.

Program Flow

The basic flow of data from a triggering event to the RMS tables holding tax information is shown below. There are three diagrams, one for retail and two for cost.

Figure 10–1 Retail Tax Flow
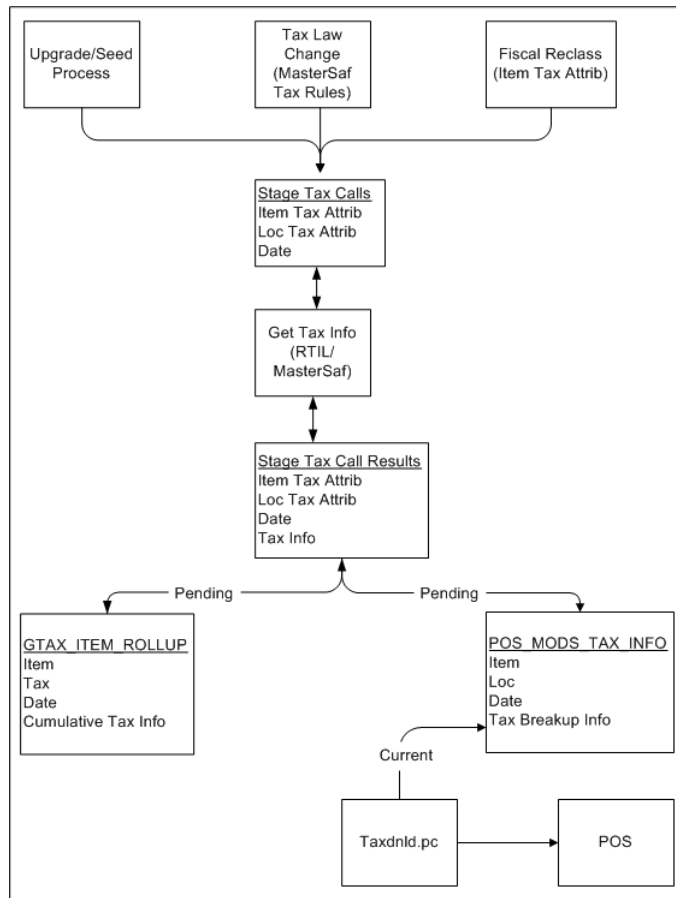
Figure 10–2   Cost Tax Flow
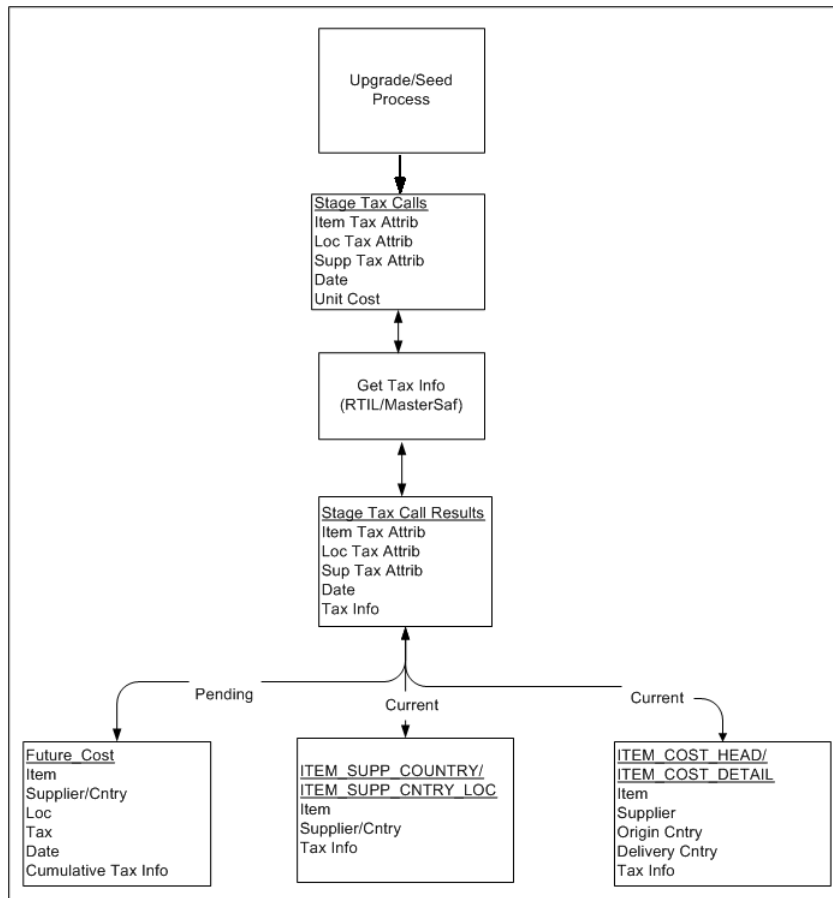
Figure 10–3   Cost Tax Flow Maintenance
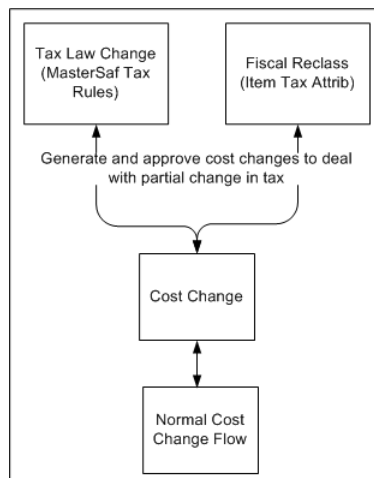
Maintaining External Tax Information (L10N_BR_EXTAX_MAINT)

The tables can be divided into two categories depending on whether they hold data related to retail tax or cost tax.

■   Retail Tables – GTAX_ITEM_ROLLUP and POS_MODS_TAX_INFO

- Cost Tables – ITEM_SUPP_COUNTRY, ITEM_SUPP_COUNTRY_LOC, FUTURE_ COST, ITEM_COST_HEAD, ITEM_COST_DETAIL

These tables can also be categorized by whether they hold the current state tax information or pending state tax information. The current state tables only get updated when the transaction effecting tax becomes active and that pending state tables get updated as soon as the transaction effecting tax becomes known. For example, when we understand from TaxWeb Tax Rules that the sales tax is moving from 10% to 12% in San Paulo in two weeks, the pending state tables are updated as soon this information is known. The current state tables are not updated until two weeks from now when the rate change actually occurs.

- Current State Tables – ITEM_SUPP_COUNTRY, ITEM_SUPP_COUNTRY_LOC, ITEM_COST_HEAD, ITEM_COST_DETAIL
- Pending State Tables – GTAX_ITEM_ROLLUP, POS_MODS_TAX_INFO, FUTURE_COST

Since the tax information is stored in RMS tables, these tables have to contain the factors that determine how tax change is calculated.

The following factors drive retail tax recalculations:

- Item tax attributes (including dimensions)
- Location tax attributes/location address
- Date

The following factors drive cost tax recalculations:

- Item tax attributes (including dimensions)
- Location tax attributes/location address
- Supplier tax attributes/supplier address
- Unit cost
- Date

L10N_BR_EXTAX_MAINT_SQL uses data grouping to minimize the number of calls to an external tax engine.

The idea behind grouping is that it is the fiscal attributes that determine what the tax is, not the particular entity (many different items may all share the same fiscal attributes - same thing for suppliers and locations). The ITEM_LOC table might have 1 million rows on it, but there might only be 125,000 unique combinations of item fiscal attributes and location fiscal attributes between them. If that is the case, only 125,000 calls need to be made to the external tax engine instead of 1 million calls.

The following batches are called by this package:

- Fiscal_Item_Reclass_Cost.ksh
- Fiscal_Reclass_Item_Extax_Finish_Retail.ksh
- Fiscal_Reclass_Item_Extax_Setup_Retail.ksh
- Fiscal_Reclass_Item_Process_Retail.ksh
- Refresh_Extax_Finish_Retail.ksh
- Refresh_Extax_Future_Cost.ksh
- Refresh_Extax_Process_Retail.ksh
- Refresh_Extax_Setup_Retail.ksh

Fiscal_Item_Reclass_Cost.ksh

This module creates cost changes for all item/supplier/origin country/location combinations that are affected by fiscal item reclassifications. The normal cost change processes will then take care of updating RMS with the new tax law information when necessary. If the default location is not ranged to an item being reclassified, the ITEM_COST_HEAD/DETAIL tables are handled directly.

Functional Area

TaxWeb Tax Rules Integration

Module Affected

Fiscal_Item_Reclass_Cost.ksh

Design Overview

This batch stores and maintains data that comes from the external tax engine like TaxWeb Tax Rules.

The work is done in the L10N_BR_EXTAX_MAINT _SQL package.

Scheduling Constraints

Table 10–1   Scheduling Constraints

Schedule Information	Description
Processing Cycle	Adhoc
Scheduling Considerations	Phase 1
Pre-Processing	None
Post-Processing	Fiscal_Reclass_Item_Extax_Setup_Retail.ksh
Threading Scheme	NA

Restart/Recovery

NA

Key Tables Affected

Table 10–2   Tables Affected

Tables	Select	Insert	Update	Delete
GTAX_COST_CHANGE_GTT	Yes	Yes	Yes	No
L10N_BR_FISCAL_RECLASS	Yes	No	No	No
ITEM_SUPP_COUNTRY_LOC	Yes	No	No	No
ITEM_MASTER	Yes	No	No	No
ADDR	Yes	No	No	No
FUTURE COST TABLES	Yes	Yes	Yes	Yes
ITEM_COST_HEAD	Yes	Yes	Yes	No
MV_L10N_ENTITY	Yes	No	No	No
COUNTRY_ATTRIB	Yes	No	No	No

Table 10–2 (Cont.) Tables Affected

Tables	Select	Insert	Update	Delete
ITEM_SUPP_COUNTRY	Yes	No	No	No
ITEM_COST_DETAIL	Yes	Yes	Yes	No
SYSTEM_OPTIONS	Yes	No	No	No
SUPS	Yes	No	No	No
cost_susp_sup_head	No	Yes	No	No
cost_susp_sup_detail_loc	No	Yes	No	No
cost_event_run_type_config	Yes	No	No	No
cost_event	No	Yes	No	No
item_cost_details	No	No	No	Yes
store	Yes	No	No	No
WH	Yes	No	No	No
partner	Yes	No	No	No
l10n_tax_object_config	Yes	No	No	No
Vat_item	Yes	No	No	No

I/O Specification

NA

Fiscal_Reclass_Item_Extax_Finish_Retail.ksh

This module picks up the group level tax call results that `fiscal_reclass_item_ process_retail.ksh` places on l10n_br_extax_res_retail and l10n_br_extax_res_ retail_det and explodes them back to the item/location level. It then uses the item/location level information to write tax information to GTAX_ITEM_ROLLUP and POS_MODS_TAX_INFO tables.

