

**Oracle® Retail Xstore Suite 21.0/Oracle® Retail
Merchandising Suite 22.1.401.0**

Implementation Guide

Release 21.0

F72943-01

October 2022

Copyright © 2022, Oracle and/or its affiliates. All rights reserved.

Primary Author: Gerlinde Rust

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

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

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

U.S. GOVERNMENT END USERS: Oracle programs, including any operating system, integrated software, any programs installed on the hardware, and/or documentation, delivered to U.S. Government end users are "commercial computer software" pursuant to the applicable Federal Acquisition Regulation and agency-specific supplemental regulations. As such, use, duplication, disclosure, modification, and adaptation of the programs, including any operating system, integrated software, any programs installed on the hardware, and/or documentation, shall be subject to license terms and license restrictions applicable to the programs. No other rights are granted to the U.S. Government.

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

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

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

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

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**TM 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**TM 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

| | |
|---|------|
| Send Us Your Comments | xi |
| Preface | xiii |
| Audience..... | xiii |
| Documentation Accessibility | xiii |
| Related Documents | xiii |
| Customer Support..... | xiv |
| Review Patch Documentation | xiv |
| Improved Process for Oracle Retail Documentation Corrections | xiv |
| Oracle Retail Documentation on the Oracle Help Center (docs.oracle.com) | xv |
| Conventions | xv |
| 1 Overview | |
| Inbound: Integration Options with Xstore Office | 1-2 |
| Outbound: Integration Options with Xstore Office | 1-2 |
| 2 Data Flow from Merchandising to Xstore | |
| Conceptual Data Flow | 2-1 |
| Using RMFCS/RPCS Direct Services..... | 2-1 |
| Using OCDS..... | 2-2 |
| Using Flat Files | 2-3 |
| Xstore Office Deployments..... | 2-5 |
| Using RMFCS/RPCS direct services or OCDS..... | 2-6 |
| Using Flat Files | 2-6 |
| Xstore Dataloader..... | 2-7 |
| Using RMFCS/RPCS direct services or OCDS..... | 2-7 |
| Using Flat Files | 2-8 |
| File Sorting | 2-8 |
| File Loading Dependency..... | 2-9 |
| File Iteration and Transformation | 2-9 |
| Persisting into the Database | 2-10 |
| Phased Rollout of Stores | 2-10 |
| Using RMFCS/RPCS direct services or OCDS (Cloud Only) | 2-10 |
| Using Flat Files | 2-10 |

| | |
|--|------|
| Manual Refresh of an Xstore Database | 2-11 |
| Using RMFCS/RPCS direct services or OCDS | 2-11 |
| Using Flat Files | 2-13 |
| 3 Transaction Flow from Xstore to Sales Audit | |
| Conceptual Data Flow | 3-1 |
| Technical Implementation - Sales Audit REST Integration | 3-3 |
| Sales Audit REST Xstore Broadcaster | 3-3 |
| Sales Audit REST Endpoints | 3-4 |
| Scheduled Job to Query Sales Audit Rejected Transactions | 3-4 |
| Xadmin UI to View and Republish Sales Audit Rejected Transactions..... | 3-4 |
| Technical Implementation - RTLog Files Integration | 3-5 |
| RTLog File Xstore Broadcaster | 3-5 |
| Xstore RTLog Generator | 3-5 |
| Sales Audit saimptlog/i | 3-5 |
| 4 Webservice Integration Configuration | |
| Xstore Office: Direct Service Integration Communication | 4-1 |
| Merchandising Configuration | 4-1 |
| POM Configuration | 4-1 |
| Xstore Office: OCDS Integration Communication | 4-2 |
| Xstore Office: Omnichannel Data Service Configuration Options | 4-2 |
| Merchandising Chain to Xstore Organization Mapping | 4-3 |
| Integration Database Tables | 4-4 |
| Enabling and Disabling Data Types | 4-4 |
| Download: Immediate vs Store Close | 4-4 |
| "NOT ON FILE" Items | 4-4 |
| 5 Integration Considerations | |
| Foundation Data | 5-1 |
| Seed Data | 5-1 |
| Transaction Details..... | 5-1 |
| Multi-Line Text..... | 5-2 |
| Currency Exchange Rates | 5-2 |
| Stores..... | 5-2 |
| Merchandise Hierarchy | 5-3 |
| Items | 5-3 |
| Merchandise Items | 5-3 |
| Non-Merchandise Items..... | 5-3 |
| Kit/Pack Items..... | 5-4 |
| Differentiators..... | 5-4 |
| Product Restrictions..... | 5-4 |
| Related Items | 5-4 |
| Other Item Attribute Notes..... | 5-5 |
| Tax | 5-5 |
| Value Added Tax (VAT) | 5-5 |

