

**Oracle® Retail Fiscal Management/RMS Brazil
Localization**

Implementation Guide

Release 13.2.2

July 2011

Copyright © 2011, Oracle and/or its affiliates. All rights reserved.

Primary Author: Nathan Young

Contributing Author: Divya Begur

Contributor: Selvaprabha Balakrishnan, Srinivas Dharmavaram, Shinod Pampadi, Jeffrey Touhey

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

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

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

U.S. GOVERNMENT RIGHTS Programs, software, databases, and related documentation and technical data delivered to U.S. Government customers are "commercial computer software" or "commercial technical data" pursuant to the applicable Federal Acquisition Regulation and agency-specific supplemental regulations. As such, the use, duplication, disclosure, modification, and adaptation shall be subject to the restrictions and license terms set forth in the applicable Government contract, and, to the extent applicable by the terms of the Government contract, the additional rights set forth in FAR 52.227-19, Commercial Computer Software License (December 2007). Oracle USA, Inc., 500 Oracle Parkway, Redwood City, CA 94065.

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

Oracle is a registered trademark of Oracle Corporation and/or its affiliates. Other names may be trademarks of their respective owners.

This software and documentation may provide access to or information on content, products, and services from third parties. Oracle Corporation and its affiliates are not responsible for and expressly disclaim all warranties of any kind with respect to third-party content, products, and services. Oracle Corporation and its affiliates will not be responsible for any loss, costs, or damages incurred due to your access to or use of third-party content, products, or services.

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

Preface	xiii
Audience	xiii
Related Documents	xiii
Customer Support	xiv
Review Patch Documentation	xiv
Oracle Retail Documentation on the Oracle Technology Network	xiv
Conventions	xiv
1 Oracle Retail Fiscal Management Overview	
What Is Oracle Retail Fiscal Management?	1-1
Functional Assumptions	1-1
What Is the Retail Tax Integration Layer?	1-2
What Is the TaxWeb Tax Rules?	1-3
What Is Nota Fiscal?	1-3
What Is Nota Fiscal Eletronica (NF-e)?	1-3
2 ORFM Implementation Checklist	
Overview	2-1
Prerequisites	2-1
Checklist for Legacy Systems	2-2
Implementation Checklist	2-3
3 RMS Setup for ORFM and Brazil Localization	
Decouple RMS and ORFM Batch	3-1
Dynamic Function Calls	3-1
PL/SQL Package Calls	3-2
PL/SQL Type Inheritance	3-3
Localization Layer	3-3
Accomplishing Decoupling	3-4
Decoupling Batch Programs	3-4
Set Up Fiscal Download	3-4
Set Up Item Attributes	3-4
Set Up Fiscal Attributes	3-5
Set Up Utilization	3-5
Set Up Document Types	3-5

Set Up System Options	3-5
Set Up Reason Codes	3-6
Set Primary Country	3-6
Set VAT Indicator	3-6
Set Default Tax Type	3-6
Verify Localized Indicator	3-6
Set Up Tax Codes	3-7
Set Up Flex Field Attribute	3-8

4 ORFM Integration with Other Applications

Overview	4-1
Nota Fiscal Receiving	4-1
Inbound Process Flow	4-2
Nota Fiscal Issuing	4-3
Outbound Process Flow	4-3
Purchase Orders	4-4
Triangulation	4-5
ORFM and RWMS	4-5
ORFM and SIM	4-5
ORFM Integration with a Financial Application	4-6

5 ORFM Integration – Functional

Purchase Order Receiving	5-2
PO Receiving at a Warehouse	5-3
PO Receiving at a Store	5-5
Transfer Shipment and Receiving	5-6
Transfers	5-6
Transfer Creation	5-6
Return to Vendor (RTV) Shipments	5-9
RTV Shipped from Warehouse	5-9
RTV Shipped from Store	5-11
Inventory Adjustment	5-12
Finishers/Repairing Flow (Two-Legged Transfer)	5-13
Two-Legged Transfer Shipping from RWMS	5-13
Two-Legged Transfer Receiving at RWMS	5-14
Two-Legged Transfer Shipping from SIM	5-15
Two-Legged Transfer Receiving at SIM	5-17

6 ORFM Integration – Technical

RIB-Based Integration	6-1
Web Service-Based Integration	6-1

7 Publication and Subscription API Designs

Secondary ASNOut Publication API	7-1
Business Overview	7-1
Functionality Checklist	7-1

Form Impact.....	7-1
Business Object Records.....	7-1
Package Impact.....	7-2
Business Object ID	7-2
Package Name: FM_SECONDARY_ASNOUT	7-2
Spec File Name: fm_secondary_asnout_sql_s.pls	7-2
Body File Name: fm_secondary_asnout_sql_b.pls	7-2
Package Specification - Global Variables.....	7-2
Function Level Description -	7-2
Trigger Impact	7-4
Trigger Name:	7-4
Trigger File Name:	7-4
Table:	7-4
Message XSD.....	7-4
Table Impact.....	7-4
Design Assumptions.....	7-4
POSchedule Publication API	7-5
Business Overview	7-5
Functionality Checklist.....	7-5
Form Impact.....	7-5
Business Object Records.....	7-5
Package Impact.....	7-5
Business Object ID	7-5
Package Name: FM_SCHED_SUBMIT	7-5
Spec File Name: fm_sched_submit_sql_s.pls.....	7-5
Body File Name: fm_sched_submit_sql_b.pls	7-5
Package Specification - Global Variables.....	7-5
Function Level Description	7-6
Trigger Impact	7-7
Trigger Name:	7-7
Trigger File Name:	7-7
Table:	7-7
Message XSD.....	7-8
Table Impact.....	7-8
Design Assumptions.....	7-8

8 Archiving and Purging Strategy

Archiving and Purging.....	8-1
----------------------------	-----

9 Integration with TaxWeb Tax Rules – Retail Tax Integration Layer

Integration Overview	9-2
Retail Tax Data Model	9-2
Object Structure Overview	9-3
Retail Tax Data Model to TaxWeb Tax Rules Data Model Object Mapping	9-4
RTIL Architecture.....	9-4
High Level Integration View	9-4

RTIL Integration Architecture	9-5
Components Deployment View	9-6
Client Components	9-7
RTIL Configuration	9-7
Logging/Audit	9-7
LoggerAspect.....	9-8
ObjectLoggerAspect	9-8
TimerLoggerAspect:	9-8
User Friendly Exception Messages.....	9-9
Input Parameters Data Mapping and Expected Output Results	9-9
Tax Calculation Scenarios	9-11
Scenario 1 - Inbound NF Validation	9-11
Scenario 2 - Outbound NF Issuing	9-11
Scenario 3 - PO Tax Calculation	9-11
Scenario 4 - Item Creation	9-11
Scenario 5 - Freight NF Calculation.....	9-11
Nota Fiscal Eletrônica (NFe)	9-12
Overview	9-12
NFe Options.....	9-12
NFe Publishing	9-13
NFe Subscription.....	9-13
Sistema Público de Escrituração Digital (SPED)	9-15
SPED File Structure.....	9-15
Overview	9-16

10 Tax Information Maintenance

Integrated Batch Schedule	10-1
Fiscal item reclassification Retail	10-1
Fiscal item reclassification cost	10-2
Tax Law Changes Retail	10-2
Tax Law Changes Cost	10-2
Program Flow	10-3
L10N_BR_EXTAX_MAINT (External Tax Maintenance)	10-5
Fiscal_Item_Reclass_Cost.ksh	10-6
Functional Area	10-6
Module Affected	10-6
Design Overview	10-6
Scheduling Constraints	10-6
Restart/Recovery	10-6
Key Tables Affected	10-7
I/O Specification	10-7
Fiscal_Reclass_Item_Extax_Finish_Retail.ksh	10-7
Functional Area	10-7
Module Affected	10-7
Design Overview	10-7
Scheduling Constraints	10-7
Restart/Recovery	10-8

Key Tables Affected	10-8
I/O Specification	10-8
Fiscal_Reclass_Item_Extax_Setup_Retail.ksh	10-8
Functional Area	10-8
Module Affected	10-9
Design Overview	10-9
Scheduling Constraints	10-9
Restart/Recovery	10-9
Key Tables Affected	10-9
I/O Specification	10-10
Fiscal_Reclass_Item_Process_Retail.ksh.....	10-10
Functional Area.....	10-10
Module Affected	10-10
Design Overview	10-10
Scheduling Constraints	10-10
Restart/Recovery	10-11
Key Tables Affected	10-11
I/O Specification	10-11
Refresh_Extax_Finish_Retail.ksh.....	10-11
Functional Area.....	10-11
Module Affected	10-11
Design Overview	10-11
Scheduling Constraints	10-11
Restart/Recovery	10-11
Key Tables Affected	10-12
I/O Specification	10-12
Refresh_Extax_Future_Cost.ksh	10-12
Functional Area.....	10-12
Module Affected	10-12
Design Overview	10-12
Scheduling Constraints	10-13
Restart/Recovery	10-13
Key Tables Affected	10-13
I/O Specification	10-14
Refresh_Extax_Process_Retail.ksh.....	10-14
Functional Area.....	10-14
Module Affected	10-15
Design Overview	10-15
Scheduling Constraints	10-15
Restart/Recovery	10-15
Key Tables Affected	10-15
I/O Specification	10-15
Refresh_Extax_Setup_Retail.ksh.....	10-15
Functional Area.....	10-15
Module Affected	10-16
Design Overview	10-16
Scheduling Constraints	10-16

Restart/Recovery	10-16
Key Tables Affected	10-16
I/O Specification	10-17

11 Batch Processes

TaxWeb Tax Rules Tax Engine Integration Batch	11-1
Batch Design Summary	11-1
L10nbrfisdnlld (Fiscal Download)	11-1
Functional Area	11-1
Module Affected	11-1
Design Overview	11-1
Scheduling Constraints	11-2
Restart/Recovery	11-2
Locking Strategy	11-3
Security Considerations	11-3
Performance Considerations	11-3
Key Tables Affected	11-3
I/O Specification	11-3
L10nbrfreclsprg (Fiscal Reclassification Purge)	11-3
Functional Area	11-3
Module Affected	11-3
Design Overview	11-3
Scheduling Constraints	11-4
Restart/Recovery	11-4
Locking Strategy	11-4
Security Considerations	11-4
Performance Considerations	11-4
Key Tables Affected	11-4
I/O Specification	11-4
Financial Postings Batch	11-5
Batch Design Summary	11-5
fmfinpost.pc	11-5
Functional Area	11-5
Module Affected	11-5
Design Overview	11-5
Scheduling Constraints	11-5
Restart/Recovery	11-5
Key Tables Affected	11-5
I/O Specification	11-6
fmtrandata.pc.....	11-6
Functional Area	11-6
Module Affected	11-6
Design Overview	11-6
Scheduling Constraints	11-6
Restart/Recovery	11-6
Key Tables Affected	11-6
I/O Specification	11-7

Purging Process Batch	11-7
fmpurge.pc	11-7
Functional Area	11-7
Design Overview	11-7
Scheduling Constraints	11-7
Restart/Recovery	11-7
Key Tables Affected	11-7
I/O Specification	11-8
Localization Batch	11-9
refmvl10entity (Refresh MV MV_L10N_ENTITY).....	11-9
Functional Area	11-9
Module Affected	11-9
Design Overview	11-9
Scheduling Constraints	11-9
Restart/Recovery	11-9
Locking Strategy	11-9
Security Considerations	11-9
Performance Considerations	11-9
Key Tables Affected	11-9
I/O Specification	11-10
SPED Batch	11-10
Import_sped	11-10
Functional Area	11-10
Module Affected	11-10
Design Overview	11-10
Scheduling Constraints	11-10
Restart/Recovery	11-11
Locking Strategy	11-11
Security Considerations	11-11
Performance Considerations	11-11
Key Tables Affected	11-11
I/O Specification	11-11

Preface

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

Related Documents

For more information, see the following documents in the Oracle Retail Fiscal Management Release 13.2.2 documentation set:

- *Oracle Retail Merchandising System Release Notes*
- *Oracle Retail Merchandising System Installation Guide*
- *Oracle Retail Fiscal Management/RMS Brazil Localization Installation Guide*
- *Oracle Retail Merchandising System Operations Guide*
- *Oracle Retail Merchandising System User Guide and Online Help*
- *Oracle Retail Merchandising Data Conversion Operations Guide*
- *Oracle Retail Merchandising System Data Model*
- *Oracle Retail Fiscal Management Data Model*
- *Oracle Retail Merchandising Batch Schedule*
- *Oracle Retail Fiscal Management User Guide*
- *RMS-ReSA User Guide for Brazil Localization*

Also see the documentation library for Oracle Business Intelligence Enterprise Edition at this URL:

http://www.oracle.com/technology/documentation/bi_ee.html

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, 13.2) or a later patch release (for example, 13.2.1). 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.
<i>italic</i>	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.