Functional Area

TaxWeb Tax Rules Integration

Module Affected

`Fiscal_Reclass_Item_Extax_Finish_Retail.ksh`

Design Overview

This batch stores and maintains data that comes from the external tax engine like TaxWeb Tax Rules.

The work is done in the L10N_BR_EXTAX_MAINT _SQL package.

Scheduling Constraints

Table 10–3 Scheduling Constraints

Schedule Information	Description
Processing Cycle	Adhoc

Table 10–3   (Cont.) Scheduling Constraints

Schedule Information	Description
Scheduling Considerations	Phase 1
Pre-Processing	None
Post-Processing	Fiscal_Reclass_Item_Extax_Finish_Retail.ksh
Threading Scheme	NA

Restart/Recovery

NA

Key Tables Affected

Table 10–4   Tables Affected

Tables	Select	Insert	Update	Delete
L10N_BR_EXTAX_HELP_RETAIL_GTT	Yes	Yes	No	No
ITEM_LOC	Yes	No	No	No
ITEM_MASTER	Yes	No	No	No
ITEM_SUPP_COUNTRY	Yes	No	No	No
ITEM_SUPP_COUNTRY_DIM	Yes	No	No	No
L10N_BR_EXTAX_RES_RETAIL	Yes	No	No	No
L10N_BR_FISCAL_RECLASS	Yes	No	Yes	No
L10N_BR_EXTAX_DEST_GROUP_HELP	Yes	No	No	No
V_BR_ITEM_FISCAL_ATTRIB	Yes	No	No	No
L10N_BR_EXTAX_HELP_NV_PAIR	Yes	No	No	No
ITEM_LOC	Yes	No	No	No
ITEM_MASTER	Yes	No	No	No
V_PACKSKU_QTY	Yes	No	No	No
GTAX_ITEM_ROLLUP	No	Yes	Yes	No
POS_MODS_TAX_INFO	No	Yes	Yes	No
L10N_BR_EXTAX_RES_RETAIL_DET	Yes	No	No	No
L10N_BR_FISCAL_RECLASS_NV	Yes	No	No	No
L10N_ATTRIB	Yes	No	No	No
L10N_ATTRIB_GROUP	Yes	No	No	No
EXT_ENTITY	Yes	No	No	No
ITEM_COUNTRY_L10N_EXT	No	Yes	Yes	No

I/O Specification

NA

Fiscal_Reclass_Item_Extax_Setup_Retail.ksh

This module looks at all the valid item/location combinations in RMS for items having their fiscal attributes changed. It determines the unique fiscal attribute groups that cover all the item/locations combinations and puts the groups on a stage table (l10n_br_extax_stg_retail).

Functional Area

TaxWeb Tax Rules Integration

Module Affected

Fiscal_Reclass_Item_Extax_Setup_Retail.ksh

Design Overview

This batch stores and maintains data that comes from the external tax engine like TaxWeb Tax Rules.

The work is done in the L10N_BR_EXTAX_MAINT _SQL package.

Scheduling Constraints

Table 10–5   Scheduling Constraints

Schedule Information	Description
Processing Cycle	Adhoc
Scheduling Considerations	Phase 1
Pre-Processing	Fiscal_Item_Reclass_Cost.ksh
Post-Processing	Fiscal_Reclass_Item_Process_Retail.ksh
Threading Scheme	NA

Restart/Recovery

NA

Key Tables Affected

Table 10–6   Tables Affected

Tables	Select	Insert	Update	Delete
L10N_BR_EXTAX_STG_RETAIL	Yes	Yes	Yes	Yes
L10N_BR_EXTAX_RES_RETAIL	No	No	No	Yes
L10N_BR_EXTAX_RES_RETAIL_DET	No	No	No	Yes
V_BR_STORE	Yes	No	No	No
L10N_BR_ENTITY_CNAE_CODES	Yes	No	No	No
V_BR_PARTNER	Yes	No	No	No
V_BR_WH	Yes	No	No	No
WH	Yes	No	No	No
L10N_BR_SUP_TAX_REGIME	Yes	No	No	No
ADDR	Yes	No	No	No

Table 10–6   (Cont.) Tables Affected

Tables	Select	Insert	Update	Delete
COUNTRY_TAX_JURISDICTION	Yes	No	No	No
COUNTRY	Yes	No	No	No
STATE	Yes	No	No	No
L10N_BR_EXTAX_HELP_NV_PAIR	Yes	Yes	Yes	Yes
L10N_BR_EXTAX_DEST_GROUP_HELP	Yes	Yes	Yes	Yes
EXT_ENTITY	Yes	No	No	No
L10N_ATTRIB_GROUP	Yes	No	No	No
L10N_ATTRIB	Yes	No	No	No
L10N_BR_FISCAL_RECLASS	Yes	Yes	No	No
ITEM_LOC	Yes	No	No	No
ITEM_MASTER	Yes	No	No	No
ITEM_SUPP_COUNTRY	Yes	No	No	No
ITEM_SUPP_COUNTRY_DIM	Yes	No	No	No
V_PACKSKU_QTY	Yes	No	No	No
DIM	Yes	No	No	No
ITEM_STG	Yes	No	No	No
L10N_BR_TAX_CALL_STAGE_ROUTING	Yes	No	No	No
V_BR_ITEM_FISCAL_ATTRIB	Yes	No	No	No
V_BR_SUPS	Yes	No	No	No
V_BR_ITEM_COUNTRY_ATTRIBUTES	Yes	No	No	No

I/O Specification

NA

Fiscal_Reclass_Item_Process_Retail.ksh

This module picks the staged groups placed on l10n_br_extax_stg_retail by `fiscal_reclass_item_extax_setup_retail.ksh` and calls the external tax provider with them. The results of these calls are placed in the l10n_br_extax_res_retail and l10n_br_extax_res_retail_det tables.

Functional Area

TaxWeb Tax Rules Integration

Module Affected

`Fiscal_Item_Reclass_Cost.ksh`

Design Overview

This batch stores and maintains data that comes from the external tax engine like TaxWeb Tax Rules.

The work is done in the L10N_BR_EXTAX_MAINT _SQL package.

Scheduling Constraints

Table 10–7   Scheduling Constraints

Schedule Information	Description
Processing Cycle	Adhoc
Scheduling Considerations	Phase 1
Pre-Processing	Fiscal_Reclass_Item_Extax_Setup_Retail.ksh
Post-Processing	Fiscal_Reclass_Item_Extax_Finish_Retail.ksh
Threading Scheme	None

Restart/Recovery

NA

Key Tables Affected

Table 10–8   Tables Affected

Tables	Select	Insert	Update	Delete
RETAIL_SERVICE_REPORT_URL	Yes	No	No	No

I/O Specification

NA

Refresh_Extax_Finish_Retail.ksh

This module writes the group level tax call results to GTAX_ITEM_ROLLUP and POS_MODS_TAX_INFO tables.

Functional Area

TaxWeb Tax Rules Integration

Module Affected

REFRESH_EXTAX_FINISH_RETAIL.KSH

Design Overview

This module picks the group level tax call results that `refresh_extax_process_ retail.ksh` places on l10n_br_extax_res_retail and l10n_br_extax_res_retail_det tables and explodes them back to the item/location level. It then uses the item/location level information to write tax information to GTAX_ITEM_ROLLUP and POS_MODS_TAX_INFO tables.

Scheduling Constraints

Table 10–9   Scheduling Constraints

Schedule Information	Description
Processing Cycle	Adhoc
Scheduling Considerations	Phase 1

Table 10–9   (Cont.) Scheduling Constraints

Schedule Information	Description
Pre-Processing	refresh_extax_process_retail.ksh
Post-Processing	None
Threading Scheme	None

Restart/Recovery

NA

Key Tables Affected

Table 10–10   Tables Affected

Tables	Select	Insert	Update	Delete
L10N_BR_EXTAX_REFRESH_CONFIG	No	No	Yes	No
L10N_BR_EXTAX_HELP_RETAIL_GTT	Yes	Yes	No	No
ITEM_LOC	Yes	No	No	No
ITEM_MASTER	Yes	No	No	No
ITEM_SUPP_COUNTRY	Yes	No	No	No
ITEM_SUPP_COUNTRY_DIM	Yes	No	No	No
L10N_BR_EXTAX_RES_RETAIL	Yes	No	No	No
L10N_BR_EXTAX_DEST_GROUP_HELP	Yes	No	No	No
V_BR_ITEM_FISCAL_ATTRIB	Yes	No	No	No
L10N_BR_EXTAX_HELP_NV_PAIR	Yes	No	No	No
V_PACKSKU_QTY	Yes	No	No	No
GTAX_ITEM_ROLLUP	No	Yes	Yes	No
POS_MODS_TAX_INFO	No	Yes	Yes	No
L10N_BR_EXTAX_RES_RETAIL_DET	Yes	No	No	No
V_BR_ITEM_COUNTRY_ATTRIBUTES	Yes	No	No	No

I/O Specification

NA

Refresh_Extax_Future_Cost.ksh

This module gets a list of fiscal attributes that have tax law changes scheduled against them since its last run.