| | |
|--|------|
| US Sales Tax | 5-5 |
| Inventory | 5-6 |
| Serialized Inventory | 5-6 |
| Customer Orders | 5-6 |
| In-Store Orders | 5-6 |
| Recognition of a Sale | 5-7 |
| Pricing | 5-7 |
| Multi-Unit Pricing | 5-7 |
| Promotions | 5-7 |
| Sales Audit | 5-8 |
| Store Data Configuration | 5-8 |
| Register-level Balancing | 5-8 |
| Sales Person | 5-8 |
| Tender Types | 5-8 |
| Coupons | 5-9 |
| File Format | 5-9 |
| Employee IDs | 5-9 |
| | |
| 6 RTLog Generator On-Premise | |
| Configuration | 6-1 |
| Deployment | 6-5 |
| WebLogic Cluster Setup | 6-7 |
| Deployment of the RTLog Generator Application on a Cluster | 6-18 |
| Security Configuration | 6-21 |
| Container Level Security | 6-21 |
| Transport Level Security | 6-24 |
| Complete the Security Configuration | 6-26 |
| | |
| 7 RTLog Generator Cloud | |
| RTLog Generator Cloud | 7-1 |
| Configuration | 7-1 |
| Integration | 7-1 |
| Updating Mapping Configuration | 7-1 |
| Updating Format Configuration | 7-3 |
| Retrieving Published RTLog Files | 7-4 |
| Security Configuration | 7-5 |
| Acquiring IDCS Token | 7-5 |
| Provide IDCS Authentication | 7-6 |
| | |
| A Appendix: Xstore to Sales Audit Mapping Details | |
| Transaction Types | A-1 |
| Tender Types | A-4 |
| Tender Totals | A-6 |
| Item Types | A-7 |
| Sales Audit Reason Codes | A-8 |
| Reason Codes | A-8 |

| | |
|---|-------------|
| Return Reason Codes..... | A-9 |
| Discount Reason Codes..... | A-9 |
| Item Price Override Reason Codes..... | A-10 |
| Item Status/and Sales Types..... | A-10 |
| Customer ID Types | A-12 |
| Sales Audit Tax Codes | A-12 |
| Reference Codes/Labels | A-12 |
| Sale Return Transaction | A-12 |
| Transaction Header Mapping | A-12 |
| Line Item (TITEM) Mapping | A-13 |
| Item Discount (IDISC) Mapping and Round off Discount Mapping..... | A-13 |
| Item Level Tax Mapping | A-14 |
| Tender Line Mapping..... | A-15 |
| Rounded Off Amount..... | A-15 |

B Appendix: Flat File Mapping

| | |
|--|--------------|
| RMS Diff Group Detail..... | B-1 |
| RMS Diff Group Header..... | B-3 |
| RMS Diffs | B-5 |
| RMS Item Header..... | B-7 |
| RMS Item Location | B-42 |
| RMS Merchandise Hierarchy..... | B-60 |
| RMS Organizational Hierarchy..... | B-65 |
| RMS Related Item Detail..... | B-71 |
| RMS Related Item Header..... | B-74 |
| RMS Store | B-78 |
| RMS Store Address | B-87 |
| RMS VAT..... | B-89 |
| RMS VAT Item | B-96 |
| RPM Clearance Price | B-99 |
| RPM Regular Price | B-101 |
| RPM Promotions | B-104 |

List of Tables

| | | |
|------|---|-------|
| 1-1 | Inbound Xstore Suite V21 Merchandising Integration Matrix..... | 1-2 |
| 1-2 | Outbound Xstore Suite V21 Merchandising Integration Matrix..... | 1-2 |
| 2-1 | MNT Files..... | 2-6 |
| 2-2 | File Loading Dependency..... | 2-9 |
| 2-3 | Flat File Properties..... | 2-11 |
| 4-1 | RMFCS Connection Configuration..... | 4-1 |
| 4-2 | OCDS Connection Configuration..... | 4-2 |
| 4-3 | OCDS Configuration Options..... | 4-2 |
| 4-4 | Integration Database Tables..... | 4-4 |
| 7-1 | REST Services related to the RTLogMappingConfig.xml..... | 7-2 |
| 7-2 | REST Services related to the RTLogFormatConfig.xml..... | 7-3 |
| 7-3 | REST Services to download RTLog Files..... | 7-5 |
| A-1 | Transaction Type Mapping..... | A-2 |
| A-2 | Tender Type Mapping..... | A-4 |
| A-3 | Total Tender ID Mapping..... | A-7 |
| A-4 | Item Type Mapping..... | A-7 |
| A-5 | Sales Audit Reason Codes..... | A-8 |
| A-6 | Return Reason Codes..... | A-9 |
| A-7 | Discount Reason Codes..... | A-9 |
| A-8 | Item Price Override Reason Codes..... | A-10 |
| A-9 | Item Status/ and Sales Type Mapping..... | A-11 |
| B-1 | RMS Diff Group Detail Mapping..... | B-1 |
| B-2 | RMS Diff Group Header Mapping..... | B-3 |
| B-3 | RMS Diffs Mapping..... | B-5 |
| B-4 | RMS Item Header Mapping..... | B-7 |
| B-5 | RMS Item Location Mapping..... | B-43 |
| B-6 | RMS Merchandise Hierarchy Mapping..... | B-60 |
| B-7 | RMS Organizational Hierarchy Mapping..... | B-66 |
| B-8 | RMS Related Item Detail Mapping..... | B-71 |
| B-9 | RMS Related Item Header Mapping..... | B-74 |
| B-10 | RMS Store Mapping..... | B-78 |
| B-11 | RMS Store Address Mapping..... | B-87 |
| B-12 | RMS VAT Mapping..... | B-90 |
| B-13 | RMS VAT Item Mapping..... | B-96 |
| B-14 | RPM Clearance Price FDETL Record Mapping..... | B-99 |
| B-15 | RPM Clearance Price FDELE Record Mapping..... | B-101 |
| B-16 | RPM Regular Price FDETL Record Mapping..... | B-101 |
| B-17 | RPM Regular Price FDELE Record Mapping..... | B-103 |
| B-18 | RPM Promotions Record Mappings..... | B-104 |

Send Us Your Comments

Oracle Retail Xstore Suite 21.0/Oracle Retail Merchandising Suite 22.1.401.0
Implementation Guide

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

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

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

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

Note: Before sending us your comments, you might like to check that you have the latest version of the document and if any concerns are already addressed. To do this, access the new Applications Release Online Documentation CD available on My Oracle Support and www.oracle.com. It contains the most current Documentation Library plus all documents revised or released recently.