Oracle Retail Fiscal Management Overview

What Is Oracle Retail Fiscal Management?

Oracle Retail Fiscal Management (ORFM) manages Nota Fiscal 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 Nota Fiscal.

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. 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 Brazil Localization can be accessed by ORFM through direct database lookup. ORFM also integrates with SIM and RWMS using the Oracle Retail Integration Bus (RIB).

Functional Assumptions

ORFM/RMS Brazil localization has the following functional assumptions:

- AUTO_RCV_STORE Indicator on the supplier screen in RMS is never selected.
- 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 (NF) 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.
- The open purchase order quantity is affected only after physical receiving and not after fiscal receiving.
- 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.
- The following fields in the NFe data mapping are out of scope:
 - CT-e is the electronic freight NF which is out of scope of this first release of NFE
 - Fiscal Coupon
 - NF-e Identification- Field #21-DANFE Print format
 - Collection Place
 - Delivery place
 - Imports Declaration
 - Additions
 - B2B
 - New vehicles Detailing
 - Medicine Detailing
 - Weapons Detailing
 - Fuel Detailing
 - IPI
 - IPI – CST 00, 49, 50, 99
 - Importation Taxes
- 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 WAC.

What Is the Retail Tax Integration Layer?

The Retail Tax Integration Layer (RTIL) is implemented as a Java Enterprise Edition application, hosting the canonical Web services and the associated tax service provider adapters. 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 a canonical tax data model based on the configuration. The RTIL hosts vendor-specific connectors which can communicate to the external third-party services with various protocols such as EJB, JMS, SOAP, and HTTP, 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 the 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 Oracle Retail 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 Nota Fiscal?

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 a Nota Fiscal (NF). This document contains all the information related to the transaction. 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.

What Is Nota Fiscal Eletronica (NF-e)?

NF-e is a Brazilian government project tasked with 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 Nota Fiscal Eletrônica (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 DANFE (Documento Auxiliar da Nota Fiscal Eletrônica - Auxiliary Document of the Electronic Invoice) 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.

ORFM Implementation Checklist

This chapter discusses the prerequisites and checklist required for implementing the ORFM module of RMS.

This chapter consists of the following sections:

- [Overview](#)
- [Prerequisites](#)
- [Checklist for Legacy Systems](#)
- [Implementation Checklist](#)

Overview

This chapter contains checklist for standard implementation, and for customer's from legacy systems. The chapter captures mandatory prerequisites and actions required for the successful new implementation of ORFM. The implementation team has to ensure that all prerequisites are in place before initiating set-ups and configurations stated in this checklist.

Prerequisites

The following prerequisites are required before using the ORFM application:

- Installation of RMS, ORFM, RTIL, and TaxWeb Tax Rules – Ensure successful installation of RMS, ORFM, RTIL, and TaxWeb Tax Rules. For more information, refer to the *Oracle Retail Merchandising System Installation Guide*, and *Oracle Retail Fiscal Management and Brazil Localization Installation Guide*.
- Interface between RTIL and TaxWeb Tax Rules – The Web service has to be running between RTIL and TaxWeb Tax Rules. This interface enables retrieval of tax details from the external tax engine, TaxWeb Tax Rules, during item, PO and NF processing. The interface is also used in código fiscal de operações prestações (CFOP) determination for NF processing. For more information, refer to the *Oracle Retail Fiscal Management and Brazil Localization Installation Guide*.
- RIB Adapters (formerly called 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 -- Note 1277421.1.
- Localization Flex Attributes Setup (LFAS) – Set up the localization flexible attributes for company, country, store, warehouse, supplier, outside locations, and

for the item-country combination. Localization Flex Attributes facilitate linking of fiscal attributes to respective entities. For more information, refer to the *Oracle Retail Fiscal Management and Brazil Localization Installation Guide*.

Checklist for Legacy Systems

The checklist for implementing ORFM module from a legacy system is detailed in the following steps:

- Run the data conversion script to load the data from the legacy system.
- Ensure that the following are populated with correct values before making the TaxWeb Tax Rules calls:
 - Future_cost table
 - Currency_rates table are populated with the EFFECTIVE_DATE prior to the transaction date
- Refresh the mv_currency_conversion_rates materialized view.
- Run the scripts manually in the following order:
 1. transform_br_mv_l10n_entity.ksh – This script is used for refreshing the materialized view mv_l10n_entity.
 2. seed_extax_setup_cost.ksh – This module looks at 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).
 3. 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.
 4. 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
 5. seed_extax_setup_retail.ksh – This module looks at 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).
 6. 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.
 7. 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.

8. `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.
9. `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`).
10. `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.
11. `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`.
12. `transform_br_country_attrib_final.ksh` – This script will update 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.
13. `transform_br_upd_nic_cost_type.ksh` – This script is used for updating 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

The checklist for implementing the ORFM module is detailed in the following steps:

- Security Setup – The first step in the implementation process requires finalization of user roles for ORFM. Decide and set up user roles for ORFM. In ORFM user access can be controlled only at the form/menu level. ORFM uses the RMS security infrastructure of RMS. 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, refer to security set up of RMS in *RMS Operations Guide*.
- In the `COUNTRY_ATTRIB` table, ensure the `ITEM_COST_TAX_INCL_IND` parameter to 'Y', and the `DEFAULT_PO_COST` parameter to 'NIC'.
- TaxWeb Tax Rules URL and Server Configure – Capture valid TaxWeb Tax Rules URL and server details for Web service integration of RTIL with TaxWeb Tax Rules. The `RETAIL_SERVICE_REPORT_URL` table in RMS has to be updated with the URL, and server details for TaxWeb Tax Rules has to be validated. The Web service integration of RTIL with TaxWeb Tax Rules facilitates determination and capture of tax and CFOP details.

- 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.
- RMS GTAX Setup – Ensure the system option - Tax Type to 'GTAX', and VAT_IND to 'Y'. This enables global tax functionality.
- Fiscal information download from TaxWeb Tax Rules – Next 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_IPL_CODES
 - CNAE_CODES
 - NOP Codes

For more information, refer to the *ORFM User Guide*, and *RMS-ReSA Brazil Localization User Guide*.

- Document Type Setup – Set the Document Type information in ORFM. 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 the same to one or more utilizations. ORFM requires mandatory association of a document type with an NF. For certain functionalities ORFM uses predefined document types, for instance NFe can only be associated with document type 55.
- Utilization Codes Setup – Set up the different utilization codes for different transaction types in ORFM. Utilization codes control the NF processing logic. Utilization code has to be captured for every inventory transaction. The utilization code parameters control the application behavior. Utilization codes and their functions are as follows:

Table 2–1 Parameters

Utilization Parameter	Description
Complementary NF for Triangulation	Indicates that the utilization has to be used only for complementary NFs from a delivery supplier in a triangulation transaction.
Complementary NF for Freight	Indicates the utilization is of type complementary NF for freight. When this check box is selected, the Complementary NF for Triangulation check box cannot be selected, i.e. these two indicators are mutually exclusive.
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.
Automatic NF Approval	Controls whether or not the outbound fiscal documents are validated, matched and approved automatically. This is relevant only for outbound NFs.

Table 2–1 (Cont.) Parameters

Utilization Parameter	Description
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.

- ORFM System Configuration – Set the ORFM system parameters. System parameters ensure alignment of ORFM functionality 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 2–2 System Options for Tolerance

System Option	Description
TOLERANCES_MANDATORY	Tolerances are applied or not
TOLERANCES_PRIORITY	Tolerances Priority is applied: All, Supplier, Location, Company
CALC_TOL_TYPE	The calculation tolerance type (P/V)
CALC_TOL_VALUE	The calculation tolerance value

- Discrepancy Resolution – 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 2–3 System Options for Discrepancy Resolution

System Option	Description
COST_RESOLUTION_RULE	Default Resolution Action for Cost Discrepancies
QTY_RESOLUTION_RULE	Default Resolution Action for Quantity Discrepancies
TAX_RESOLUTION_RULE	Default Resolution Action for Tax Discrepancies

- Electronic Data Interchange (EDI) – Certain system options need to be set up around EDI and supporting partners.

Table 2–4 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	Default freight type used by the process 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 2–5 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 worksheet status
PURGE_DAYS	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 2–6 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

- 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 2–7 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

Table 2-7 (Cont.) System Options for Document Types

System Option	Description
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 Nfe
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)

- ORFMi/Days – System option has to be defined for 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 2-8 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 – System options need to be set for formats of data fields as per client's requirements.

Table 2-9 System Options for Field Format

System Option	Description
HOUR_FORMAT	Format to represent fields of hour. Must be 'HH24:MI' or 'HH:MI AM'
NUMBER_FORMAT	Value Field Format
PERCENT_FORMAT	Percentage Field Format

Table 2–9 (Cont.) System Options for Field Format

System Option	Description
QTY_FORMAT	Quantity Field Format

- Other Options – A few other important system options affecting system's behavior are provided in [Table 2–10](#).

Table 2–10 Other System Options

System Option	Description
CHECK_APPROVED_DOCS	Variable to check whether it is mandatory to approve all fiscal documents before approving the schedule
DEFAULT_COUNTRY	Default supplier country
DEFAULT_CURRENCY	Default currency code
ORIGIN_CST_IND	Default Origin Code and CST Type for ICMS tax situation
RECOVERABLE_TAX_CST	Value related to the selected CST
DEFAULT_NOP	Default Nature of Operation

- Tolerance Setup – Tolerances have to be set up at cost, quantity (percentage or value set for system/supplier) and calculation level (for rounding or truncation). 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.
- NFe configuration – ORFM enables generation of NFe by location and utilization code. Location has to be identified for NFe issuance for a particular transaction, and for that location NFe parameters have to be configured. The transactions for which NFe 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 will not be controlled in the system and has to be defined by the user.
- 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_CODE 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, you have to ensure that the tax code/VAT codes set up in RMS and ORFM are as available in TaxWeb Tax Rules.
- Fiscal Document Numbering Sequence Setup – The fiscal document number sequencing details has to be set up through the Location Fiscal Number screen 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.
- RMS Setup – The following system options have to be set up in RMS:
 - The COUNTRY_ATTRIB table should have localization_ind - 'Y' for Brazil.
 - Flex fields has to be set to country BR. For the country BR, the fiscal attribute value should be 105.

- Set up the flex fields for all entities (Company, Store, Warehouse, Supplier/Site, Partner, Outside Location, Transfer Entity, Set of Books).
- RMS system options **Allow Auto Shipping/Receiving at Store** is not selected.
- Default location should be set up for Brazil on COUNTRY_ATTRIB table. Default location is set up after creation of the first location. Default location is most likely to be set as the location for which most of the purchasing occurs.
- RMS Validations – The following have to be verified 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' to enable tax functionality.
 - External Invoice Matching Indicator is set to 'Y'.
 - All tax codes in RMS 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 *RMS-ReSA User Guide for Brazil Localization*.

- Brazil Localization Setup – Verify that the table L10N_PKG_CONFIG for decoupling is filled with the entries for Brazil before any transactions happen within ORFM. Entries in the table will enable Brazil localization functionalities. For more information, refer to the *ORFM Implementation Guide*.
- Tax Service Provider Configuration – Ensure that the "tax-service-provider.properties" in RTIL is configured correctly to point to the deployed TaxWeb Tax Rules instance. The configuration ensures valid Web service integration of RTIL with TaxWeb Tax Rules.
- SPED Configuration – 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. It again depends upon the decision of the customer 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.

RMS Setup for ORFM and Brazil Localization

This chapter describes the processes for preparing RMS to use ORFM and Brazil localization.