Functional Area

TaxWeb Tax Rules Integration

Module Affected

REFRESH_EXTAX_FUTURE_COST.KSH

Design Overview

This module gets a list of fiscal attributes that have tax law changes scheduled against them since its last run. It then uses this information to determine what item/supplier/origin country/locations are affected by those scheduled tax law changes. These item/supplier/origin country/location combinations are then used to create new cost changes on the date of their tax law changes. The normal cost change processes then take care of updating RMS with the new tax law information when necessary.

The work is done in the L10N_BR_EXTAX_MAINT _SQL package.

Scheduling Constraints

Table 10–11   Scheduling Constraints

Schedule Information	Description
Processing Cycle	Adhoc
Scheduling Considerations	Phase 1
Pre-Processing	
Post-Processing	
Threading Scheme	None

Restart/Recovery

NA

Key Tables Affected

Table 10–12   Tables Affected

Tables	Select	Insert	Update	Delete
L10N_BR_EXTAX_REFRESH_CONFIG	Yes	No	No	No
L10N_BR_TAX_CALL_RES_FSC_COUNT	Yes	No	No	Yes
L10N_BR_EXTAX_REFRESH_COST	Yes	Yes	No	Yes
V_BR_STORE	Yes	No	No	No
L10N_BR_ENTITY_CNAE_CODES	Yes	No	No	No
V_BR_PARTNER	Yes	No	No	No
V_BR_WH	Yes	No	No	No
WH	Yes	No	No	No
V_BR_SUPS	Yes	No	No	No
L10N_BR_SUP_TAX_REGIME	Yes	No	No	No
ADDR	Yes	No	No	No
COUNTRY_TAX_JURISDICTION	Yes	No	No	No
COUNTRY	Yes	No	No	No
STATE	Yes	No	No	No
L10N_BR_EXTAX_HELP_NV_PAIR	Yes	Yes	Yes	Yes
L10N_BR_EXTAX_DEST_GROUP_HELP	Yes	Yes	Yes	Yes

Table 10–12   (Cont.) Tables Affected

Tables	Select	Insert	Update	Delete
L10N_ATTRIB	Yes	No	No	No
L10N_ATTRIB_GROUP	Yes	No	No	No
EXT_ENTITY	Yes	No	No	No
L10N_BR_TAX_CALL_STAGE_FSC_FDN	Yes	Yes	No	Yes
L10N_BR_TAX_CALL_STAGE_ROUTING	Yes	Yes	No	Yes
RETAIL_SERVICE_REPORT_URL	Yes	No	No	No
L10N_BR_TAX_CALL_STAGE_RMS	Yes	No	No	Yes
L10N_BR_TAX_CALL_RES_TAX_CNTRB	Yes	No	No	Yes
L10N_BR_TAX_CALL_RES_SRVC_PRD	Yes	No	No	Yes
L10N_BR_TAX_CALL_RES_ECO_CLASS	Yes	No	No	Yes
L10N_BR_TAX_CALL_RES_ITEM_RULE	Yes	No	No	Yes
L10N_BR_TAX_CALL_RES_ITEM_TAX	Yes	No	No	Yes
L10N_BR_TAX_CALL_RES_ITEM	Yes	No	No	Yes
L10N_BR_TAX_CALL_RES	Yes	No	No	Yes
L10N_BR_TAX_STAGE_ECO	Yes	No	No	Yes
L10N_BR_TAX_STAGE_FIS_ENTITY	Yes	No	No	Yes
L10N_BR_TAX_STAGE_ITEM	Yes	No	No	Yes
L10N_BR_TAX_STAGE_NAME_VALUE	Yes	No	No	Yes
L10N_BR_TAX_STAGE_ORDER	Yes	No	No	Yes
L10N_BR_TAX_STAGE_ORDER_EXP	Yes	No	No	Yes
L10N_BR_TAX_STAGE_ORDER_INFO	Yes	No	No	Yes
L10N_BR_TAX_STAGE_REGIME	Yes	No	No	Yes
L10N_BR_TAX_CALL_RES_FSC_FND	Yes	No	No	Yes
L10N_BR_EXTAX_REFRESH_GTT	Yes	Yes	No	Yes
GTAX_COST_CHANGE_GTT	Yes	Yes	Yes	No
ITEM_MASTER	Yes	No	No	No
ITEM_SUPP_COUNTRY_LOC	Yes	No	No	No
COST_SUSP_SUP_HEAD	No	Yes	No	No
COST_SUSP_SUP_DETAIL_LOC	No	Yes	No	No
COST_EVENT_RUN_TYPE_CONFIG	Yes	No	No	No
COST_EVENT_COST_CHG	Yes	Yes	No	No
COST_EVENT	Yes	Yes	No	No
COST_EVENT_RESULT	Yes	No	No	No
COST_EVENT_RUN_TYPE_CONFIG	Yes	No	No	No
l10n_br_fiscal_reclass_nv	Yes	No	No	No
l10n_br_fiscal_reclass	Yes	No	No	No

Table 10–12   (Cont.) Tables Affected

Tables	Select	Insert	Update	Delete
vat_codes	Yes	No	No	No
l10n_br_extax_stg_cost	Yes	No	No	No
country_attrib	Yes	No	No	No
fm_system_options	Yes	No	No	No
l10n_br_extax_stg_cost	Yes	No	No	No
l10n_br_extax_res_cost	No	Yes	No	No
v_br_item_fiscal_attrib	Yes	No	No	No
item_supplier	Yes	No	No	No
item_loc	Yes	No	No	No
cost_event_deal	Yes	No	No	No
cost_event_result	No	Yes	No	No

I/O Specification

NA

Refresh_Extax_Process_Retail.ksh

This module picks the staged groups placed in the l10n_br_extax_stg_retail table by `refresh_extax_setup_retail.ksh` and calls the external tax provider with them. The results of these calls are placed on the l10n_br_extax_res_retail and l10n_br_extax_res_retail_det tables.

Functional Area

TaxWeb Tax Rules Integration

Module Affected

REFRESH_EXTAX_PROCESS_RETAIL.KSH

Design Overview

This module picks the staged groups placed in the staging table by `refresh_extax_setup_retail.ksh` and calls the external tax provider with them. The results of these calls are placed on the staging tables.

The work is done in the L10N_BR_EXTAX_MAINT _SQL package.

Scheduling Constraints

Table 10–13   Scheduling Constraints

Schedule Information	Description
Processing Cycle	Adhoc
Scheduling Considerations	Phase 1
Pre-Processing	refresh_extax_setup_retail.ksh
Post-Processing	refresh_extax_finish_retail.ksh

Table 10–13   (Cont.) Scheduling Constraints

Schedule Information	Description
Threading Scheme	None

Restart/Recovery

NA

Key Tables Affected

Table 10–14   Tables Affected

Tables	Select	Insert	Update	Delete
RETAIL_SERVICE_REPORT_URL	Yes	No	No	No
L10N_BR_TAX_CALL_STAGE_ROUTING	Yes	No	No	No
VAT_CODES	Yes	No	No	No
L10N_BR_EXTAX_STG_RETAIL	Yes	No	No	No
L10N_BR_EXTAX_HELP_NV_PAIR	Yes	No	No	No
L10N_BR_EXTAX_RES_RETAIL	No	Yes	No	No
L10N_BR_EXTAX_RES_RETAIL_DET	No	Yes	No	No

I/O Specification

NA

Refresh_Extax_Setup_Retail.ksh

This module gets a list of fiscal attributes that have tax law changes scheduled against them since the last run.

Functional Area

TaxWeb Tax Rules Integration

Module Affected

REFRESH_EXTAX_SETUP_RETAIL.KSH

Design Overview

This module gets a list of fiscal attributes that have tax law changes scheduled against them since the last run. It then uses this information to determine what item/locations are affected by those scheduled tax law changes. It determines the unique fiscal attribute groups that cover those item/locations combinations and puts the groups on a stage table (l10n_br_extax_stg_retail).

The work is done in the L10N_BR_EXTAX_MAINT _SQL package.

Scheduling Constraints

Table 10–15   Scheduling Constraints

Schedule Information	Description
Processing Cycle	Adhoc

Table 10–15   (Cont.) Scheduling Constraints

Schedule Information	Description
Scheduling Considerations	Phase 1
Pre-Processing	
Post-Processing	refresh_extax_process_retail.ksh
Threading Scheme	NA

Restart/Recovery

NA

Key Tables Affected

Table 10–16   Tables Affected

Tables	Select	Insert	Update	Delete
L10N_BR_EXTAX_REFRESH_CONFIG	Yes	No	No	No
L10N_BR_TAX_CALL_RES_FSC_COUNT	Yes	No	No	No
L10N_BR_EXTAX_REFRESH_RETAIL	Yes	Yes	No	Yes
L10N_BR_EXTAX_STG_RETAIL	No	Yes	Yes	Yes
L10N_BR_EXTAX_RES_RETAIL	No	No	No	Yes
L10N_BR_EXTAX_RES_RETAIL_DET	No	No	No	Yes
L10N_BR_TAX_CALL_STAGE_FSC_FDN	Yes	Yes	No	Yes
L10N_BR_TAX_CALL_STAGE_ROUTING	Yes	Yes	No	No
RETAIL_SERVICE_REPORT_URL	Yes	No	No	No
L10N_BR_TAX_CALL_STAGE_RMS	Yes	No	No	Yes
L10N_BR_TAX_CALL_RES_TAX_CNTRB	Yes	No	No	Yes
L10N_BR_TAX_CALL_RES_SRVC_PRD	Yes	No	No	Yes
L10N_BR_TAX_CALL_RES_ECO_CLASS	Yes	No	No	Yes
L10N_BR_TAX_CALL_RES_ITEM_RULE	Yes	No	No	Yes
L10N_BR_TAX_CALL_RES_ITEM_TAX	Yes	No	No	Yes
L10N_BR_TAX_CALL_RES_ITEM	Yes	No	No	Yes
L10N_BR_TAX_CALL_RES	Yes	No	No	Yes
L10N_BR_TAX_STAGE_ECO	Yes	No	No	Yes
L10N_BR_TAX_STAGE_FIS_ENTITY	Yes	No	No	Yes
L10N_BR_TAX_STAGE_ITEM	Yes	No	No	Yes
L10N_BR_TAX_STAGE_NAME_VALUE	Yes	No	No	Yes
L10N_BR_TAX_STAGE_ORDER	Yes	No	No	Yes
L10N_BR_TAX_STAGE_ORDER_EXP	Yes	No	No	Yes
L10N_BR_TAX_STAGE_ORDER_INFO	Yes	No	No	Yes
L10N_BR_TAX_STAGE_REGIME	Yes	No	No	Yes
L10N_BR_TAX_CALL_RES_FSC_FND	Yes	No	No	Yes