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

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

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

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

Preface

This implementation guide describes the implementation steps that you should take when integrating the Xstore Suite with the Merchandising applications.

Audience

This Implementation Guide is intended for the integrators and implementation staff, as well as the retailer's IT personnel.

Documentation Accessibility

For information about Oracle's commitment to accessibility, visit the Oracle Accessibility Program website at

<http://www.oracle.com/pls/topic/lookup?ctx=acc&id=docacc>.

Access to Oracle Support

Oracle customers that have purchased support have access to electronic support through My Oracle Support. For information, visit

<http://www.oracle.com/pls/topic/lookup?ctx=acc&id=info> or visit <http://www.oracle.com/pls/topic/lookup?ctx=acc&id=trs> if you are hearing impaired.

Related Documents

For more information, see the following Release documents:

- Oracle Retail Merchandising System documentation set
- Oracle Retail Price Management documentation set
- *Oracle Retail Xstore Suite Implementation and Security Guide*

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, 21.0) or a later patch release (for example, 21.0.1). If you are installing the base release or additional patches, read the documentation for all releases that have occurred since the base release before you begin installation. Documentation for patch releases can contain critical information related to the base release, as well as information about code changes since the base release.

Improved Process for Oracle Retail Documentation Corrections

To more quickly address critical corrections to Oracle Retail documentation content, Oracle Retail documentation may be republished whenever a critical correction is needed. For critical corrections, the republication of an Oracle Retail document may at times not be attached to a numbered software release; instead, the Oracle Retail document will simply be replaced on the Oracle Help Center (docs.oracle.com) Web site, or, in the case of Data Models, to the applicable My Oracle Support Documentation container where they reside.

This process will prevent delays in making critical corrections available to customers. For the customer, it means that before you begin installation, you must verify that you have the most recent version of the Oracle Retail documentation set. Oracle Retail documentation is available on the Oracle Help Center (docs.oracle.com) at the following URL:

<https://docs.oracle.com/en/industries/retail/index.html>

An updated version of the applicable Oracle Retail document is indicated by Oracle part number, as well as print date (month and year). An updated version uses the same part number, with a higher-numbered suffix. For example, part number E123456-02 is an updated version of a document with part number E123456-01.

If a more recent version of a document is available, that version supersedes all previous versions.

Oracle Retail Documentation on the Oracle Help Center (docs.oracle.com)

Oracle Retail product documentation is available on the following web site:

<https://docs.oracle.com/en/industries/retail/index.html>

(Data Model documents can be obtained through My Oracle Support.)

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. |

Overview

This document describes integration between the Xstore suite of solutions and the Merchandising suite of solutions. For Merchandising, this document covers functionality supported in both the on premise and cloud service versions of the following products:

| On Premise Solution | Cloud Service Solution |
|--|--|
| Oracle Retail Merchandising System (RMS) | Oracle Retail Merchandising Foundation Cloud Service (RMFCS) |
| Oracle Retail Sales Audit (ReSA) | Oracle Retail Merchandising Foundation Cloud Service (RMFCS) |
| Oracle Retail Pricing (RP) | Oracle Retail Pricing Cloud Service (RPCS) |

For the Xstore Suite, this document covers functionality supported in the following products:

| On Premise Solution | Cloud Service Solution |
|---------------------------------------|--|
| Oracle Retail Xstore Office | Oracle Retail Xstore Office Cloud Service (XOCS) |
| Oracle Retail Xstore Point of Service | NA |

The integration of the Merchandising Suite and the Xstore Suite consists of two major data flows:

- **Inbound:** Merchandising and pricing information is consumed by Xstore Office where data is loaded into the Xcenter database and files are automatically distributed for data loading into Xstore databases.
- **Point of Service transactions** from Oracle Retail Xstore Point of Service flow to Xstore Office for transformation and distribution to the Merchandising Foundation Cloud Service Sales Audit module.

In combination, these data flows represent the round trip of data between the stores and headquarters. New items, other foundation data, and prices from headquarters are communicated to Xstore. Sales and returns from Xstore are communicated to Merchandising, where these transactions impact inventory. Merchandising further integrates summarized sales and inventory information from Xstore to other Oracle Retail applications, such as Planning and Analytics.

Inbound: Integration Options with Xstore Office

The inbound Xstore Suite integration from Merchandising is achieved using platform specific integration interfaces. [Table 1–1](#) lists important details about the integration options available to the two Xstore Office V21 deployment topologies: Cloud vs On-Premise. The inbound Xstore Office integration interfaces are:

- RMFCS/RPCS web services – Xstore Office requests merchandising and pricing data directly from Merchandising and Pricing Cloud Services
- OCDS web services - Xstore Office requests merchandising and pricing data from OCDS, which is a data hub.
- RMFCS/RPCS flat file export – Merchandising and pricing data exported from RMFCS and RPCS can be imported into Xstore Office using momzip archive files.

Table 1–1 Inbound Xstore Suite V21 Merchandising Integration Matrix

| Xstore Office | Merchandising and Pricing System Data | | |
|---------------|---------------------------------------|-------------------|------------------------|
| | Direct Web Services | OCDS ¹ | Flat File ² |
| Cloud | Yes | Yes | Yes |
| On-Premise | No | Yes | Yes |

¹ OCDS is only available with MFCS V19

² RPCS flat file export does not support export of promotions

Outbound: Integration Options with Xstore Office

The outbound Xstore Suite integration to the Merchandising Foundation Cloud Service Sales Audit module is achieved with two options.