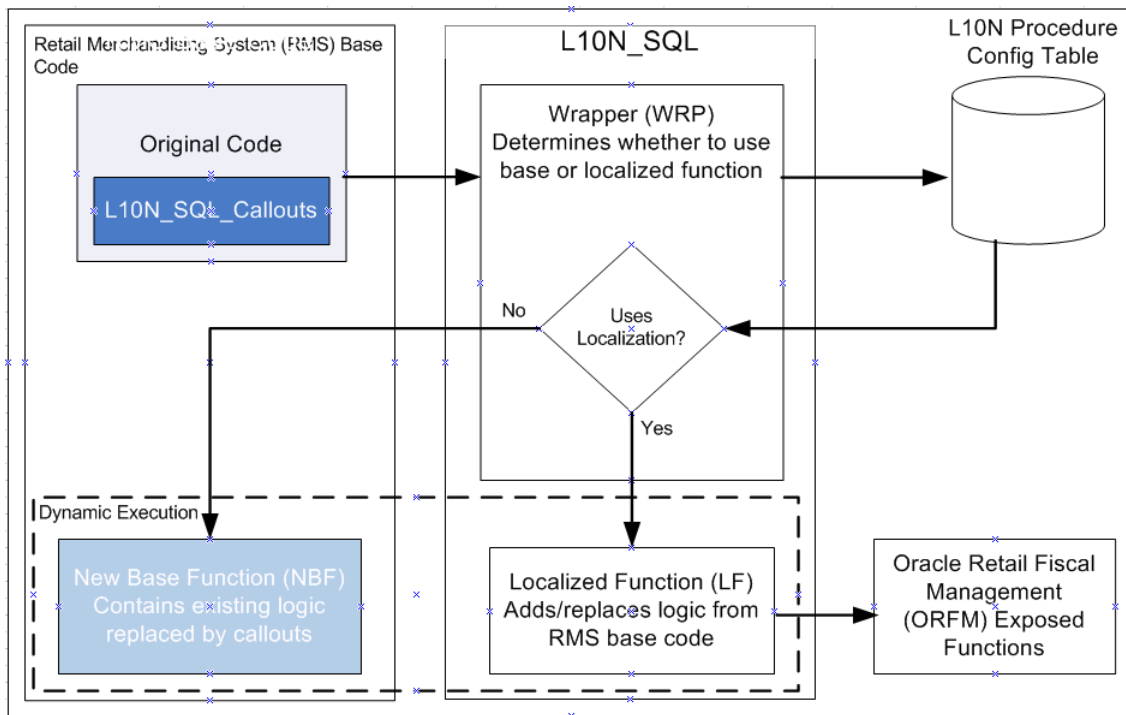
Decouple RMS and ORFM Batch

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:

- **The original code** – This is the base code that does the callout to the localization layer.
- **L10N_SQL wrapper (WRP)** – This function decides whether to use the base code or the localized version of the code.
- **L10N configuration table** – This table contains the procedure name to be executed per localized instance.
- **New base function (NBF)** – This function contains the existing logic from the original code replaced by callouts to the localization layer.
- **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

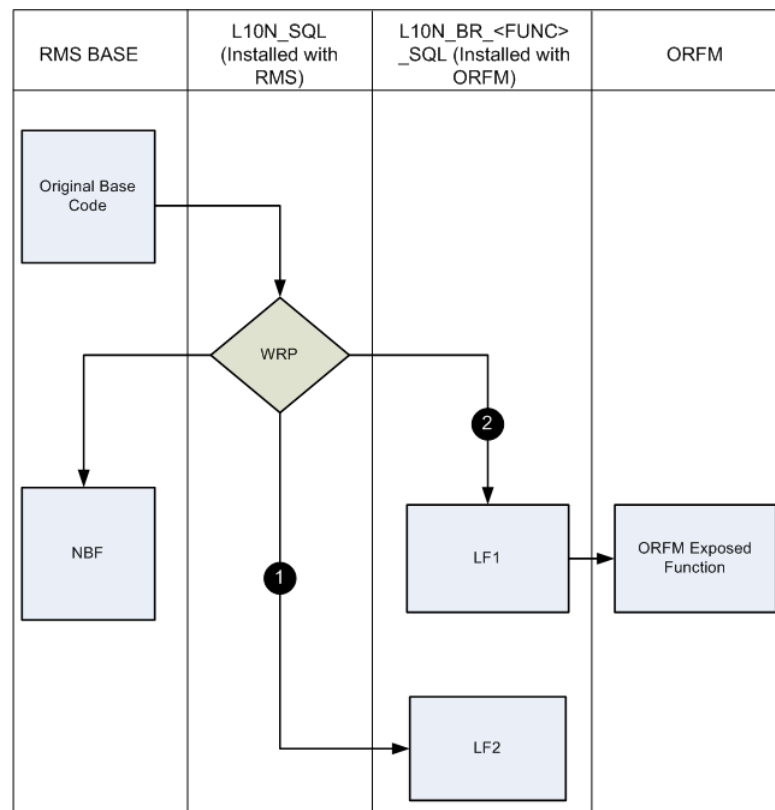
In this approach, the in/out parameters to be passed to functions should be encapsulated in object types. The object types follow PL/SQL Type Inheritance (a supertype is created from which several subtypes will be derived). The supertype is a generic object that can be used across functions. Derived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

Set Up Fiscal Download

Fiscal Download holds data such as the Nature of Operation (NOP), ncm_codes, 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. There is a batch available to download the data.

Set Up Item Attributes

Item 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.

Set Up Fiscal Attributes

Fiscal attributes 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.

Navigation is available for the setting up fiscal attributes. This is available from the Options menu of Warehouses, Stores, Outside Locations, Partners, Supplier, Transfer Entity, and Set of Books.

Set Up 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 contains.

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 various parameters are:

- Complementary NF for Triangulation
- Complementary NF for Freight
- ICMS-ST recovery
- Allow Receiving
- Automatic NF Approval
- Choose NF

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

Set Up 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 Ids 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.

Set Up System Options

ORFM System Parameters, including the Default Utilization codes need to be set up. 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.

Set Up 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 will fail.

Set Primary Country

To use ORFM, the Primary Country must be set to **BR**. This can be done during the RMS installation process. See the *Oracle Retail Merchandising System Installation Guide* for more information.

Set VAT Indicator

To use ORFM, VAT_IND must be set to **Y**. This can be done during the RMS installation process. See the *Oracle Retail Merchandising System Installation Guide* for more information.

Set Default Tax Type

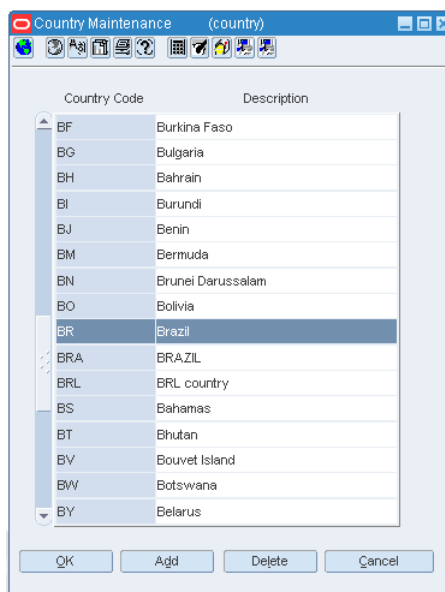
To use ORFM, the Default Tax Type must be set to **GTAX**. This can be done during the RMS installation process. See the *Oracle Retail Merchandising System Installation Guide* for more information.

Verify Localized Indicator

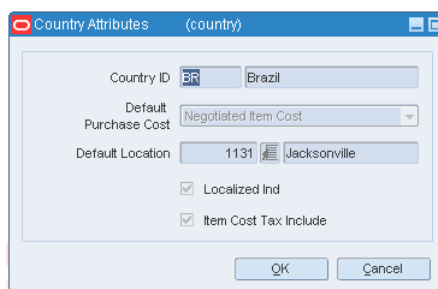
The Localized indicator check box indicates whether the Brazil patch is installed. It is auto-checked by the system if the Brazil patch is installed.

Use the following procedure to verify the Localized Indicator:

Navigate: From the **RMS Start Menu**, select **Control**, then **Setup**. In the Contents of Setup window, select **Country**, then **Edit**. The Country Maintenance window opens.

Figure 3–3 Country Maintenance Window

1. Select **BR** and click the **Options** menu.
2. From the Options menu, select **Attributes**. The Country Attributes window opens.

Figure 3–4 Country Attributes Window

3. Verify that the Localized Ind check box is selected and click **OK**. You are returned to the Country Maintenance window.
4. Click **OK** to save your changes and close the window.

Set Up 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.

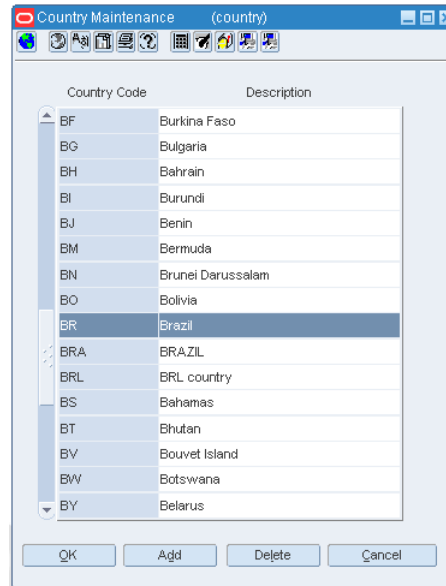
This table is populated from the back end and does not have any navigation available.

Set Up Flex Field Attribute

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

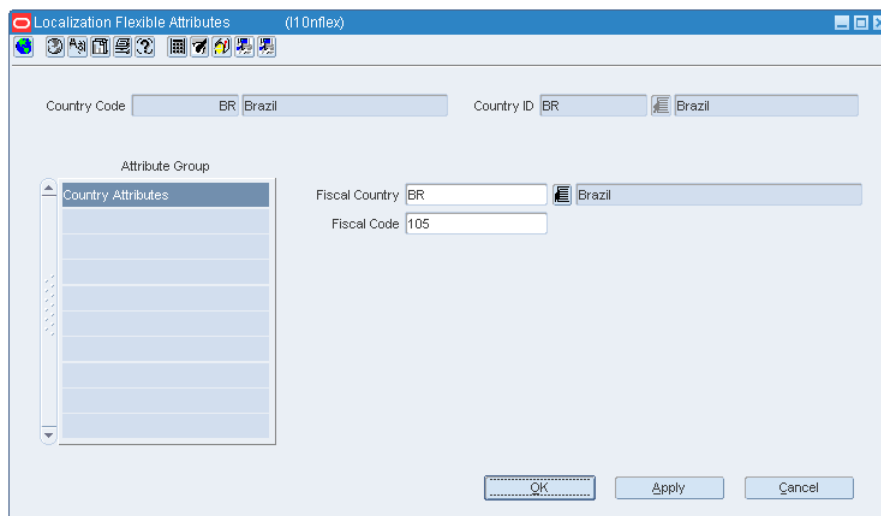
Navigate: From the main menu, select Control > Setup > Country > Edit. The Country Maintenance window opens.

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.

ORFM Integration with Other Applications

This chapter provides a functional overview of how ORFM integrates with other systems.

Overview

The first section in this chapter provides a diagram illustrating the various Oracle Retail products and databases with which ORFM interacts, as well as the overall dataflow among the products. The accompanying explanations are written from a system-to-system perspective, illustrating the movement of data.

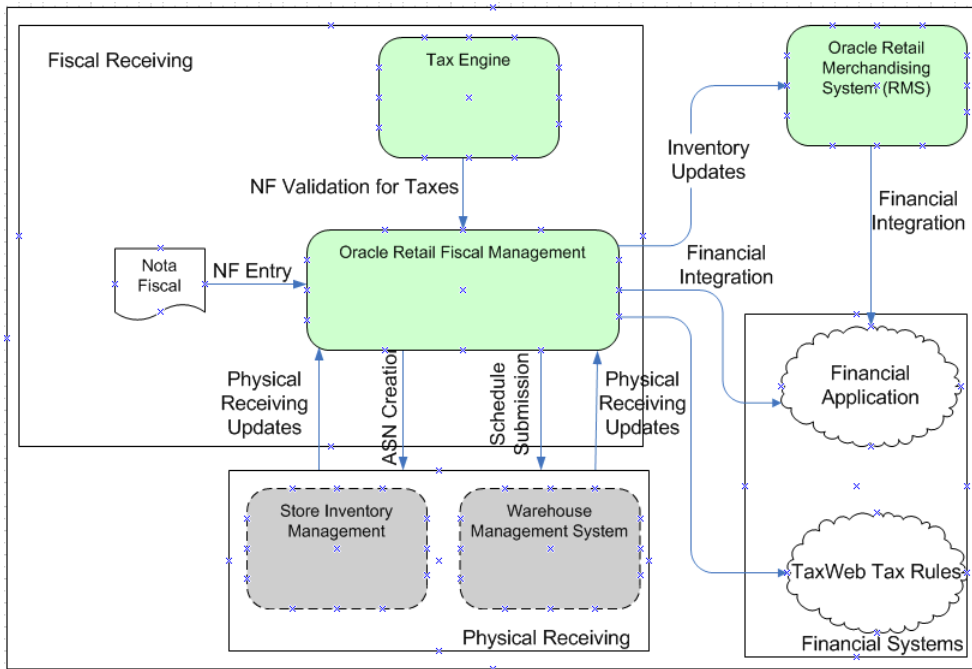
The Nota Fiscal receiving (inbound) and issuing (outbound) are controlled within ORFM and all integration with RMS, RWMS, and SIM is based on the physical movement of the products.