Table 10–16   (Cont.) Tables Affected

Tables	Select	Insert	Update	Delete
L10N_BR_EXTAX_REFRESH_GTT	Yes	Yes	Yes	Yes
V_BR_ITEM_FISCAL_ATTRIB	Yes	No	No	No
V_BR_STORE	Yes	No	No	No
L10N_BR_ENTITY_CNAE_CODES	Yes	No	No	No
V_BR_PARTNER	Yes	No	No	No
V_BR_WH	Yes	No	No	No
WH	Yes	No	No	No
V_BR_SUPS	Yes	No	No	No
L10N_BR_SUP_TAX_REGIME	Yes	No	No	No
ADDR	Yes	No	No	No
COUNTRY_TAX_JURISDICTION	Yes	No	No	No
COUNTRY	Yes	No	No	No
STATE	Yes	No	No	No
L10N_BR_EXTAX_HELP_NV_PAIR	Yes	No	No	No
L10N_BR_EXTAX_DEST_GROUP_HELP	No	Yes	Yes	No
ITEM_MASTER	Yes	No	No	No
ITEM_LOC	Yes	No	No	No
ITEM_SUPP_COUNTRY	Yes	No	No	No
ITEM_SUPP_COUNTRY_DIM	Yes	No	No	No
V_PACKSKU_QTY	Yes	No	No	No
vat_codes	Yes	No	No	No
l10n_br_extax_stg_retail	Yes	No	No	No
l10n_br_extax_res_retail	No	Yes	No	No
l10n_br_extax_res_retail_det	No	Yes	No	No
l10n_br_tax_call_res_fsc_count	No	No	No	Yes
l10n_br_tax_call_stage_routing	No	No	No	Yes
l10n_attrib	Yes	No	No	No
l10n_attrib_group	Yes	No	No	No
ext_entity	Yes	No	No	No
l10n_br_extax_help_nv_pair	No	Yes	Yes	Yes
l10n_br_extax_stg_retail	Yes	No	No	No
l10n_br_extax_dest_group_help	Yes	No	No	Yes

I/O Specification

NA

11

Understanding Batch Processes

The batch overview provided in this chapter identifies the functional area description with the batch processes illustrated in the designs. This overview allows you to determine how a business function works behind the scenes.

Batch designs describe how, on a technical level, an individual batch module works and the database tables that it affects. In addition, batch designs contain file layout information that is associated with the batch process.

This chapter covers the following topics:

- TaxWeb Tax Rules Tax Engine Integration Batch
- Financial Postings Batch
- Deal Income Postings Batch
- Financial Postings for Tax Changes Batches
- Purging Process Batch
- Localization Batch
- SPED Batch
- NF EDI Batch

TaxWeb Tax Rules Tax Engine Integration Batch

The RMS application provides integration with an external tax solution specific to the Brazilian market. The integration involves RMS, ORFM, RTIL, and a tax engine provided by TaxWeb Tax Rules. Two batch programs have been introduced to facilitate the data transfer between applications.

Batch Design Summary

The following batch designs are included in this functional area:

- L10nbrfisdnld (Fiscal Download)
- L10nbrfreclsprg (Fiscal Reclassification Purge)

L10nbrfisdnld (Fiscal Download)

Use this batch program to facilitate the download of fiscal attributes from the external tax engine.

Functional Area

TaxWeb Tax Rules Integration

Module Affected

L10NBRFISDNLD.PC

Design Overview

This batch downloads the fiscal attributes from the external tax engine. The attribute can be passed as an input parameter to the program. The attribute is the key to be downloaded from the tax engine and it can be passed as input parameter to the program. No parameter is required, if the user wants to download all the fiscal attributes. The cutoff date to download the data from TaxWeb Tax Rules is fetched from FISCAL_ATTR_UPDATE table.

The majority of the work is done in the L10N_BR_FND_SQL package. Depending on the attribute requested to download the corresponding function in the package is invoked. An exception is codes/attributes for the nature of operation (NOP). The NOP download function is contained in the FM_UTILIZATION_SQL package. The list of package function calls mapped to the attributes are as follows:

- NCM – L10N_BR_FND_SQL.MERG_INTO_NCM_CODES
- MASSERV – L10N_BR_FND_SQL.MERG_INTO_MASSERV
- FEDSERV – L10N_BR_FND_SQL.MERG_INTO_FEDSERV
- NCMCARAC – L10N_BR_FND_SQL.MERG_INTO_NCM_CHAR_CODES
- NCMPAUTA – L10N_BR_FND_SQL.MERG_INTO_NCM_PAUTA_CODES
- NCMIPI – L10N_BR_FND_SQL.MERG_INTO_NCM_IPI_CODES
- CNAE – L10N_BR_FND_SQL.MERG_INTO_CNAE_CODES
- NOP – FM_UTILIZATION_SQL.P_CREATE_FM_NOP

The L10N_BR_FISCAL_FDN_QUERY_SQL and FM_UTILIZATION_SQL both call the fiscal foundation query package (L10N_BR_FISCAL_FDN_QUERY_SQL) which invoke RTIL to retrieve the requested fiscal attributes from TaxWeb Tax Rules.

If any fiscal attributes are returned, they will be loaded (merged) to the corresponding table for that fiscal attribute. The tables loaded per attribute are as follows:

- NCM_CODES with NCM codes
- NCM_CHAR_CODES with NCM Characteristics codes
- NCM_IPI_CODES with EX-IPI codes
- NCM_PAUTA_CODES with the NCM Pauta codes
- FEDERAL_SERVICE_CODES with the Federal service codes
- MASTERSAF_SERVICE_CODES with TaxWeb Tax Rules specific codes
- CNAE_CODES with the CNAE codes

Finally, the LAST_UPD_DATE in the FISCAL_ATTRIB_UPDATES tables are updated to the current VDATE for the corresponding attributes once the data is loaded.

Scheduling Constraints

Table 11–1   Scheduling Constraints

Schedule Information	Description
Processing Cycle	Ad-hoc
Scheduling Considerations	This program can run in ad-hoc basis whenever the new fiscal attributes needs to be downloaded from TaxWeb Tax Rules.
Pre-Processing	NA
Post-Processing	NA
Threading Scheme	NA

Restart/Recovery

NA

Locking Strategy

NA

Security Considerations

NA

Performance Considerations

NA

Key Tables Affected

Table 11–2   Tables Affected

Tables	Select	Insert	Update	Delete
NCM_CODES	Yes	Yes	Yes	No
NCM_CHAR_CODES	Yes	Yes	Yes	No
NCM_PAUTA_CODES	Yes	Yes	Yes	No
FEDERAL_SERVICE_CODES	Yes	Yes	Yes	No
MASTERSAF_SERVICE_CODES	Yes	Yes	Yes	No
CNAE_CODES	Yes	Yes	Yes	No
FISCAL_ATTRIB_UPDATES	Yes	Yes	No	No
L10N_BR_TAX_CALL_STAGE_FSC_FDN	Yes	Yes	No	Yes
L10N_BR_TAX_CALL_STAGE_ROUTING	Yes	Yes	No	Yes
RETAIL_SERVICE_REPORT_URL	Yes	No	No	No
L10N_BR_TAX_CALL_RES_FSC_COUNT	No	Yes	No	Yes
L10N_BR_TAX_CALL_RES_FSC_FND	Yes	Yes	No	Yes
FM_NOP	No	Yes	Yes	No

I/O Specification

NA

L10nbrfreclsprg (Fiscal Reclassification Purge)

Use this batch program to clear the database tables.

Functional Area

TaxWeb Tax Rules Integration

Module Affected

L10NBRFRECLSPRG.PC

Design Overview

This batch purges the processed reclassification data from `L10N_BR_FISCAL_RECLASS` table. The records to be purged are based on its processed_date or active_date less than the current vdate along with the status.

Scheduling Constraints

Table 11–3   Scheduling Constraints

Schedule Information	Description
Processing Cycle	Ad-hoc
Scheduling Considerations	This program can run after the successful completion of the following scripts: ■　Fiscal_reclass_item_extax_setup_retail.ksh ■　Extax_process_retail.ksh ■　Fiscal_reclass_item_extax_finish_retail.ksh
Pre-Processing	■　Fiscal_reclass_item_extax_setup_retail.ksh ■　Extax_process_retail.ksh ■　Fiscal_reclass_item_extax_finish_retail.ksh
Post-Processing	NA
Threading Scheme	Threading based on reclassification id.

Restart/Recovery

NA

Locking Strategy

NA

Security Considerations

NA

Performance Considerations

NA

Key Tables Affected

Table 11–4   Tables Affected

Tables	Select	Insert	Update	Delete
L10N_BR_FISCAL_RECLASS	Yes	No	No	Yes
PERIOD	Yes	No	No	No

I/O Specification

NA

Financial Postings Batch

Use this batch program to roll up financial data for accounting purposes.

Batch Design Summary

The following batch designs are included in this functional area:

- fmfinpost.pc
- fmtrandata.pc

fmfinpost.pc

Functional Area

Financial Postings

Module Affected

fm_financial_posting_sql

Design Overview

Rolls up the transaction amount into accounts based on general ledger.