- Using RTLog flat files generated by RTLog generator. Depending on Xstore suite and Sales Audit deployment topologies, different file delivery options are supported including SFTP, FTS, REST download, or publishing to file system folder.
- Using direct Sales Audit REST service integration. RTLog generator is not involved.

Table below lists important details about the integration options available to the two Xstore Office V21 deployment topologies: Cloud vs On-Premise.

Table 1–2 Outbound Xstore Suite V21 Merchandising Integration Matrix

| Xstore Office | Merchandising Foundation Cloud Service Sales Audit | | | File Download (REST) | Publish to File System Folder |
|---------------|--|------------------|------------------|----------------------|-------------------------------|
| | Sales Audit (REST) | SFTP | FTS | | |
| Cloud | Yes ¹ | Yes ² | Yes ³ | Yes | No |
| On-Premise | No | No | No | No | Yes |

¹ XOCS integration with Sales Audit REST API only available in RMFCS V21

² XOCS integration with SFTP is only available in RMFCS V19

³ XOCS integration with FTS only available in RMFCS V21

The details of the integration are covered in the remaining sections of this guide:

- [Chapter 2, "Data Flow from Merchandising to Xstore"](#): This chapter describes the flow of data from the Merchandising applications to the Xstore Suite.
- [Chapter 3, "Transaction Flow from Xstore to Sales Audit"](#): This chapter describes the flow of transactions from Xstore Point of Service to Sales Audit.
- [Chapter 4, "Webservice Integration Configuration"](#): This chapter provides information on the configuration changes that can be made for the integration.
- [Chapter 5, "Integration Considerations"](#): This chapter covers functional and technical points about the integration that need to be taken into consideration when implementing the integration.
- [Chapter 6, "RTLog Generator On-Premise"](#): This chapter covers how to install, deploy, and configure the RTLog Generator application.
- [Chapter 7, "RTLog Generator Cloud"](#): This chapter covers the RTLog Generator Cloud.
- [Appendix A, Appendix: Xstore to Sales Audit Mapping Details](#): This appendix provides tables that describe the mappings.
- [Appendix B, Appendix: Flat File Mapping](#): This appendix describe the flat file mapping.

Data Flow from Merchandising to Xstore

This chapter covers the data flow from Merchandising and Pricing to Xstore, where data can be consumed by Xstore Office for loading into the Xcenter database, and files distributed to Xstore for loading into the Xstore database.

The Xstore Suite can consume Merchandising and Pricing data in the following categories:

- Merchandise hierarchy
- Organizational hierarchy
- Store (including addresses)
- Dimension type (derived from item differentiator, or diff usage)
- Dimension value (derived from item diff usage)
- Items and Item Location
- Item UPC
- Value Added Tax (VAT) rules and item associations
- Related items
- Initial prices
- Price changes
- Promotions
- Clearance prices

Conceptual Data Flow

This section describes the conceptual data flow for each of the inbound integration interfaces supported by the Xstore Suite.

Note: Please refer to [Table 1-1](#) for data flow option availability with respect to hosting and versions.

Using RMFCS/RPCS Direct Services

The Merchandising Foundation Cloud Service and Pricing Cloud Service expose a set of REST web services that allow Xstore Office to communicate directly with MFCS and RPCS.

Xstore Office leverages these interfaces to poll, at a regularly scheduled intervals, for updates (for example, additions, deletions, and modifications) of Merchandising and Pricing data used by the Xstore Suite. Xstore Office communicates requests for changes in data by directly calling the Merchandising and Pricing REST web services.

When changes to Merchandising and Pricing data are detected, Xstore Office generates .mnt files containing the commands to update Xstore Suite databases. When .mnt files are generated, they are automatically deposited into the Xstore Office auto-drop folder for data-loading and distribution.

- If any detected updates necessitate updating the Xcenter database then the appropriate .mnt file will be automatically dataloaded.
- If any detected updates necessitate updating the Xstore database then the appropriate .mnt files will be deployed to the store where they can be dataloaded either immediately or at store close.

Using OCDS

Oracle Omnichannel Cloud Data Service (OCDS) is a data hub, enabling the merchandising applications to share information with the Oracle Retail Omnichannel applications.

OCDS is composed of three major components:

- BDI Batch Job Admin - Enables the flow of data into OCDS using the Oracle Bulk Data Integration (BDI) technology. Job Admin has a User Interface (UI) to support the management of BDI batch jobs.
- RIB Injector - Enables the flow of data into OCDS from the Oracle Retail Integration Bus (RIB).
- ORDS Web Services - Enables the data contained in OCDS to be accessed by Omnichannel applications, such as the Xstore Suite, through the use of RESTful web services.

Figure 2–1 illustrates the major system components that make up OCDS, and the interactions of the applications major actors.