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.

The following diagram outlines the ORFM business process for inbound operations:

Figure 4–1 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: More than one NF can be linked to a schedule.

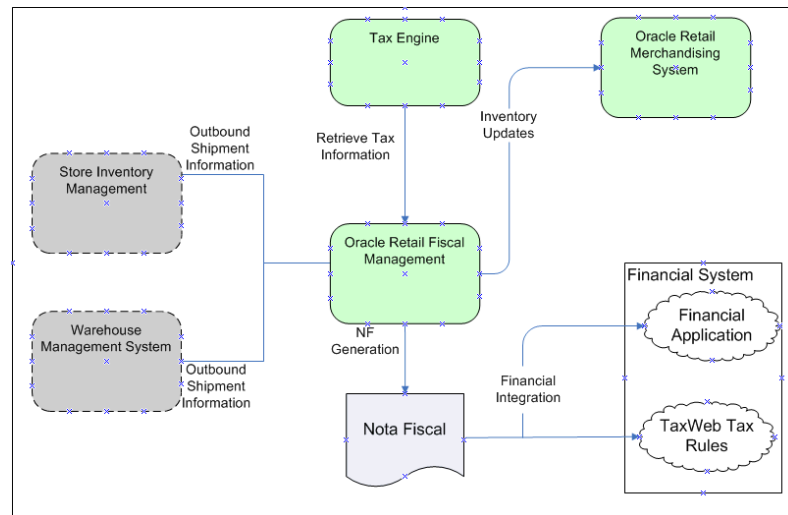
3. After NF entry, validate the NFs. In the validation process, the application checks for data integrity. Match the NF with the requisition documents in Retail Merchandising System (RMS). This process is called as Fiscal Receiving.
4. If the NF and the PO does not match, the NF is in discrepancy. You 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.

- 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.

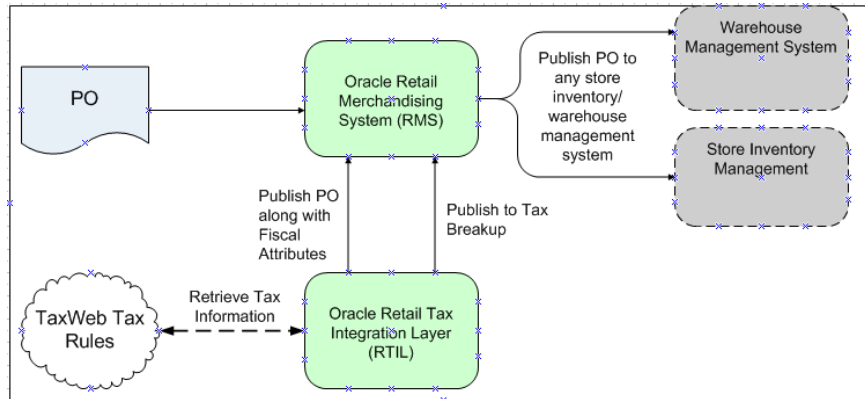


Outbound Process Flow

- Enter the outbound shipment information in SIM or RWMS and send it to ORFM.
- ORFM sends the outbound shipment information to the tax engine. The tax engine figures the tax information and returns it to ORFM.
- Create the Nota Fiscal based on the outbound shipment information and the tax information.
- After NF processing is complete, ORFM calls RMS to update inventory and WAC.

Purchase Orders

In Brazil, receiving without a valid purchase order (PO) number is not allowed, because fiscal receiving must precede physical receiving. For this reason, a direct store delivery (DSD) without a purchase order cannot be received in SIM. A purchase order created in RMS should be sent to RWMS and SIM through RIB as per the base process flow. The purchase order created should flow to the TaxWeb Tax Rules through the Retail Taxation Integration Layer (RTIL) along with the fiscal attributes of the supplier(s), receiving location(s), and the items. The TaxWeb Tax Rules should calculate the tax breakup and send it back to RMS through RTIL.



1. Create a purchase order in RMS.
2. Publish the PO. Retrieve the Brazil tax information and include it in the PO. Send the PO with the Brazil tax information to RTIL.
3. RTIL publishes the tax breakdown of the PO.
4. The PO is published in SIM or 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

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.

ORFM and RWMS

The 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.

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 a Financial Application

Note: There is no integration available between ORFM and any financial application. All the NF transaction data is posted into financial tables in ORFM.

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 to 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.

ORFM Integration – Functional

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. See "Overall Solution Landscape," which depicts the inbound operations process flow.

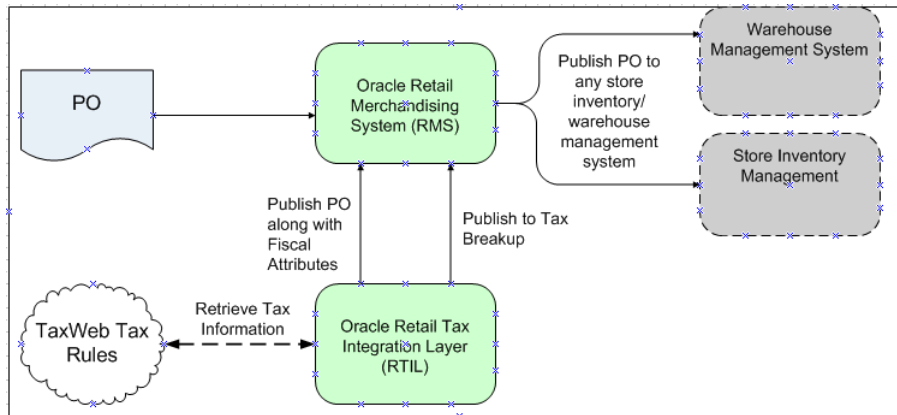
If Brazil localization is enabled, RMS inventory is not updated immediately when any inventory is shipped out from RWMS or SIM. The inventory movement information should be sent to ORFM for the generation of outbound NF. After approval of the outbound NF, ORFM should send the inventory updates to RMS.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

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

Purchase Order Receiving

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 5–1 Purchase Order Receiving



A purchase order goes through the following process to be 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 breakup 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

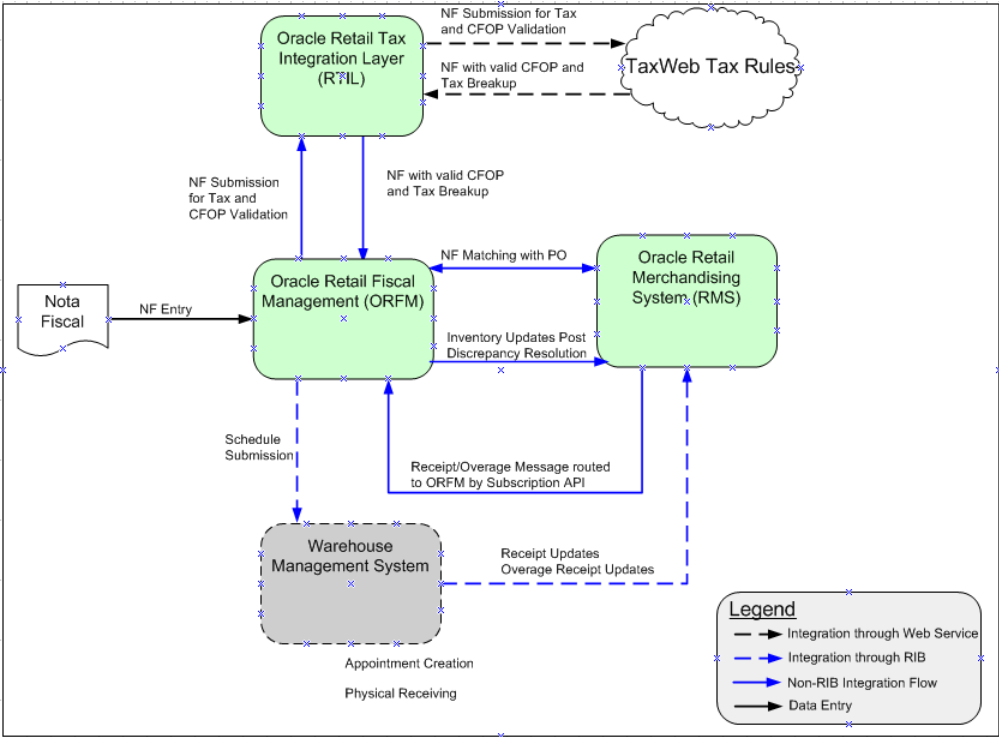
PO Receiving at a Warehouse

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 5-2 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.

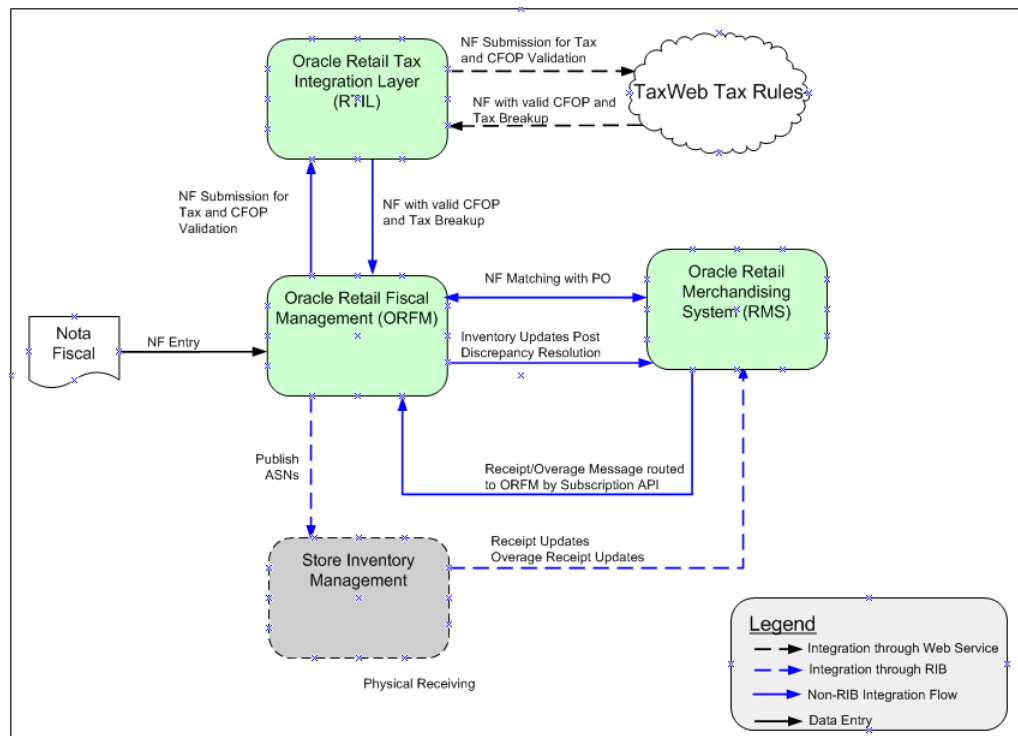
For the received items, the taxes are stripped from the unit cost to arrive at the tax-exclusive cost, additional cost components (if any) are added to calculate the actual landed cost, and the inventory and WAC updates are sent to RMS along with the NF number. Other than the cost and NF, all other information 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 5–3 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 two 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.

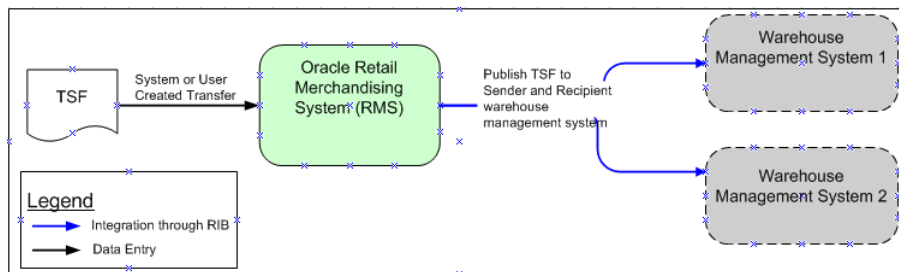
Transfers

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.