Scheduling Constraints

Table 11–5   Scheduling Constraints

Schedule Information	Description
Processing Cycle	Ad-hoc
Scheduling Considerations	Ad-hoc
Pre-Processing	NA
Post-Processing	NA
Threading Scheme	NA

Restart/Recovery

NA

Key Tables Affected

Table 11–6   Tables Affected

Tables	Select	Insert	Update	Delete
fm_gl_options	Yes	No	No	Yes
Fm_coa_setup	Yes	No	No	No
Fm_sob_setup	Yes	No	No	No
Fm_account_setup	Yes	No	No	No
Fm_dynamic_segment_setup	Yes	Yes	No	No
Fm_gl_cross_ref	Yes	Yes	No	No
Fm_gl_dynamic_attributes	Yes	Yes	No	No
Fm_tran_data	Yes	No	No	No
fm_fiscal_doc_header	Yes	No	Yes	No
fm_schedule	Yes	No	Yes	No

I/O Specification
NA

fmtrandata.pc

Functional Area
Financial Postings

Module Affected
fm_trandata_posting_sql

Design Overview
Computes transactional data based on the transaction codes.

Scheduling Constraints

Table 11–7   Scheduling Constraints

Schedule Information	Description
Processing Cycle	Ad-hoc
Scheduling Considerations	Ad-hoc
Pre-Processing	NA
Post-Processing	NA
Threading Scheme	Multi-threaded based on location.

Restart/Recovery
The logical unit of work is location_id and fiscal_doc_id.

Key Tables Affected

Table 11–8   Tables Affected

Tables	Select	Insert	Update	Delete
fm_tran_data	No	Yes	No	No
fm_fiscal_doc_header	Yes	No	Yes	No

I/O Specification

NA

Deal Income Postings Batch

Use this batch program to create NFs in Approved status.

Batch Design Summary

The following batch design is included in this functional area:

- fmdealinc.pc

fmdealinc.pc

Functional Area

Deal Income Postings

Module Affected

NA

Design Overview

This batch program reads the data from RMS staging tables `stage_fixed_deal_head`, `stage_fixed_deal_detail`, `stage_complex_deal_head`, and `stage_complex_deal_detail` and creates the NFs in Approved status. If the NF creation fails for some fatal reason, the error is logged in the error file. This batch runs on a daily basis after vendinvc and vendinvf RMS batches.

Scheduling Constraints

Table 11–9   Scheduling Constraints

Schedule Information	Description
Processing Cycle	Ad-hoc
Scheduling Considerations	Ad-hoc
Pre-Processing	vendinvc and vendinvf RMS batches
Post-Processing	NA
Threading Scheme	Multi-threaded based on deal_id

Restart/Recovery

The logical unit of work is Deal_id, Deal_detail_id, and location_id.

Key Tables Affected

Table 11–10   Tables Affected

Tables	Select	Insert	Update	Delete
FM_FIXED_DEAL_LOC _ DEFAULT	Yes	No	No	No
STAGE_FIXED_DEAL_ HEAD	Yes	No	No	No
STAGE_FIXED_DEAL_ DETAIL	Yes	No	No	No
STAGE_COMPLEX_DEAL_ HEAD	Yes	No	No	No
STAGE_COMPLEX_DEAL_ DETAIL	Yes	No	No	No
TSF_ENTITY_ORG_UNIT_ SOB	Yes	No	No	No
MV_L10N_ENTITY	Yes	No	No	No
SYSTEM_OPTIONS	Yes	No	No	No
FM_SYSTEM_OPTIONS	Yes	No	No	No
FM_FISCAL_DOC_HEADER	Yes	Yes	No	No
FM_FISCAL_DOC_DETAIL	No	Yes	No	No

I/O Specification
NA

Financial Postings for Tax Changes Batches

Use these batch programs to update the Weighted Average Cost (WAC) and the
TaxWeb historical data.

Batch Design Summary

The following batch designs are included in this functional area:

- fmprepost.pc
- fmtaxupld.pc
- fmtaxchg.pc
- fmpurge.pc

fmprepost.pc

This batch has pre/post dependencies on the fmtaxupld batch.

Functional Area

Pre/post functionality for ORFM

Module Affected

NA

Design Overview

This batch program has pre/post module for fmtaxupld batch. It contains fmtaxupld_ post function. This function processes the records from staging table `FM_STG_ TAXCHG_DTL`, contains process_ind as 'N', and inserts the records into `FM_TAXCHG_ HEADER` and `FM_TAXCHG_DETAIL` tables.

This program considers the following four parameters:

- Username/password to log on to Oracle

- A program before or after which this script must run

- An indicator indicating whether the script is a pre or post function

- chunk_size for fmtaxupld only, which defines the number of records under one location that need to clubbed as single chunk

Scheduling Constraints

Table 11–11   Scheduling Constraints

Schedule Information	Description
Processing Cycle	Ad-hoc
Scheduling Considerations	Ad-hoc
Pre-Processing	Fmtaxupld
Post-Processing	NA
Threading Scheme	NA

Restart/Recovery

NA

Key Tables Affected

Table 11–12   Tables Affected

Tables	Select	Insert	Update	Delete
FM_STG_TAXCHG_DTL	Yes	No	Yes	Yes
FM_TAXCHG_HEADER	Yes	Yes	No	No
FM_TAXCHG_DETAIL	Yes	Yes	No	No
V_FISCAL_ATTRIBUTES	Yes	No	No	No
FM_TAXCHG_HEADER_ HIST	Yes	No	No	No
FM_TAXCHG_DETAIL_ HIST	Yes	No	No	No

I/O Specification
NA

fmtaxupld.pc

Use this batch program to upload the 'Tax Change Upload' input file consisting of tax rule changes into the system for processing purpose.

Functional Area
WAC and Substituição Tributária (ST) History Update

Module Affected
NA

Design Overview
FMTAXUPLD.PC is a Pro*C program that runs as a module in Adhoc. The purpose is to upload and process Tax Change input file consisting of tax rule changes. The module accepts Tax Change data contained in a flat file (ASCII text) and formatted to match the prescribed retail input file format.

All items are validated, FMTAXUPLD.PC writes a row into the FM_STG_TAXCHG_DTL table.

Scheduling Constraints

Table 11–13   Scheduling Constraints

Schedule Information	Description
Processing Cycle	Ad-hoc
Scheduling Considerations	Ad-hoc
Pre-Processing	NA
Post-Processing	Fmtaxupld post
Threading Scheme	NA

Restart/Recovery
Since this is a file based upload, file based restart/recovery logic is applied. The commit_max_ctr field should be set to prevent excessive rollback space usage, and to reduce the overhead of file I/O.

Key Tables Affected

Table 11–14   Tables Affected

Tables	Select	Insert	Update	Delete
FM_STG_TAXCHG_DTL	No	Yes	No	Yes
ITEM_MASTER	Yes	No	No	No
ITEM_LOC_SOH	Yes	No	No	No
SYSTEM_OPTIONS	Yes	No	No	No
WH	Yes	No	No	No

I/O Specification

The input file name is not fixed and it is determined by a run time parameter. Records rejected by the import process are written to a reject file. The reject file name is not fixed and it is determined by a run time parameter.

Table 11–15   Input File

Record Name	Field Name	Field Type	Default Value	Description
FHEAD	File record descriptor	Char (5)	FHEAD	It describes the file line type.
	Line number	Number (10)	0000000001	it is a sequential file line number.
	File Type	Char (4)	TXUP	It is the Tax change Upload.
	File create date	Char (14)		It is YYYYMMDDHH24MISS format.
FDETL	File record descriptor	Cha r(5)	FDETL	It describes the file line type.
	Line number	Number (10)		It is the sequential file line number.
	Action Type	Char (1)		Following is the type of mode: 'I'nsert or 'D'elete
	NF number	Number (15)		The NF number is used only for delete scenario.
	Series No	Char (20)		The series number is used only for delete scenario.
	Item	Char (25)		It is the item number.
	Location_id	Number (10)		It is the location ID.
	Location_type	Char (1)		Following is the type of location: 'S'tore or 'W'arehouse
	Quantity	Number (12)		Quantity * 10000 (4 implied decimal places), quantity of item at that location. It is used to update the MTR history tables.
	Unit cost	Number (20)		Unit cost * 10000 (4 implied decimal places), Unit cost of the item. It is assumed that it will be in location currency.
	ICMS unit amount	Number (20)		Imposto Sobre Circulação de Mercadorias e Serviços (ICMS) unit amount * 10000 (4 implied decimal places), ICMS amount changed per unit. It is assumed that it will be in location currency.

Table 11–15   (Cont.) Input File

Record Name	Field Name	Field Type	Default Value	Description
	ICMSST unit amount	Number (20)		ICMSST unit amount * 10000 (4 implied decimal places), ICMSST amount changed per unit. It is assumed that it will be in location currency.
	ICMSST Base Value	Number (20)		ICMSST Base Value* 10000 (4 implied decimal places), ICMSST Base value per unit for the item. It is assumed that it will be in location currency.
	ICMSSTE unit amount	Number (20)		ICMSSTE unit amount * 10000 (4 implied decimal places), ICMSSTE amount changed per unit. It is assumed that it will be in location currency
	ICMSSTE Base Value	Number (20)		ICMSSTE Base Value * 10000 (4 implied decimal places), ICMSSTE Base value per unit for the item. It is assumed that it will be in location currency.
	New average cost	Number (20)		New average cost * 10000 (4 implied decimal places), New average cost of the item after tax law change. It is assumed that it will be in location currency.
FTAIL	File record descriptor	Char (5)	FTAIL	It describes the file record type.
	Line number	Number (10)		It is a sequential file line number (total number of lines in the file).
	Number of detail records	Number (10)		It is the number of FDETL lines in the file.

Fmtaxchg.pc

Use this batch program to process the item/location records affected by tax change.