Figure 2–1 OCDS Component Diagram

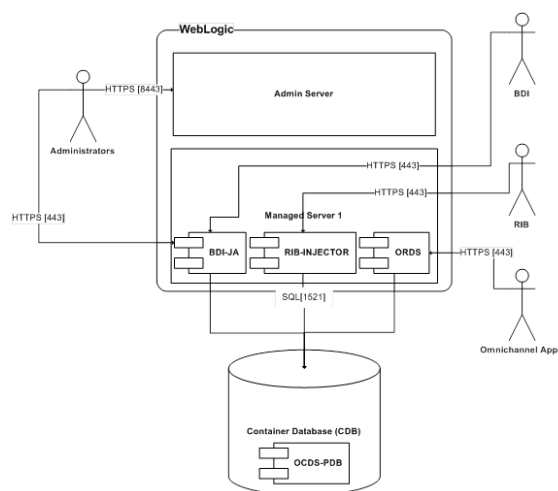
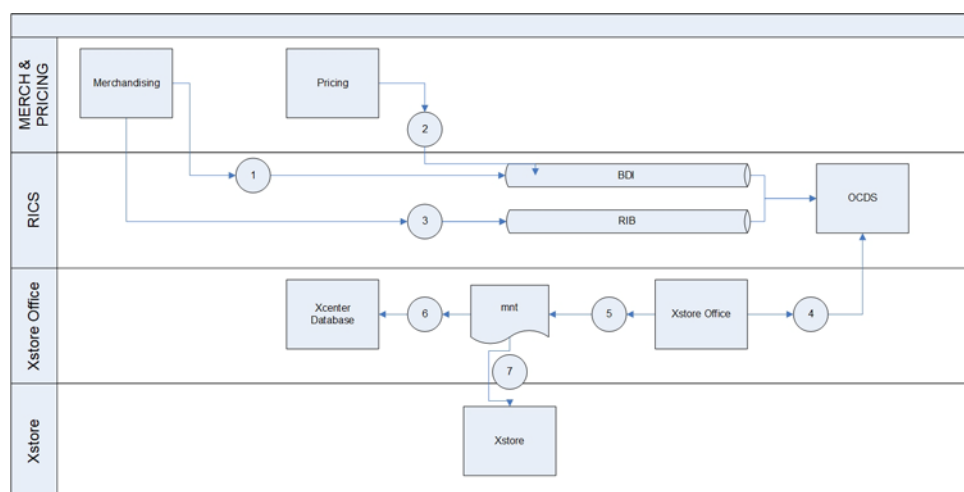


Figure 2–2 Conceptual Data Flow from Merchandising and Pricing to Xstore POS

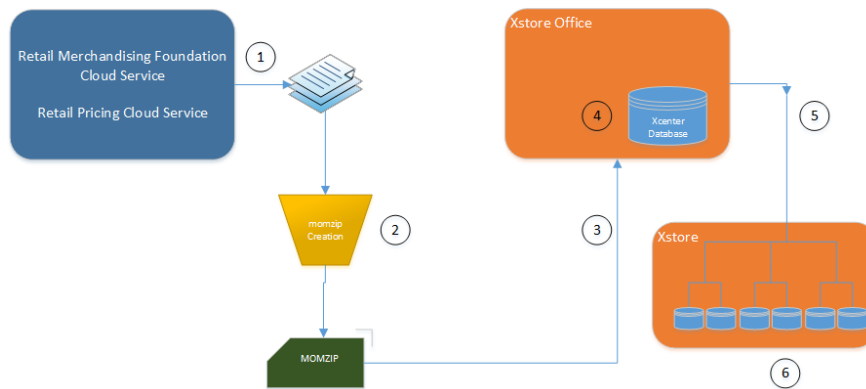
The following steps describe the flow in [Figure 2–2](#).

1. OCDS receives an initial load of Merchandising foundation data using BDI as the data transport. This is generally a one-time push of data over BDI into OCDS.
2. OCDS starts to receive, on-going, regularly scheduled pricing and promotion data using BDI as the data transport.
3. OCDS starts to receive, on-going, near-real-time updates of Merchandising data using the RIB as the data transport.
4. Xstore Office starts polling OCDS, at a regularly scheduled interval, to check for updates (for example, additions, deletions, and modifications) of Merchandising and Pricing data used by the Xstore Suite. Xstore Office communicates requests for changes to OCDS data by calling the OCDS REST web services.
5. When changes to OCDS data are detected, Xstore Office generates .mnt files containing the commands to update Xstore Suite databases. When .mnt files are generated, they are automatically deposited into the Xstore Office auto-drop folder for data-loading and distribution.
6. If any detected OCDS updates necessitate updating the Xcenter database then the appropriate .mnt file will be automatically dataloaded.
7. If any detected OCDS updates necessitate updating the Xstore database then the appropriate .mnt files will be deployed to the store where they can be dataloaded either immediately or at store close.

Using Flat Files

Note: The flat file integration will be deprecated. For more information about this feature, see the *Oracle Retail Merchandising Cloud Services Deprecation Advisory* (Doc ID 2760721.1) on My Oracle Support. For new integrations, use the direct services.

The Merchandising Foundation Cloud Service and Pricing Cloud Service support the export of data into files that can be packaged for consumption by the Xstore Suite. [Figure 2–3](#) illustrates the data flow from the Merchandising applications to Xstore Office and Xstore.

Figure 2–3 Conceptual Data Flow from Merchandising to Xstore Suite

The following steps describe the flow in [Figure 2–3](#):

1. RMFCS/RPCS writes data to .dat files using a series of .ksh extract scripts. These scripts support both kill/fill (full) and delta processing. Many of these scripts also support creating files that apply either to all stores or store-specific files.

Use the kill/fill export option in the merchandising and pricing system's extract scripts to generate full-load files along with running the staging batch for every store and extracting the data using the standard batch processes to convert items, price, and other key data elements from the Merchandising solutions into Xstore.

The following merchandising batch jobs are used for the integration:

- nexport_merchhier.ksh
- nexport_orghier.ksh
- nexport_stores.ksh
- nexport_diffs.ksh
- nexport_diffgrp.ksh
- nexport_itemmaster.ksh
- nexport_itemloc.ksh
- nexport_itemvat.ksh
- nexport_relitem.ksh
- nexport_vat.ksh
- nexport_stg_purge.ksh

For more information, see the *Oracle Retail Merchandising System Operations Guide, Volume 1 - Batch Overviews and Designs*.