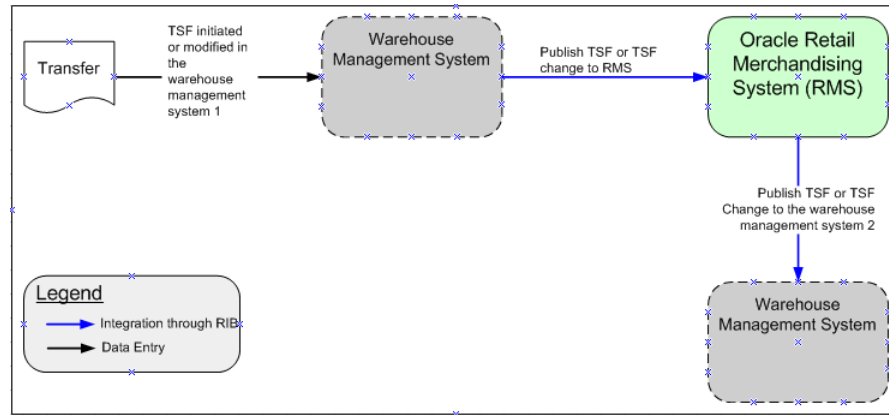
Figure 5–4 Warehouse to Warehouse Transfer Creation



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 5-5 Warehouse to Warehouse Transfer

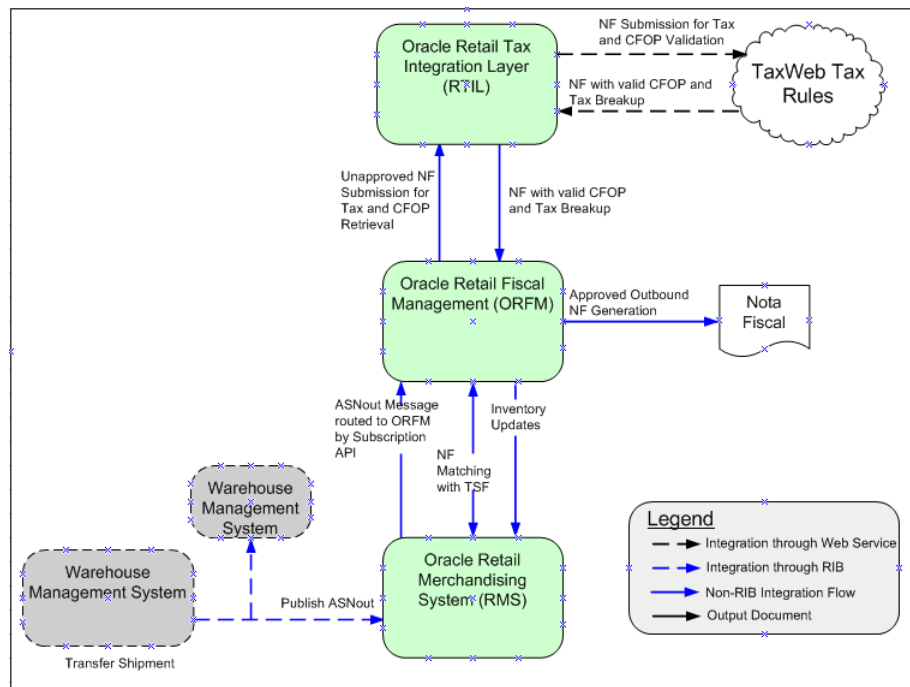


Transfer Shipment at Source

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 5-6 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 ASNout 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.

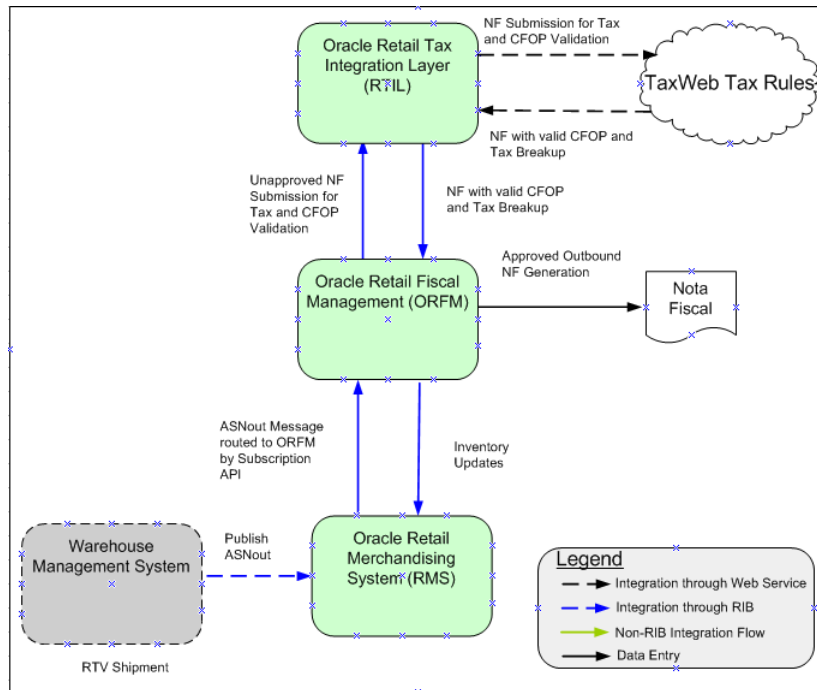
Using the information in ASNout message, ORFM should generate the outbound RTV NF. If there is reference to the inbound NF (through which the stock was received), the tax information is also retrieved from the inbound NF. If no inbound NF reference is required, the system should get the tax information 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 5-7 RTV Shipped from Warehouse



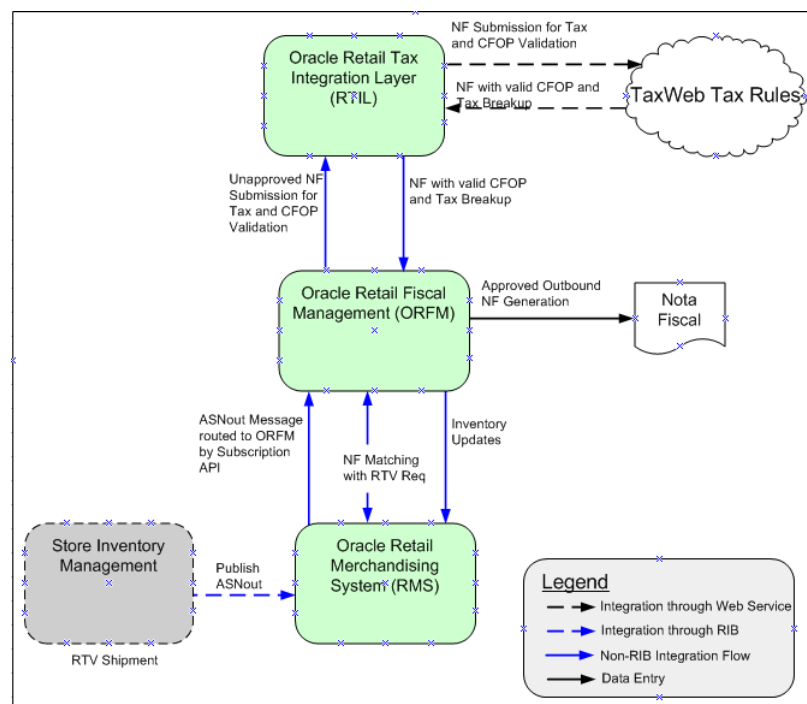
RTV Shipped from Store

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

Using the information in the ASNout 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 5–8 RTV Shipped from Store



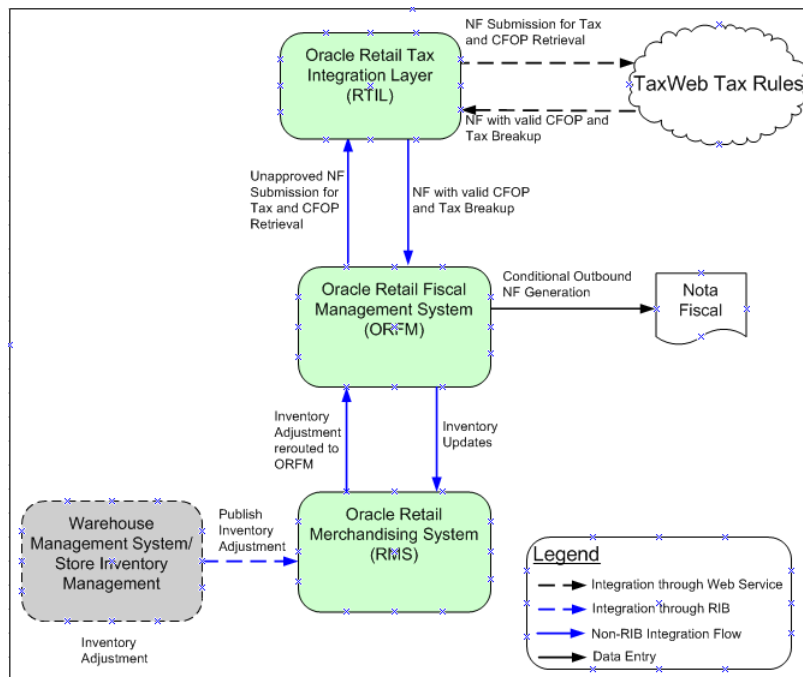
Inventory Adjustment

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 reason 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 5–9 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.

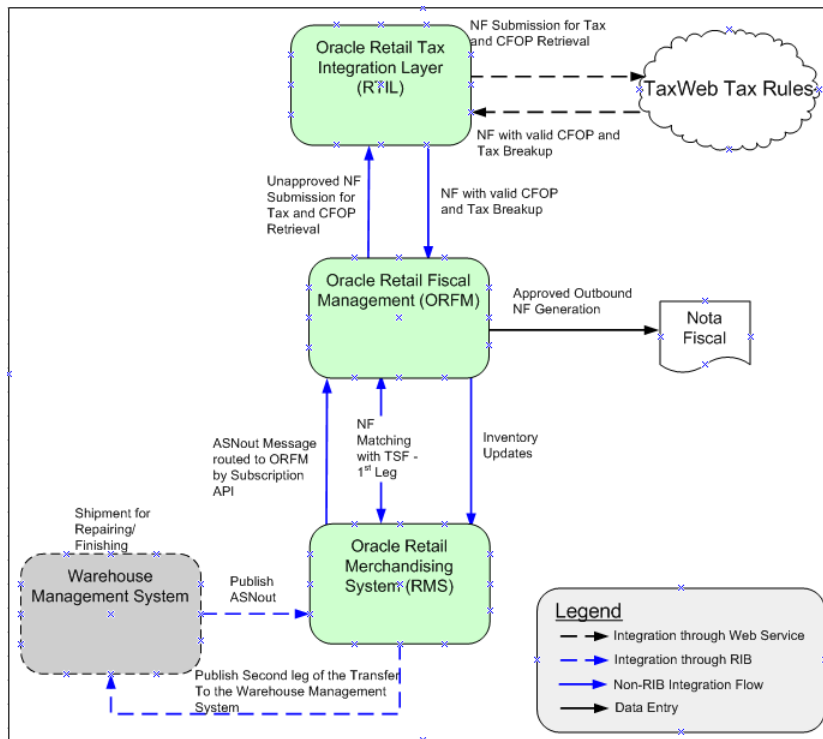
RWMS, as the physical owner of the inventory, has better visibility to the goods that need to be repaired. Therefore, a repairing stock order can be initiated in RWMS as well. As a stock order with the type of Repairing is created in RWMS, it flows to RMS through RIB. RMS, on consuming the message, creates 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 5–10 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:

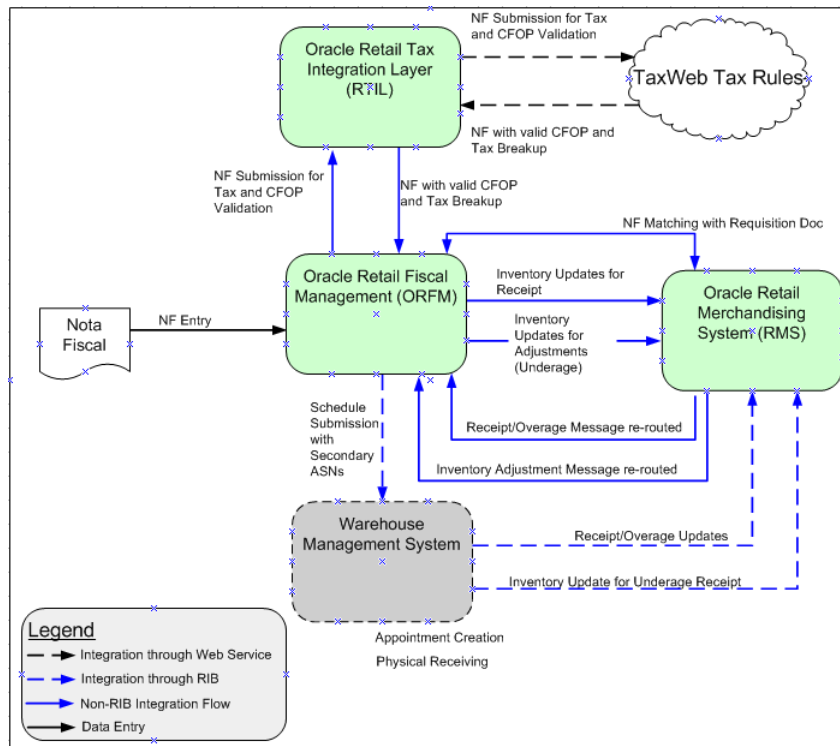
- ASN Number (ORFM should generate the ASN# and maintain the next up sequence number for it).
- Requisition Number (the secondary transfer number)
- Receiving Warehouse Number
- Distinct Item Ids per transfer leg two
- Consolidated Quantities per item-requisition
- UoM
- Schedule number

- 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 5–11 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.