Functional Area

TAXCHG Processing

Modules Affected

- l10n_br_wacadj_sql

- lfm_ext_taxes_sql

Design Overview

The fmtaxchg RFM batch processes the records from `FM_TAXCHG_HEADER` and `FM_TAXCHG_DETAIL` tables with 'U' un-processed or 'E' error status.

The major functionality of this batch program is divided into the following three steps:

1. Making a call to RMS packaged function L10N_BR_WACADJ_SQL.WAC_UPDATE to update WAC and TRAN_DATA postings for cost variance. In case of unhandled exceptions and package errors, the batch stops further processing and an error is reported in error file.

2. Making a call to FM_EXT_TAXES_SQL.BULK_ST_HIST_UPDATE with taxchg_ids for locations having control_rec_st_ind set to 'Y' in V_FISCAL_ATTRIBUTES, to update the ST history in MTR tables. In case of unhandled exceptions and package errors, the batch stops further processing and an error is reported in error file.

3. The STATUS column is updated to 'A' in `FM_TAXCHG_HEADER` table of WAC and ST history tables if the records are updated successfully.

Scheduling Constraints

Table 11–16   Scheduling Constraints

Schedule Information	Description
Processing Cycle	Ad-hoc
Scheduling Considerations	Ad-hoc
Pre-Processing	NA
Post-Processing	NA
Threading Scheme	Multi Threaded based on taxchg_id

Restart/Recovery

The logical unit of work is taxchg_id.

Key Tables Affected

Table 11–17   Tables Affected

Tables	Select	Insert	Update	Delete
FM_TAXCHG_HEADER	Yes	No	Yes	No
FM_TAXCHG_DETAIL	Yes	No	No	No
V_FM_RESTART_TAXCHG	Yes	No	No	No
V_FISCAL_ATTRIBUTES	Yes	No	No	No

I/O Specification

NA

fmpurge.pc

The fmpurge batch picks/purges the processed records and moves/stores the data in the respective history tables.

Functional Area

Purging Processing

Module Affected

NA

Design Overview

In production environment, as the number of transactions increases over a period of time; in order to keep the performance intact it is required to keep purging the data from the active tables of the application periodically. This batch purges the data from the active tables and stores them in history tables.

You can also purge the data from history tables over a period of time.

Scheduling Constraints

Table 11–18   Scheduling Constraints

Schedule Information	Description
Processing Cycle	Ad-hoc
Scheduling Considerations	Ad-hoc
Pre-Processing	NA
Post-Processing	NA
Threading Scheme	NA

Restart/Recovery

NA

Key Tables Affected

Table 11–19   Tables Affected

Tables	Select	Insert	Update	Delete
FM_TAXCHG_HEADER	No	No	No	Yes
FM_TAXCHG_DETAIL	No	No	No	Yes
FM_TRAN_DATA	No	No	No	Yes
FM_AP_STAGE_HEAD	No	No	No	Yes
FM_AP_STAGE_DETAIL	No	No	No	Yes
FM_TAXCHG_HEADER_HIST	No	Yes	No	Yes
FM_TAXCHG_DETAIL_HIST	No	Yes	No	Yes
FM_TRAN_DATA_HIST	No	Yes	No	Yes
FM_AP_STAGE_HEAD_HIST	No	Yes	No	Yes
FM_AP_STAGE_DETAIL_HIST	No	Yes	No	Yes
FM_SYSTEM_OPTIONS	Yes	No	No	No

I/O Specification

NA

Purging Process Batch

This batch is used to purge the NF records from the transaction and history tables.

fmpurge.pc

Functional Area

Purging Process

Design Overview

In production environment, as the number of transactions increases over a period of time; in order to keep the performance intact it is required to keep purging the data from the active tables of the application periodically. This batch purges the data from the active tables and stores them in history tables.

You can also purge the data from history tables over a period of time.

Scheduling Constraints

Table 11–20   Scheduling Constraints

Schedule Information	Description
Processing Cycle	Ad-hoc
Scheduling Considerations	Ad-hoc
Pre-Processing	NA
Post-Processing	NA
Threading Scheme	NA

Restart/Recovery

NA

Key Tables Affected

Table 11–21   Tables Affected

Tables	Select	Insert	Update	Delete
fm_schedule_hist	No	Yes	No	No
fm_fiscal_doc_header_hist	No	Yes	No	No
fm_fiscal_doc_detail_hist	No	Yes	No	No
fm_tran_data_hist	No	Yes	No	No
fm_receiving_header_hist	No	Yes	No	No
fm_receiving_header_hist	No	Yes	No	No
fm_receiving_detail_hist	No	Yes	No	No
fm_fiscal_doc_tax_head_hist	No	Yes	No	No

Table 11–21   (Cont.) Tables Affected

Tables	Select	Insert	Update	Delete
fm_fiscal_doc_tax_detail_hist	No	Yes	No	No
fm_resolution_hist	No	Yes	No	No
fm_correction_tax_doc_hist	No	Yes	No	No
fm_correction_doc_hist	No	Yes	No	No
fm_nf_doc_tax_head_ext_hist	No	Yes	No	No
fm_nf_doc_tax_detail_ext_hist	No	Yes	No	No
fm_nf_doc_tax_detail_wac_hist	No	Yes	No	No
fm_fiscal_doc_payments_hist	No	Yes	No	No
fm_nf_doc_tax_rule_ext_hist	No	Yes	No	No
fm_sped_fiscal_doc_header_hist	No	Yes	No	No
fm_sped_fiscal_doc_detail_hist	No	Yes	No	No
fm_schedule	Yes	No	No	Yes
fm_fiscal_doc_header	Yes	No	No	Yes
fm_fiscal_doc_detail	Yes	No	No	Yes
fm_tran_data	Yes	No	No	Yes
fm_receiving_header	Yes	No	No	Yes
fm_receiving_header	Yes	No	No	Yes
fm_receiving_detail	Yes	No	No	Yes
fm_fiscal_doc_tax_head	Yes	No	No	Yes
fm_fiscal_doc_tax_detail	Yes	No	No	Yes
fm_resolution	Yes	No	No	Yes
fm_correction_tax_doc	Yes	No	No	Yes
fm_correction_doc	Yes	No	No	Yes
fm_fiscal_doc_tax_head_ext	Yes	No	No	Yes
fm_fiscal_doc_tax_detail_ext	Yes	No	No	Yes
fm_fiscal_doc_tax_detail_wac	Yes	No	No	Yes
fm_fiscal_doc_payments	Yes	No	No	Yes
fm_fiscal_doc_tax_rule_ext	Yes	No	No	Yes
fm_sped_fiscal_doc_header	Yes	No	No	Yes
fm_sped_fiscal_doc_detail	Yes	No	No	Yes
FM_FISCAL_ENTITY_OTHER	Yes	No	No	Yes
FM_FISCAL_ENTITY_OTHER_HIST	Yes	Yes	No	Yes
FM_EDI_FISCAL_ENTITY_OTHER	Yes	No	No	Yes

I/O Specification

NA

Localization Batch

This batch refreshes a materialized view MV_L10N_ENTITY. This materialized view is built on the addresses for entities like stores, warehouses, partners, suppliers, outside locations, and set of books. This materialized view is used in the decoupling logic.

refmvl10entity (Refresh MV MV_L10N_ENTITY)

Functional Area

NA

Module Affected

REFMVL10ENTITY.PC

Design Overview

This is an ad hoc batch program that refreshes the materialized view MV_L10N_ ENTITY that is based on ADDR, OUTLOC, COMPHEAD, and COUNTRY_ATTRIB table.

Scheduling Constraints

Table 11–22

Schedule Information	Description
Processing Cycle	Ad-hoc
Scheduling Considerations	NA
Pre-Processing	NA
Post-Processing	NA
Threading Scheme	NA

Restart/Recovery

This batch program uses table-based restart/recovery. The commit happens in the database when the commit_max_ctr is reached.

Locking Strategy

NA

Security Considerations

NA

Performance Considerations

NA

Key Tables Affected

Table 11–23   Tables Affected

Tables	Select	Insert	Update	Delete
ADDR	Yes	No	No	No

Table 11–23   (Cont.) Tables Affected

Tables	Select	Insert	Update	Delete
OUTLOC	Yes	No	No	No
COMPHEAD	Yes	No	No	No
COUNTRY_ATTRIB	Yes	No	No	No

I/O Specification

NA

SPED Batch

This batch is used to post NF transactions into staging tables for SPED reporting by Mastersaf.

Import_sped

Functional Area

SPED

Module Affected

Import_SPED.ksh

Design Overview

This batch inserts all processed NFs (status = FP) into two tables for SPED - `FM_SPED_FISCAL_DOC_HEADER` and `FM_SPED_FISCAL_DOC_DETAIL`. It looks into the driving table, FM_SPED_LAST_RUN_DATE, for the last run date of SPED to fetch all closed NFs whose transaction amounts are also rolled up into ledger accounts based on gl_cross_ref in between last run date and sysdate from the main tables of ORFM. Once the records are successfully inserted, the batch updates the last _run_date column of the driving table, FM_SPED_LAST_RUN_DATE to sysdate.

It is recommended that you run this batch job on a daily basis due to performance impacts. This batch has a pre-dependency on the ORFM Financial postings batch - , so once Financial postings batch program completes and sets the Nota Fiscal status to 'F'inancially 'P'osted (FP) only after that SPED insert batch job - import_SPED.ksh should be triggered to fetch all such 'F'inancially 'P'osted (FP) NFs from the ORFM tables.