The following pricing batch jobs are used for the integration:

- RegularPriceChangePublishBatch
- ClearancePricePublishBatch

For more information, see the *Oracle Retail Price Management Operations Guide*.

2. The retailer's system integrator is responsible for packaging the extract .dat files in a zip-format archive file named using the file extension '.momzip'. The integration uses logic based on filenames so it is important to maintain the use of the file's name as generated.

Note: The momzip file must be scanned for viruses before delivery to Xstore Office for processing

3. Momzip files are delivered to Xadmin's auto-deploy file folder either by use of webservice API or direct file system access, as available per hosting type.
4. Xadmin periodically checks the auto-deploy directory for new Momzip files. The system automatically processes the dat-file contents of the momzip. For details on the set of Merchandising files targeted to corporate or stores, see "Merchandising File Consumption by Location".
5. If any RMFCS/RPCS extract files necessitate updating the Xcenter database then the appropriate .dat files will be automatically dataloaded.

If any RMFCS/RPCS extract files necessitate updating the Xstore database then the appropriate .dat files will be deployed to the store where they can be dataloaded either immediately or at store close.

Xstore Office Deployments

Each of the types of integration discussed in this guide involve the delivery of merchandising and pricing data, in file format, to Xstore Office's auto-deploy folder. Xadmin polls the auto-deploy folder at a configured interval and automatically processes new files. The polling-interval is configurable, and is 15 minutes by default. For information on how to configure these settings, see the Oracle Retail Xstore Office User Guide.

Deployments of .mnt or .dat files, to be loaded at the store, are automatically created for either immediate or scheduled distribution. Each deployment status of the files is displayed in the Xstore Office Deployments screen (Figure 2-1).

Once a store is closed or when the retail period changes in a 24x7 configuration, Xenvironment of the lead register pulls down files scheduled for end-of-day and runs the Xstore DataLoader to import all the files deployed into the store primary database. Files scheduled for immediate dataloading are pulled down and dataloaded automatically.

Figure 2-4 Xstore Office Deployments

| ID | Name | Type | Plan | Launch Date | Waves Complete | Status |
|-------|-----------------------------|-------------|-------------|--------------|----------------|-----------|
| 10239 | 2_dimensionValue_1542664871 | Auto Deploy | Single Wave | Nov 19, 2018 | 0 out of 1 | Scheduled |
| 10238 | 2_dimensionType_1542664869 | Auto Deploy | Single Wave | Nov 19, 2018 | 0 out of 1 | Scheduled |
| 10237 | 5_price_1542664619106_STCF | Auto Deploy | Single Wave | Nov 19, 2018 | 0 out of 1 | Scheduled |
| 10236 | 5_price_1542664619788_STCF | Auto Deploy | Single Wave | Nov 19, 2018 | 1 out of 1 | Complete |
| 10235 | 5_price_1542664619289_STCF | Auto Deploy | Single Wave | Nov 19, 2018 | 0 out of 1 | Scheduled |
| 10234 | 5_price_1542664613656_STCF | Auto Deploy | Single Wave | Nov 19, 2018 | 0 out of 1 | Scheduled |
| 10233 | 4_itemLoc_1542664517424_ST | Auto Deploy | Single Wave | Nov 19, 2018 | 0 out of 1 | Scheduled |
| 10232 | 4_itemLoc_1542664516629_ST | Auto Deploy | Single Wave | Nov 19, 2018 | 0 out of 1 | Scheduled |
| 10231 | 4_itemLoc_1542664515492_ST | Auto Deploy | Single Wave | Nov 19, 2018 | 0 out of 1 | Scheduled |
| 10230 | 4_itemLoc_1542664513014_ST | Auto Deploy | Single Wave | Nov 19, 2018 | 0 out of 1 | Scheduled |
| 10229 | 3_item_1542664517312_STOR | Auto Deploy | Single Wave | Nov 19, 2018 | 0 out of 1 | Scheduled |
| 10228 | 3_item_1542664516522_STOR | Auto Deploy | Single Wave | Nov 19, 2018 | 0 out of 1 | Scheduled |
| 10227 | 3_item_1542664516265_STOR | Auto Deploy | Single Wave | Nov 19, 2018 | 0 out of 1 | Scheduled |
| 10226 | 3_item_1542664512880_STOR | Auto Deploy | Single Wave | Nov 19, 2018 | 0 out of 1 | Scheduled |
| 10225 | 5_price_1542663079013_STCF | Auto Deploy | Single Wave | Nov 19, 2018 | 0 out of 1 | Scheduled |
| 10224 | 5_price_1542663076679_STCF | Auto Deploy | Single Wave | Nov 19, 2018 | 1 out of 1 | Complete |

Using RMFCS/RPCS direct services or OCDS

The .mnt files created by Xstore Office leverage an xml header to target when and where files are dataloaded.

For example, this header would appear in a store-specific file that is immediately downloaded and dataloaded.

```
<Header line_count="104" target_org_node="STORE:121"
destination="XSTORE_ONLY" apply_immediately="true" download_
time="IMMEDIATE" />
```