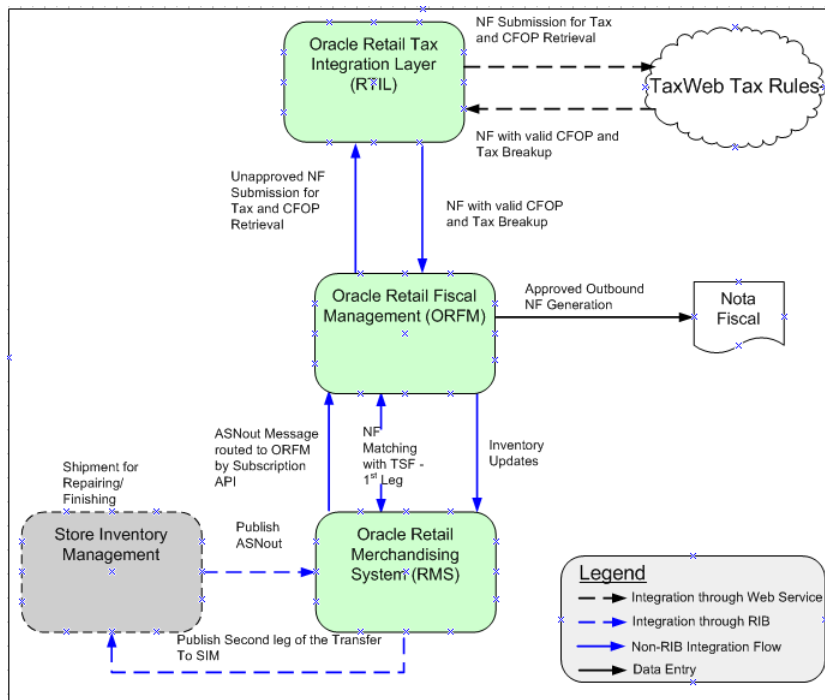
Because SIM is the physical owner of the inventory, it has better visibility to the goods that need to be repaired. Therefore, a repairing RTW can be initiated in SIM as well. As a RTW with the type set to Repairing is created in SIM, it should flow to RMS through RIB. RMS, on consuming the message, 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 5-12 SIM Two-Legged Transfer - Shipping



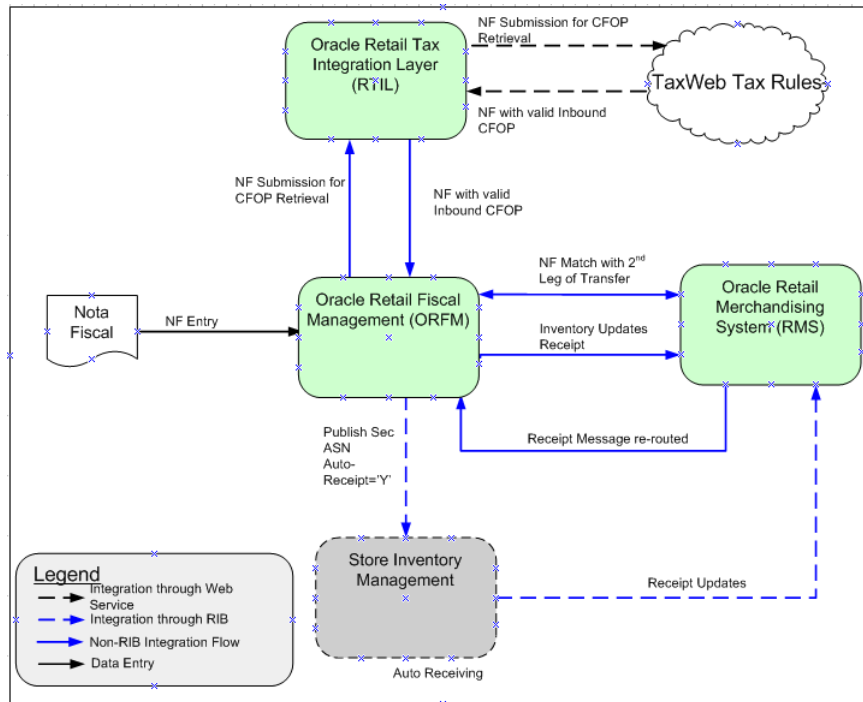
Two-Legged Transfer Receiving at SIM

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 5-13 SIM Two-Legged Transfer - Receive



ORFM Integration – Technical

This chapter is divided into the following two sections that address the ORFM methods of integration:

- ["RIB-Based Integration"](#) on page 6-1
- ["Web Service-Based Integration"](#) on page 6-1

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.

Web Service-Based Integration

ORFM integrates with RTIL through webservice based interfaces. See ["RTIL Architecture"](#) on page 9-4 for additional information.

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).

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

This message will be published from ORFM 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 7-1

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 –

ADDTQ – 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 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 – Same as GETNXT 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 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 (i.e. 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 GETNEXT.

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:

N/A

Trigger File Name:

N/A

Table:

N/A

Message XSD

Here is the filename 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 7-2

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

Table Impact

Table 7-3

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

This message is published from ORFM while submitting a schedule based on 'PO' to RWMS for receiving. RWMS subscribes to this message in 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 7-4

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 i.e. 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 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 – Same as GETNXT and 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 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 (i.e. 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 i.e. ORFM and location i.e. 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:

N/A

Trigger File Name:

N/A

Table:

N/A

Message XSD

Here is the filename that correspond 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 7-5

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

Table Impact

Table 7-6

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.

Archiving and Purging Strategy

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.

Archiving and Purging

To enter a Nota Fiscal (for a PO, TO, or RTV) a schedule must be created 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 active tables. In addition, all the NFs that are inactive for a schedule are purged. An inactive schedule is also purged from the active table and the data is stored in the history table.

Integration 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.

The Retail Tax Rules Integration Layer (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. In 13.2.2, 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.

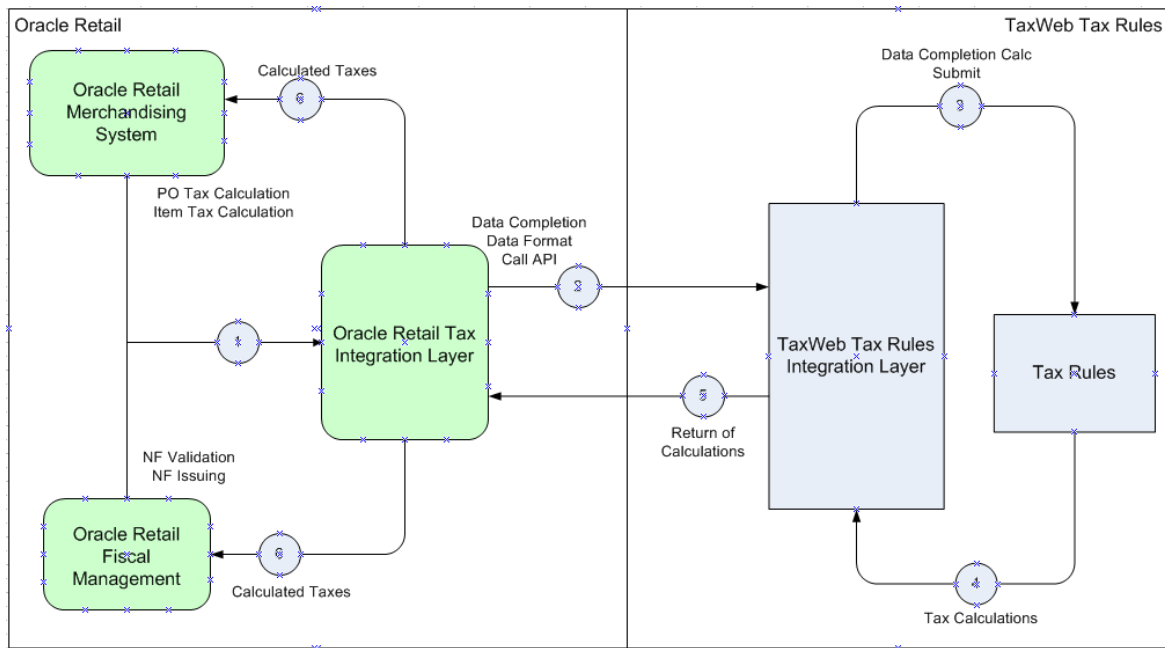
The following processes in ORFM/RMS use the tax calculation integration:

- 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.

Integration Overview

Figure 9–1 Oracle Retail/TaxWeb Tax Rules Integration



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.

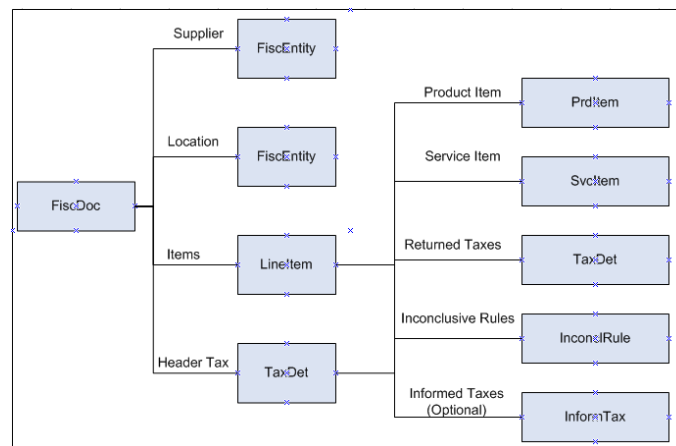
The layout of TaxWeb Tax Rules solution is based on the NF layout and has the following structure:

- 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)

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 9–2 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).

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 9-1

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

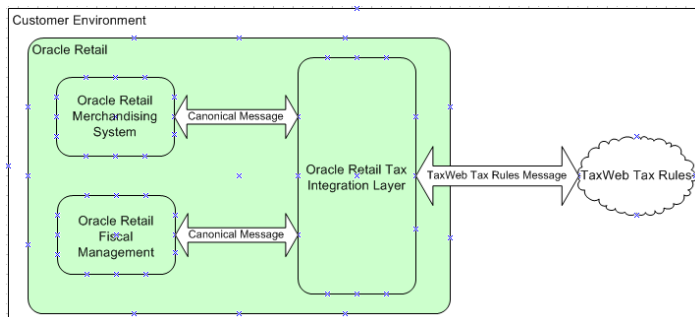
RTIL Architecture

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

Figure 9-3 RTIL High Level Integration

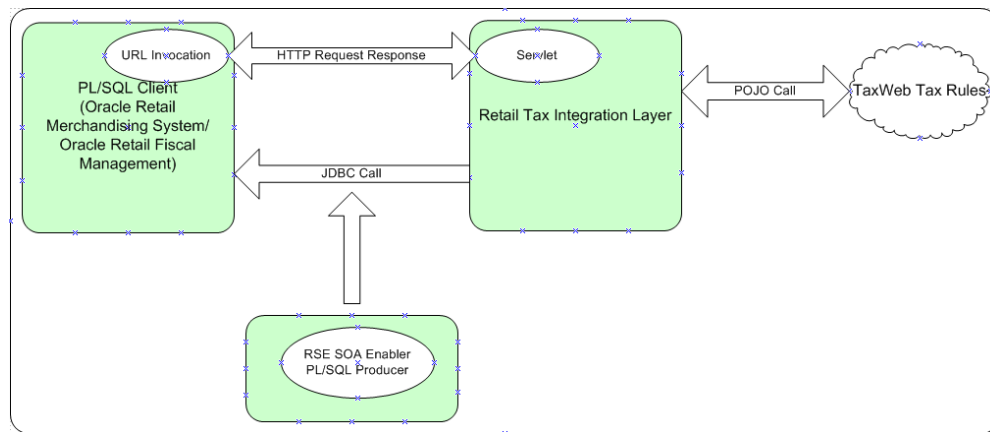


The integration layer in Oracle Retail does the following.