Scheduling Constraints

Table 11–24

Schedule Information	Description
Processing Cycle	Ad-hoc
Scheduling Considerations	This program should run only after the successful completion of FMFINPOST.PC
Pre-Processing	NA
Post-Processing	NA
Threading Scheme	NA

Restart/Recovery

NA

Locking Strategy

NA

Security Considerations

NA

Performance Considerations

NA

Key Tables Affected

Table 11–25   Tables Affected

Tables	Select	Insert	Update	Delete
FM_FISCAL_DOC_HEADER	Yes	No	No	No
FM_FISCAL_DOC_DETAIL	Yes	No	No	No
FM_SCHEDULE	Yes	No	No	No
FM_FISCAL_DOC_TAX_HEAD	Yes	No	No	No
FM_FISCAL_DOC_TAX_DETAIL	Yes	No	No	No
FM_FISCAL_DOC_TAX_DETAIL_EXT	Yes	No	No	No
FM_FISCAL_UTILIZATION	Yes	No	No	No
FM_FISCAL_DOC_PAYMENTS	Yes	No	No	No
FM_SPED_FISCAL_DOC_HEADER	No	Yes	No	No
FM_SPED_FISCAL_DOC_DETAIL	No	Yes	No	No
ITEM_MASTER	Yes	No	No	No
FM_SPED_LAST_RUN_DATE	Yes	No	Yes	No
V_BR_ITEM_FISCAL_ATTRIB	Yes	No	No	No
V_FISCAL_ATTRIBUTES	Yes	No	No	No

I/O Specification

NA

NF EDI Batch

This batch is used to generate NF from EDI staging tables.

fmedinf

Functional Area

NF creation and searching for an item.

Module Affected

fmedinf

Design Overview

This EDI batch runs to create the NF from the entries in the EDI NF tables, in Worksheet status. This NF is not linked to any schedule. Errors can only be viewed in the error log file. A new column in the NF header table displays if the NF has been created manually or through the EDI batch process.

The `fm_edi_doc_detail` table is modified to include a new field, called VPN. This field allows the user to specify the VPN number, if the user is not aware of the RMS item number.

Scheduling Constraints

Table 11–26

Schedule Information	Description
Processing Cycle	Ad-hoc
Scheduling Considerations	Ad-hoc
Pre-Processing	NA
Post-Processing	NA
Threading Scheme	Multi-threaded based on Supplier

Restart/Recovery

The logical unit of work is Supplier and edi_doc_id.

Locking Strategy

NA

Security Considerations

NA

Performance Considerations

NA

Key Tables Affected

Table 11–27   Tables Affected

Tables	Select	Insert	Update	Delete
FM_EDI_DOC_HEADER	Yes	No	Yes	No
FM_EDI_DOC_DETAIL	Yes	No	Yes	No
FM_EDI_DOC_PAYMENT	Yes	No	No	No
FM_EDI_DOC_COMPLEMENT	Yes	No	No	No
FM_EDI_DOC_TAX_HEAD	Yes	No	No	No
FM_EDI_DOC_TAX_DETAIL	Yes	No	No	No
FM_FISCAL_DOC_HEADER	Yes	Yes	Yes	No

Table 11–27   (Cont.) Tables Affected

Tables	Select	Insert	Update	Delete
FM_FISCAL_DOC_DETAIL	Yes	Yes	Yes	No
FM_FISCAL_DOC_COMPLEMENT	Yes	Yes	No	No
FM_FISCAL_DOC_PAYMENTS	No	Yes	No	No
FM_FISCAL_DOC_TAX_HEAD	Yes	Yes	No	No
FM_FISCAL_DOC_TAX_DETAIL	Yes	Yes	No	No
FM_UTILIZATION_ATTRIBUTES	Yes	No	No	No
FM_FISCAL_DOC_TAX_HEAD_EXT	Yes	Yes	No	No
FM_FISCAL_DOC_TAX_DETAIL_EXT	Yes	Yes	No	No
FM_ERROR_LOG	No	Yes	No	Yes

I/O Specification

NA

12

Configuring Security for Web Services

The localized version of the fiscal module of RMS (ORFM) integrates with TaxWeb Tax Rules, a tax engine, through the Retail Tax Integration Layer (RTIL) to take advantage of situations that may arise as a result of the Brazil tax legislations.

The communication between ORFM/RMS, the RTIL, and TaxWeb Tax Rules occurs via the Web Services. The Oracle Software Security Assurance (OSSA) program advocates that the Web Service communication should occur in a secure manner.

The different mechanisms of securing web services are:

- Transport-level security, in which Secure Sockets Layer (SSL) is used to secure the connection between a client application and the Web Service.

- Message-level security, in which data in SOAP message is digitally signed or encrypted.

- Access control security, which specifies which users, groups, and roles are allowed to access Web Services.

OSSA mandates that sensitive data like passwords should be encrypted before storing into the database. Encryption/decryption is the approach adopted for securing data where the original data must be reconstructed from the encrypted data. The data is encrypted and decrypted using a key value and a specific algorithm.

The following functional processes in ORFM/RMS use the tax calculation in which integration is realized via secured Web Services:

- Inbound Nota Fiscal validation

- Outbound Nota Fiscal issuing

- PO tax break up

- Item creation (Purchase/Sales)

- Freight Nota Fiscal

This chapter covers the following topics:

- Transport-Level Security

- Message-Level Security

- Access Control Security

- Implementing Web Service Security

Transport-Level Security

Transport-level security secures the connection between the client application and WebLogic Server with SSL. SSL provides secure connections by allowing two applications connecting over a network to authenticate the other's identity and by encrypting the data exchanged between the applications. Authentication allows a server, and optionally a client, to verify the identity of the application on the other end of a network connection. Encryption makes data transmitted over the network intelligible only to the intended recipient.

Message-Level Security

Message-level security includes all the security benefits of SSL, but with additional flexibility and features. Message-level security is end-to-end, which means that a SOAP message is secure even when the transmission involves one or more intermediaries. The SOAP message itself is digitally signed and encrypted, rather than just the connection. And finally, you can specify that only parts of the message be signed or encrypted.

Access Control Security

For access control security, specify the list of users, groups, or roles that are allowed to access a Web Service. Then, when a client application attempts to invoke a Web Service operation, the client authenticates itself to WebLogic Server, using HTTP, and if the client has the authorization, it is allowed to continue with the invocation. Access control security secures only WebLogic Server resources. This means that if you configure only access control security, the connection between the client application and WebLogic Server is not secure and the SOAP message is in plain text.

Implementing Web Service Security

The user credentials you can use to invoke the Web Service from Oracle Retail applications (ORFM/RMS) are stored in an encrypted format. The user name and password which is provided at the client side is encrypted by using a database procedure with an algorithm initially and stored into the database. At the time of invocation another database procedure is used to decrypt the password and set the authentication parameters internally in the SOAP message. On the server side, the Web Service custom security policy (usernametoken) is attached to the Web Service method. The same user name and password is attached to the usernametoken policy on the server side.

The user name profile tokens are the user name and password associated with a Web Service request and used for authentication and authorization.

Testing Web Service Security

Use the following procedure to test your Web Service security:

1. Configure the user name/password at the server side (WebLogic) and attach the user name/password to the usernametoken policy.

2. Using the encrypt package, store different encrypted user name/password to the database (retail_service_report_url table). The Web Service invocation fails if a wrong password is set up for the invocation.

A

Understanding RTIL WorkManager Settings

This appendix discusses the steps you need to take to configure the RTIL WorkManager.

Configuring RTIL WorkManager

The steps involved in configuring the RTIL WorkManager are as follows:
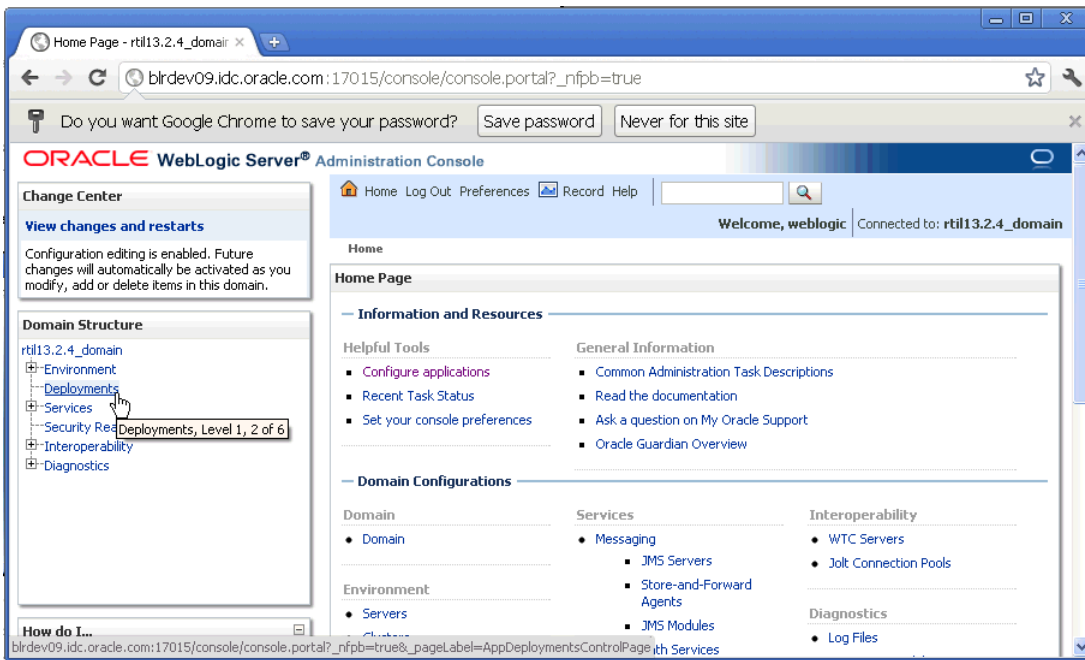
1. Login to Oracle WebLogic Server Administration Console.