Table 2–1 MNT Files

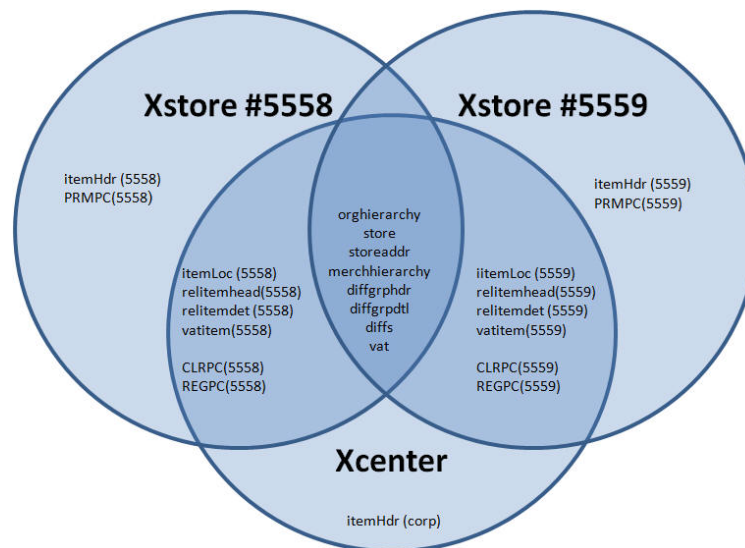
| Filename Prefix | Store Specific | Destination | Downlaod Time |
|------------------|----------------|--------------|---------------|
| 1_orgHier | No | ALL | STORE_CLOSE |
| 2_dimensionType | No | ALL | STORE_CLOSE |
| 2_dimensionValue | No | ALL | STORE_CLOSE |
| 2_merchHier | No | ALL | STORE_CLOSE |
| 2_retailLoc | No | ALL | STORE_CLOSE |
| 2_vat | No | ALL | STORE_CLOSE |
| 3_item | Yes | XSTORE_ONLY | STORE_CLOSE |
| 3_itemCorp | No | XCENTER_ONLY | |
| 3_itemUPC | Yes | XSTORE_ONLY | STORE_CLOSE |
| 4_itemLoc | Yes | ALL | STORE_CLOSE |
| 4_itemLocCorp | Yes | XCENTER_ONLY | |
| 4_relatedItem | Yes | ALL | STORE_CLOSE |
| 5_clrpc | Yes | ALL | IMMEDIATE |
| 5_price | Yes | ALL | IMMEDIATE |
| 5_promo | Yes | XSTORE_ONLY | STORE_CLOSE |
| 5_regp | Yes | ALL | IMMEDIATE |

Using Flat Files

The files produced by the extract programs containing data loaded into the Xcenter and Xstore databases comprise four data sets. A data file's targeted location is specified in its file name:

- Data loaded into Xcenter
- Data loaded into Xcenter and all stores
- Data loaded into one store
- Data Loaded into all stores

Figure 2–5 illustrates where files are dataloaded by type, using a two store chain example.

Figure 2–5 File Distribution/Dataloading by Type

Xstore Dataloader

DataLoader is the Xstore component responsible for translating delimited files into instructions to modify the contents of an Xstore Suite database. It can consume .mnt files generated by an Xstore Office integrated with OCDS and MFCS, or the .dat files extracted from Oracle Retail Merchandising and Pricing Cloud Systems.

The DataLoader interacts with Xstore Office, Xcenter, Xenvironment, and Xstore Point of Service to provide a complete automated solution for the propagation of foundation data changes to the centralized and store-level databases used in an enterprise Xstore deployment. Xstore data not supplied by an Oracle Merchandising/Pricing system can also be loaded by the DataLoader using its native .mnt format.

The DataLoader is designed to adapt flat files of data into relational data that Xstore can use. These flat files are referred to generically as data files within the DataLoader. Each field in a data file is delimited by a vertical bar (|) character. The DataLoader is configured to detect file types so it can process a data file's lines in distinct units of work appropriate for the type of file.

If a failure occurs during DataLoader processing of a data file, all SQL statements associated with the unit of work are rolled back and the error is logged. Processing continues with the next unit of work in the data file.

For more detailed information, see the following documents:

- Retail Reference Architecture
- *Oracle Retail Xstore Point of Service Host Interface Guide*

Both documents are available on My Oracle Support.

Using RMFCS/RPCS direct services or OCDS

In these integration styles, Xstore Office communicates with a webservice endpoint to obtain merchandising and pricing data that is written into files using the dataloader's native .mnt file format.

Using Flat Files

DataLoader is configured with a list of detectors to identify known file types that can be processed. Unknown file types are skipped and not processed. A Merchandising file detector is configured to identify all types of Merchandising data files and their meta data.

The detection is based solely on file names. Regular expressions are configured to perform pattern matching in file name to identify Merchandising file types and its meta-data including target store ID, fill type, timestamp, and line count:

- Merchandising file type detection
 - A file name is matched against regular expressions configured to detect its Merchandising type. If no match is found, the file is not a Merchandising file type. The keys in the bean configuration are the Merchandising file types that Xstore/Xcenter care about.
- Target store detection
 - Target store ID is used by DataLoader as well as Xstore Office to determine the deployment target of a Merchandising file.
 - A file name is matched against regular expressions configured to detect its target store ID. Not all Merchandising file types have a target store ID configured:
 - If a store ID is not detected, the file is deployed to all stores and imported into Xcenter.
 - If a store ID is detected and is corp, the file is imported into Xcenter only.
 - If a store ID is detected and is not corp, it is deployed to the store. With the exception of Item Header batch, it is also imported into Xcenter.
- Timestamp detection
 - Timestamp is used by DataLoader to sort the files. For more details, see "File Sorting". A file name is matched against regular expressions configured to detect its timestamp.
- Line count detection
 - Line count is used by DataLoader to validate a file. If the number of lines in the file does not match the line count, a warning is logged.

File Sorting

There are some data dependencies when importing .dat files into Xstore, such as related item detail that needs to be imported after the related item header. When DataLoader is called to import multiple Merchandising files in the same deployment, it applies sorting to the files before importing them.