- 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.

RTIL Integration Architecture

Figure 9–4 RTIL Integration Architecture

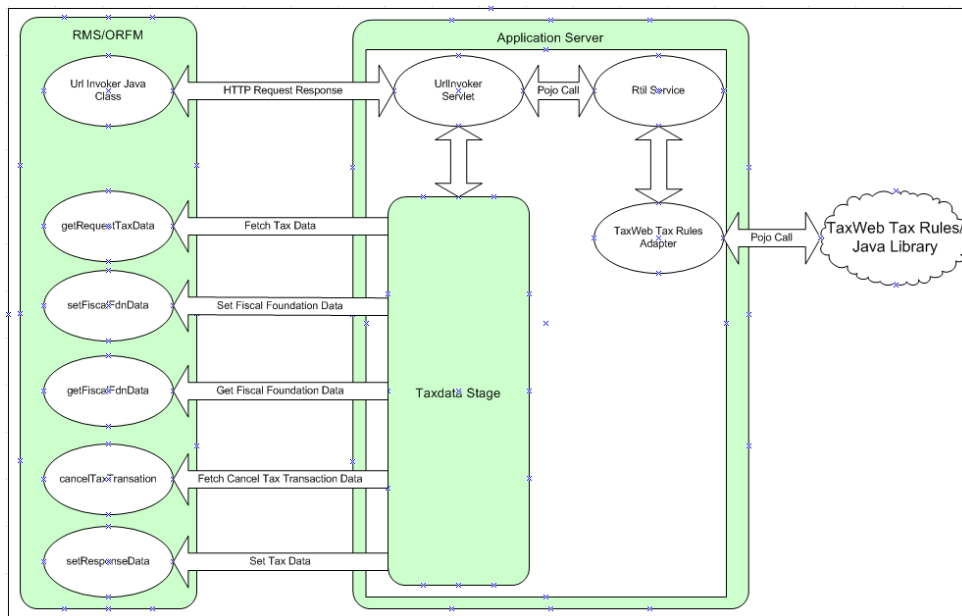


The above diagram illustrates the high level components and its interactions along with the run time and the design time view of the system. The Retail Service Enabler Utility is used to generate the necessary PL/SQL producer and its associated artifacts to fetch/set tax staged data from the database.

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 9-5 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.

For 13.2.2, 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, webserviceGateway, and adapter classes. This aspect can be used to obtain a timing profile of the method executions.

These aspects are declared via 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

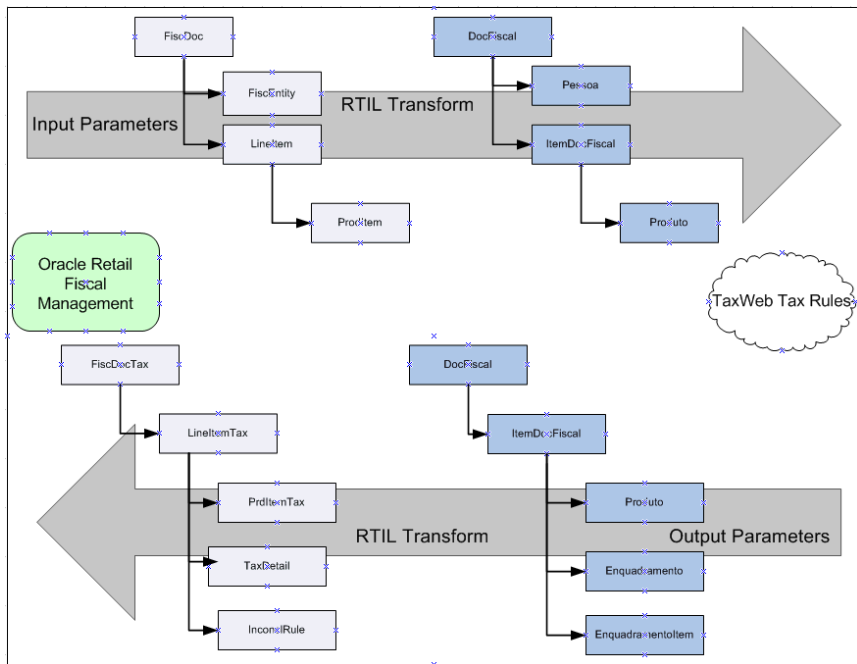
RTIL provides some flexibility in configuring user-friendly error messages based on the exception conditions encountered. The `ExceptionMessage.properties` file contains the list of system-generated exception messages in RTIL, which can be further mapped to the user messages required by the customer. 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 9–6 Input x Output Parameters



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 Validation

The inbound NF validation happens for NF receiving when ORFM calls the validate function. The necessity to validate a NF happens when an entire NF is input in ORFM, and the system must validate the informed taxes against the calculated. Informed taxes in ORFM come at the header level for manual NFs and also at the detail level for automatic NF (EDI or NFE).

Scenario 2 - Outbound NF Issuing

The NF issuing process can happen in ORFM in two different ways. The NF can be generated via EDI tables, as the transfers, RTVs and other outbound movements that are generated from RWMS and SIM, or the NF can be generated manually via ORFM, as the automatic return NF, rural producer NF and RMA. For both cases, the taxes are not informed and the tax calculation needs to come from TaxWeb Tax Rules.

For the EDI NF generation, the batch that calculates the taxes in RMS through a call to the ORFM tax engine will be changed to call the new API with TaxWeb Tax Rules. The NF must be already created in the fiscal doc tables with the correspondent number, because the NF number, NF ID and NF item ID are necessary to call the tax API. The process that calls the tax calculation is the same Validation process of ORFM screens.

Scenario 3 - 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 4 - 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. In the Oracle Retail integration layer, the PO is formatted into the API layout. There are multiple API calls depending on the delivery location set on the PO. For each location, there is one call because the tax calculation API has the NF layout, and only one origin and destination is allowed per NF.

Scenario 5 - Freight NF Calculation

Freight NF has special characteristics in terms of tax calculation. It does not have any items, so the tax mapping considers default data.

Nota Fiscal Eletrônica (NFe)

Nota Fiscal Eletrônica (NFe) 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 NFe 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 NFe 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.

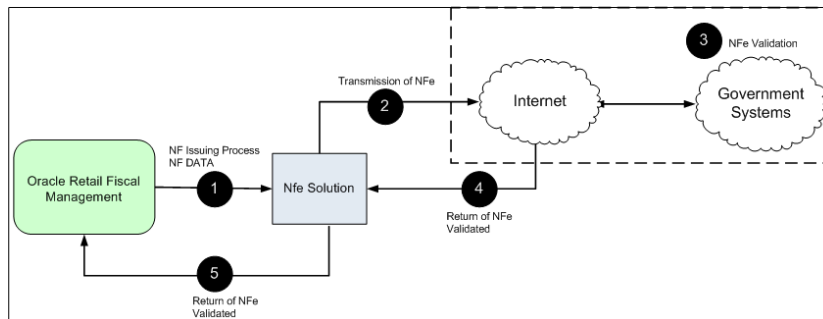
To follow the goods, a graphic representation of the NFe is printed. The DANFE (Documento Auxiliar da Nota Fiscal Eletrônica-Auxiliary Document of the Electronic Invoice) is printed in a common paper, one copy that highlights the access key for consultation of the NFe in the internet and a bi-dimensional bar code which facilitates the capture and confirmation of information of the NFe 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 NFe. It has the access code of the NFe which allows its owner to confirm the real existence of the NFe in RFB environment (Receita Federal Brasileira-Brazilian Federal Tax Authority) or the SEFAZ web site.

Overview

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

Figure 9–7 NFe Issuing Solution Approach



NFe Options

ORFM allows the generation of the NFe by location and utilization code. A system option is used to identify the location that will issue the NFe and for which transaction, such as transfers, intercompany transfers, and RTVs.

Considering that the transactions where the NFe 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 system variables 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.

One location may use NFe for transfers but not for RTVs. While NF creation, ORFM considers this parameterization in order to enter in the NFe flow.

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

NFe Publishing

The FM_STG_NFE staging table for NFe contains the fiscal doc id and its status for the Java Adapter to fetch those records, which are in "NFe Pending" or "NFe Corrected" or "NFe Canceled" state and submit them to TaxWeb Tax Rules. The EVENT_ID field contains the sequence in which the NFe messages are published to Mastersaf. The fiscal_doc_no, series_no, cnpj and justification fields are used to allow Mastersaf's NFe product to interface the fiscal document information to SEFAZ without scanning through the object source again.

There are eight status codes to capture NFe flow between ORFM, Mastersaf, and SEFAZ.

- NFe_P - Fresh NFe document waiting to be picked by Mastersaf.
- NFe_X - Corrected NFe document waiting to be picked by Mastersaf.
- NFe_C - Canceled NFe document waiting to be picked by Mastersaf.
- C_A - Mastersaf updates the staging table with this status when NFe is successfully canceled.
- N_A - Mastersaf updates the staging table with this status when NFe is successfully nullified.
- NFe_I - Intermediate flag that is updated by Mastersaf 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 NFe from SEFAZ.
- E - Erroneous NFe from SEFAZ due to data/transmission errors (in any situation; emission, cancel or nullify).

NFe Subscription

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

When the NFe gets approved successfully or processed with errors, Mastersaf makes a call to the ORFM packaged procedure FM_MS_NFE_SQL.CONSUME to update the NFe details into the respective ORFM tables. If there are any errors then it inserts the NFe transaction history table with error_id and error_description.

The consume procedure contains the following parameters:

Table 9–2 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 9–3 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 9–4 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 NFe submission to SEFAZ.

When the NFe is approved by SEFAZ without any errors, Mastersaf 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 NFe 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 Mastersaf's Java integrator monitor for polling, Mastersaf 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 Mastersaf 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 NFe flow between ORFM, Mastersaf, and SEFAZ. Such error codes are predefined by SEFAZ and do not require the user to correct anything. Mastersaf 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 via the internet. Similar to the NFe, 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:

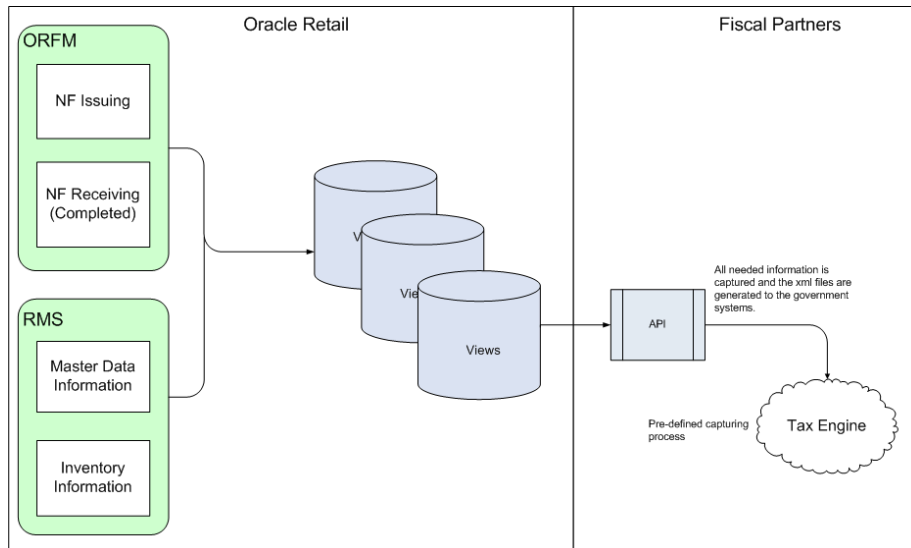
- Block 0: Opening, Identification and References
- Block C: Fiscal Documents I - Merchandise (ICMS/IPI)
- Block D: Fiscal Documents II - Services (ICMS)
- Block E: Fiscal counting of ICMS and IPI
- Block H: Physical Inventory
- Block 1: Other information
- Block 9: Control and Closing of the Digital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

Overview

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

Figure 9–8 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.