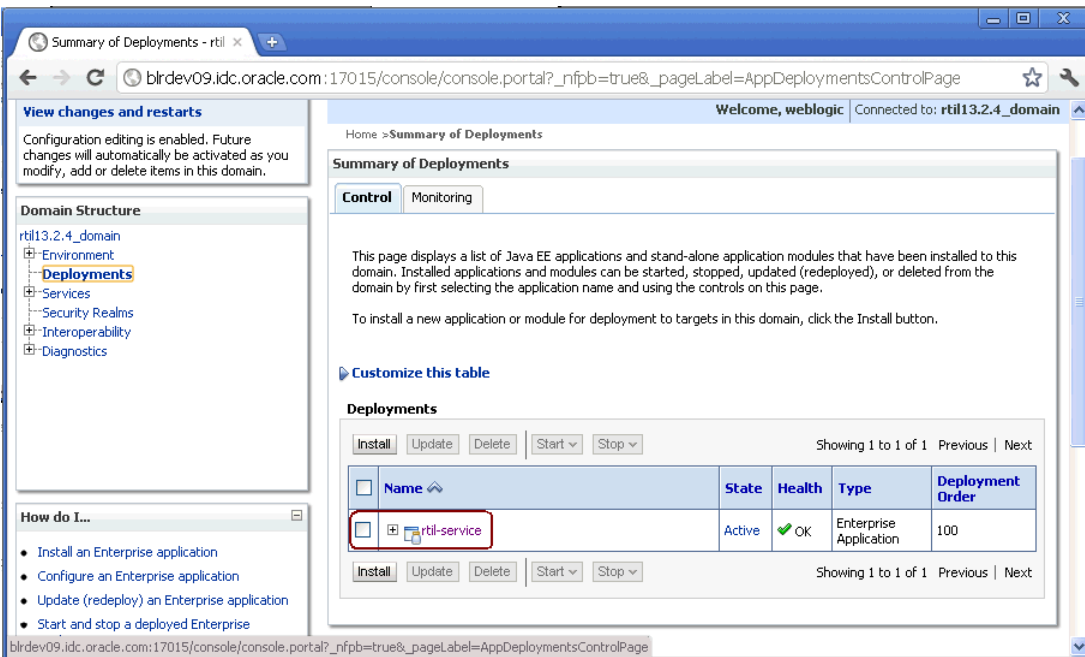*Figure A–1   Oracle WebLogic Server Login*

2. Click **Deployments** in the Domain Structure area of the console.
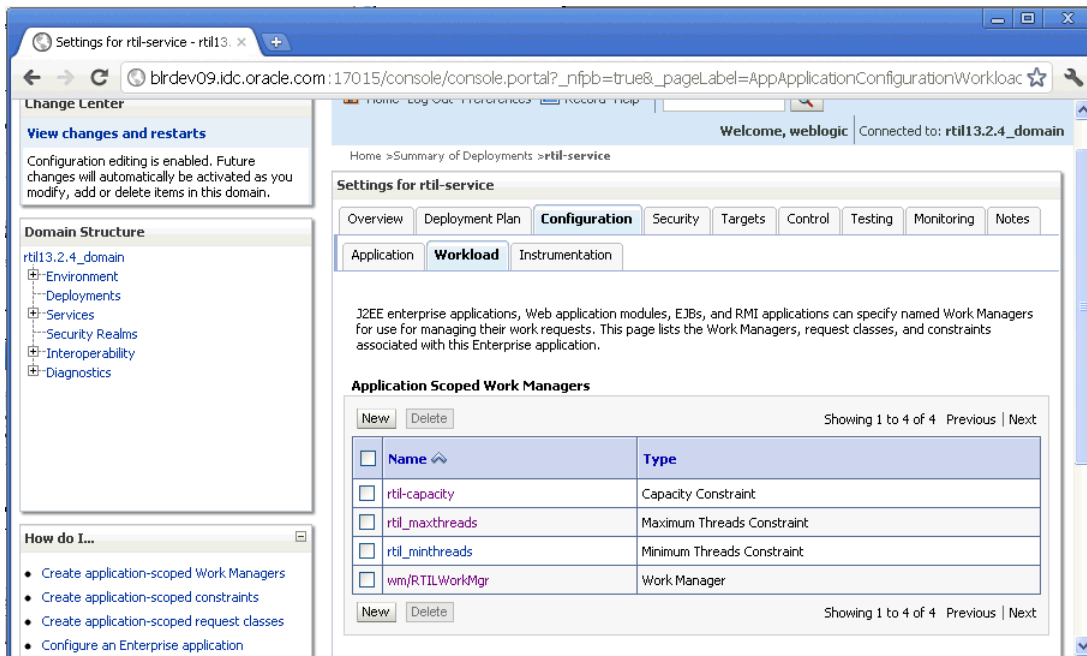
Figure A–2   Deployment

3.   Select **rtilservice** from the Summary of Deployments area of the console.

Figure A–3   RTILService

4.   In the Settings for rtil-service area of the console, from the **Configuration** tab, select **Workload**.

Figure A–4 Workload

5. The Applications Scoped Work Managers area includes rtil_capacity, rtil_ maxthreads, and rtil_minthreads. Select the parameter that you need to modify.
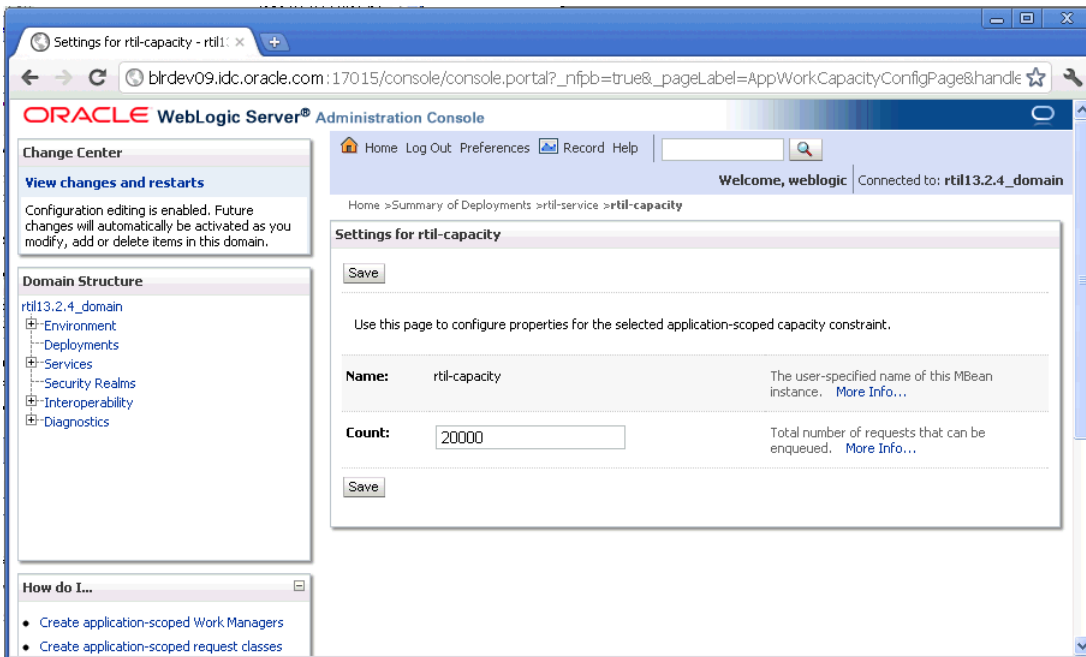
Figure A–5 Applications
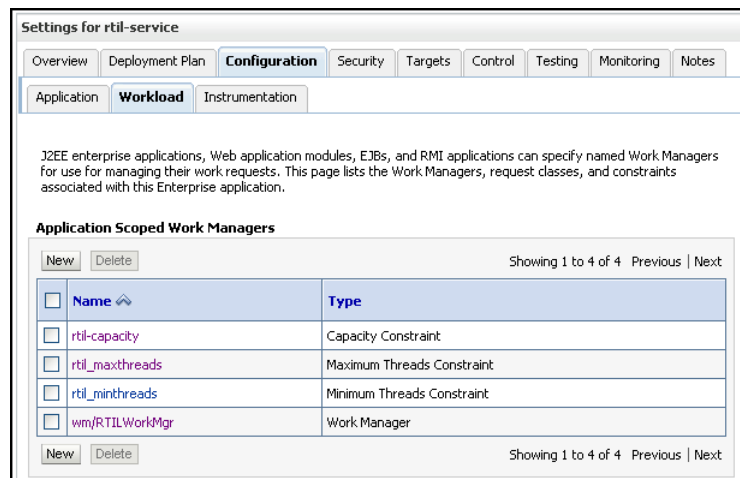
6. In case you want to modify the number of requests that can be queued, click **rtil_ capacity**.

Figure A–6   RTIL Capacity
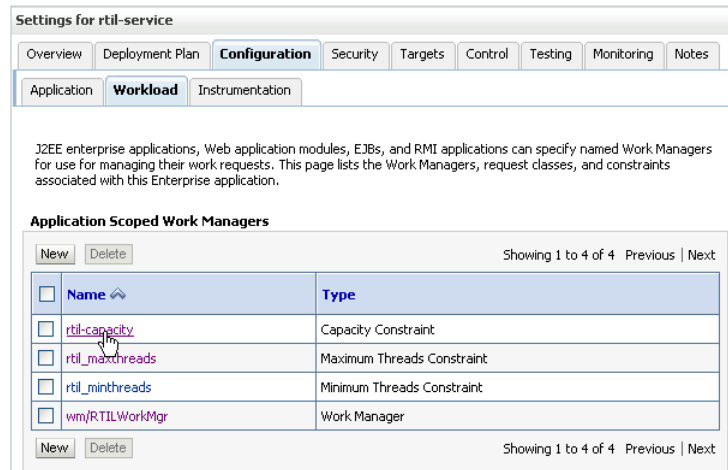
7. In the Settings for rtil_capacity window, enter the total number of requests that can be in the queue, and click **Save**.

Figure A–7   Set RTIL Capacity

8. You are returned to the Applications Scoped Work Managers window. If you want to modify the maximum number of concurrent threads that can execute requests, click **rtil_maxthreads**.

Figure A–8   RTIL Maxthreads

9. In the Settings for rtil_maxthreads area of the window, enter the maximum number of concurrent threads that can execute requests, and click **Save**.