Tax Information Maintenance

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

When new TaxWeb Tax Rules are received perform the following steps to install the updated rules:

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.

After the RTIL WebLogic server is running, ensure that the following step is performed:

- When a new Tax Web jar is loaded, L10N_BR_EXTAX_REFRESH_CONFIG.REFRESH_NEEDED should be set to 'Y'. This will trigger updates to the tables in RMS that contain data provided by (or derived from) the external tax engine (TaxWeb Tax Rules).

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 up 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 up 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 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.

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 up 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 up 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 will 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

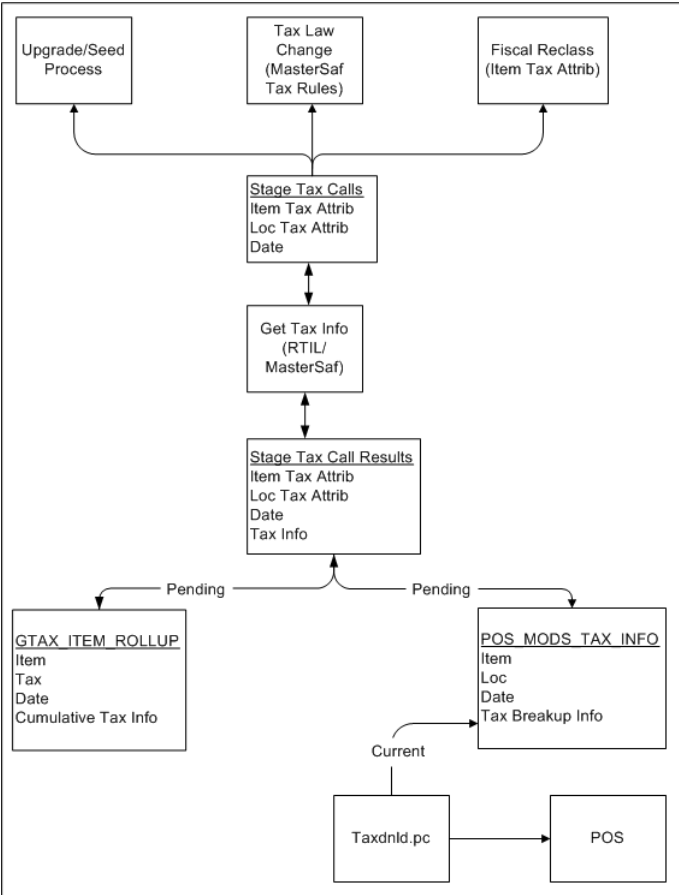


Figure 10–2 Cost Tax Flow

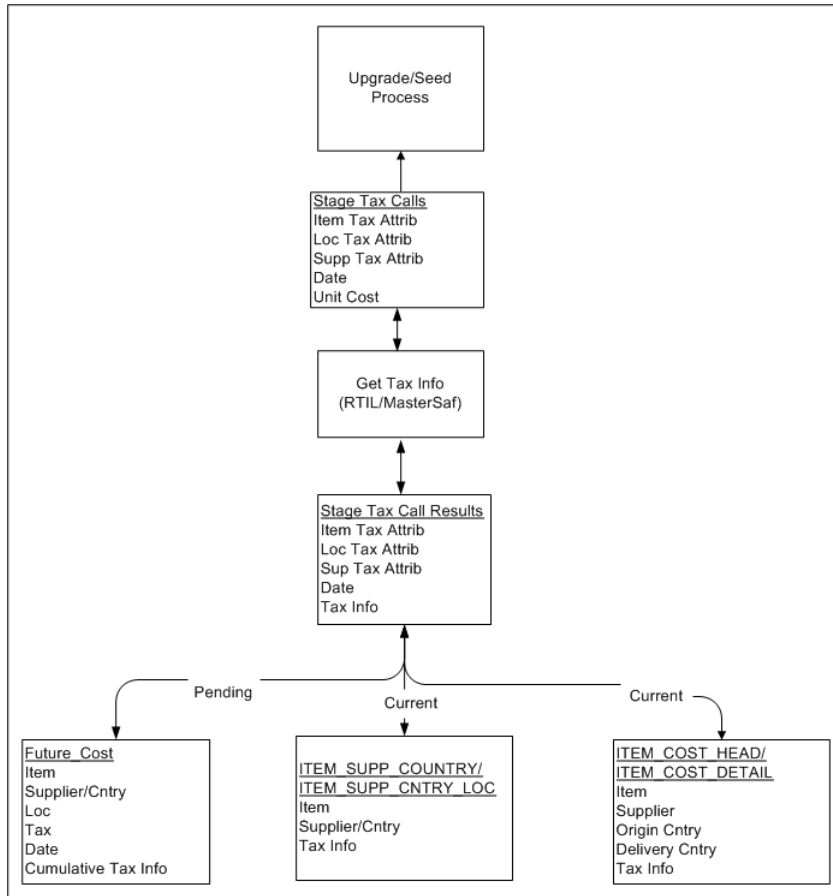
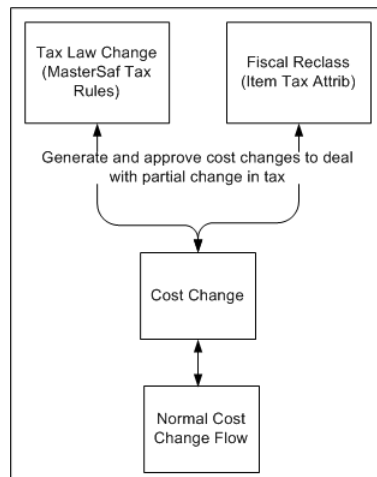


Figure 10–3 Cost Tax Flow Maintenance



L10N_BR_EXTAX_MAINT (External Tax Maintenance)

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, 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 find out from TaxWeb Tax Rules that sales tax will be moving from 10% to 12% in San Paulo in two weeks the pending state tables will be updated as soon this information is known. The current state tables will not be 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 stores in RMS tables, these tables have to contain the factors that determine how tax is calculated changes.

The following three factors drive retail tax recalculations:

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

The following five 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	N/A

Restart/Recovery

N/A

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
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

I/O Specification

N/A

Fiscal_Reclass_Item_Extax_Finish_Retail.ksh

This module picks up 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 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	N/A

Restart/Recovery

N/A

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	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
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

I/O Specification

N/A

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	N/A

Restart/Recovery

N/A

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
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	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

Table 10–6 (Cont.) Tables Affected

Tables	Select	Insert	Update	Delete
L10N_ATTRIB	Yes	No	No	No
L10N_BR_FISCAL_RECLASS	Yes	No	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

I/O Specification

N/A

Fiscal_Reclass_Item_Process_Retail.ksh

This module picks up 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.

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

N/A

Key Tables Affected**Table 10–8 Tables Affected**

Tables	Select	Insert	Update	Delete
RETAIL_SERVICE_REPORT_URL	Yes	No	No	No

I/O Specification

N/A

Refresh_Extax_Finish_Retail.ksh

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

Functional Area

TaxWeb Tax Rules Integration

Module Affected

REFRESH_EXTAX_FINISH_RETAIL.KSH

Design Overview

This module picks up 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.

The work is done in the L10N_BR_EXTAX_MAINT_SQL package.

Scheduling Constraints**Table 10–9 Scheduling Constraints**

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

Restart/Recovery

N/A

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

I/O Specification

N/A

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 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.

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

N/A

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
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

Table 10–12 (Cont.) Tables Affected

Tables	Select	Insert	Update	Delete
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

I/O Specification

N/A

Refresh_Extax_Process_Retail.ksh

This module picks up 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.

Functional Area

TaxWeb Tax Rules Integration

Module Affected

REFRESH_EXTAX_PROCESS_RETAIL.KSH

Design Overview

This module picks up the staged groups placed on 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
Threading Scheme	None

Restart/Recovery

N/A

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

N/A

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 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).

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
Scheduling Considerations	Phase 1
Pre-Processing	
Post-Processing	refresh_extax_process_retail.ksh
Threading Scheme	N/A

Restart/Recovery

N/A

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	No	Yes	No	No
L10N_BR_TAX_CALL_STAGE_ROUTING	Yes	Yes	No	No
RETAIL_SERVICE_REPORT_URL	Yes	No	No	No
L10N_BR_TAX_CALL_STAGE_FSC_FDN	Yes	No	No	Yes
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

Table 10–16 (Cont.) Tables Affected

Tables	Select	Insert	Update	Delete
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	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

I/O Specification

N/A

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.

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)

This batch program facilitates 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 cut off 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 table will be 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	N/A
Post-Processing	N/A
Threading Scheme	N/A

Restart/Recovery

N/A

Locking Strategy

N/A

Security Considerations

N/A

Performance Considerations

N/A

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

N/A

L10nbrfreclsprg (Fiscal Reclassification Purge)

This batch is used 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: <ul style="list-style-type: none"> ▪ Fiscal_reclass_item_extax_setup_retail.ksh ▪ Extax_process_retail.ksh ▪ Fiscal_reclass_item_extax_finish_retail.ksh
Pre-Processing	<ul style="list-style-type: none"> ▪ Fiscal_reclass_item_extax_setup_retail.ksh ▪ Extax_process_retail.ksh ▪ Fiscal_reclass_item_extax_finish_retail.ksh
Post-Processing	N/A
Threading Scheme	Threading based on reclassification id.

Restart/Recovery

N/A

Locking Strategy

N/A

Security Considerations

N/A

Performance Considerations

N/A

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

N/A

Financial Postings Batch

These batches are used 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

Rolling up of 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	N/A
Post-Processing	N/A
Threading Scheme	N/A

Restart/Recovery

N/A

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

Table 11–6 (Cont.) Tables Affected

Tables	Select	Insert	Update	Delete
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

N/A

fmtrandata.pc**Functional Area**

Financial Postings

Module Affected

fm_trandata_posting_sql

Design Overview

Computation of the Transaction 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	N/A
Post-Processing	N/A
Threading Scheme	N/A

Restart/Recovery

N/A

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

N/A

Purging Process Batch**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.

Scheduling Constraints**Table 11–9 Scheduling Constraints**

Schedule Information	Description
Processing Cycle	Ad-hoc
Scheduling Considerations	Ad-hoc
Pre-Processing	N/A
Post-Processing	N/A
Threading Scheme	N/A

Restart/Recovery

N/A

Key Tables Affected**Table 11–10 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

Table 11–10 (Cont.) Tables Affected

Tables	Select	Insert	Update	Delete
fm_fiscal_doc_tax_head_hist	No	Yes	No	No
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

I/O Specification

N/A

Localization Batch

refmvl10entity (Refresh MV MV_L10N_ENTITY)

Functional Area

N/A

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, COUNTRY_ATTRIB table.

Scheduling Constraints

Table 11-11

Schedule Information	Description
Processing Cycle	Ad-hoc
Scheduling Considerations	N/A
Pre-Processing	N/A
Post-Processing	N/A
Threading Scheme	N/A

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

N/A

Security Considerations

N/A

Performance Considerations

N/A

Key Tables Affected

Table 11-12 Tables Affected

Tables	Select	Insert	Update	Delete
ADDR	Yes	No	No	No

Table 11–12 (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

N/A

SPED Batch**Import_sped****Functional Area**

SPED

Module Affected

Import_SPED.ksh

Design Overview

This batch will insert 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 'C'losed 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 to run this batch job on a daily basis due to performance impacts. This batch has a pre-dependency on the ORFM Financial postings batch - FMFINPOST.PC, 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 11–13**

Schedule Information	Description
Processing Cycle	Ad-hoc
Scheduling Considerations	This program should run only after the successful completion of FMFINPOST.PC.
Pre-Processing	N/A
Post-Processing	N/A

Table 11–13 (Cont.)

Schedule Information	Description
Threading Scheme	N/A

Restart/Recovery

N/A

Locking Strategy

N/A

Security Considerations

N/A

Performance Considerations

N/A

Key Tables Affected**Table 11–14 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

N/A