A detector is configured to have a sorting strategy, which is used to sort all the files the detector detects. A Merchandising file sorting strategy bean is configured for the Merchandising file detector to perform sorting for all Merchandising files based on their file types. Files of the same Merchandising file type are sorted based on their timestamps. Out of the box the following sorting order is specified:

- Org Hierarchy
- Store
- Store Address
- Merchandise Hierarchy

- VAT
- Diff Group Head
- Diff Group Detail
- Diff
- Item Head
- Item Loc
- VAT Item
- Related Item Head
- Related Item Detail
- Regular Price Change
- Clearance Price Change
- Promotion Price Change

File Loading Dependency Although the sorting strategy configuration lists all Merchandising file types, not all file types have file loading dependencies. The actual dependencies are shown in [Table 2–2](#).

Table 2–2 File Loading Dependency

| File Type | Depends on |
|---------------------|---|
| VAT Item | Item Loc |
| Store Address | Store |
| Related Item Detail | Related Item Head, Item Loc |
| Item Loc | Item Head |
| Item Head | Diff, Diff Group Detail, Diff Group Head, Merchandise Hierarchy |
| Diff Group Detail | Diff Group Head |
| Diff | Diff Group Detail, Diff Group Head |

File Iteration and Transformation DataLoader processes each file in the sorted order. It invokes a file iterator to process each file. A file iterator implements Java Iterator interface. During each iteration, it transforms flat file records into a list of IPersistable (DAO or SQLQuery) objects, and returns them.

A Merchandising file iterator is configured for each Merchandising file type. It processes lines between unit dividers as a data unit that should be transformed together:

- A single line iterator that expects each line in the file, other than FHEAD or FTAIL, is a data unit that gets transformed during each iteration. One and only one line in a unit is expected. An exception is raised if that is not the case.
- A multi-line iterator expects multiple lines to form a data unit that gets transformed together during each iteration. A unit may contain one or more lines. Out of the box, only promotion price change is configured to have a multi-line iterator.
- Unit dividers are lines that end a unit. They are configured as unit definitions for each Merchandising file type.

A Merchandising transformer is called to convert a unit of data from a flat file to a list of IPersistable (DAO or SQLQuery) objects. A transformer is configured for each Merchandising file type.

All Merchandising transformers implement the IMOMDataTransformer interface, which defines two APIs:

- The transform API is invoked by the iterator in each iteration. It does all the transformation to turn a unit of flat file data to a list of IPersistable objects to create, update, or delete foundation data records in database.
- The purgeData API is invoked once for a file by the iterator. It is only called if the file is for a full reload. It returns a list of IPersistable objects to remove all existing records sourced from Merchandising.

Persisting into the Database DataLoader saves IPersistable objects to database in batches. A batch contains a list of AtomicPersistables objects. The maximum number of AtomicPersistables objects in a batch is configurable. An AtomicPersistables is a container of a group of IPersistable objects that must all be persisted or rolled back together as a unit. All IPersistable objects returned in one iteration are grouped into one AtomicPersistable object.

DataLoader first attempts to persist and commit all IPersistable objects from all AtomicPersistables objects in a batch together. If it fails, it tries to persist and commit IPersistable objects from one AutomiPersistables at a time. The number of succeeded and failed records are written to summary.ini files.

Phased Rollout of Stores

Retailers who are migrating an existing Xstore Suite implementation from a different source for merchandising foundation and pricing data may desire to exclude stores defined in their organization hierarchy during a phased rollout of the integration.

Using RMFCS/RPCS direct services or OCDS (Cloud Only)

Xstore Office Cloud Service's Integration Manager supports the use of Store Collections when configuring either integration method. By associating a Store Collection with an integration method a retailer can identify which stores require data from the integration system. This feature can facilitate a phased-store rollout of data to a subset of existing stores in Merchandising. An integration using a Store Collection will limit the creation and deployment of mnt files to only those stores defined by the Store Collection.

Adding another store to an existing integration will enable incremental changes to start flowing to the store. A manual refresh from Merchandising and Pricing through either integration method should be requested for the added store to seed it with a full set of foundation and pricing data.

Using Flat Files

A momzip.properties file can be included in the root folder of the momzip file to prevent a .dat files from being dataloaded in Xcenter and to limit the creation of store deployments to only a subset of stores. By associating a Store Collection with the momzip's contents a retailer can identify which stores are eligible for targeting when creating deployments. This feature can facilitate a phased-store rollout of data to a subset of existing stores.

For example, the properties below are used to dataload Xcenter, but only result in deployment to stores that are members of the store collection XOstores.

dataload.xcenterEligibleFiles=true

deploy.storeCollection=XOstores

Table 2–3 Flat File Properties

| Property | Description | Values | |
|-------------------------------|---|--|---------|
| | | Possible | Default |
| dataload.xcenterEligibleFiles | Indicates if Xcenter is to dataloaded | true/false | true |
| deploy.storeCollection | If a value is present for this property, then deployments will only be created for stores specified by the store collection. If no value is present then no deployments will be created | The ID of an existing Xadmin store collection, or no value | null |

Note: When the momzip.properties file does not contain a supported property key, or the key exists but has no value, then the key's default value is used.

Manual Refresh of an Xstore Database

The typical integration's operational flow consists of a one-time initial load to seed Xstore Suite databases with data from the Oracle merchandising and pricing systems, followed by incremental updates to update Xstore Suite databases with changes made in the upstream systems. However, exceptions to this flow are expected and the integrations provide mechanisms to refresh databases on an as-need bases.

Using RMFCS/RPCS direct services or OCDS

Both integration methods are designed to be fully automated; under normal conditions no manual steps are required to have Merchandising and Pricing data flow into a store database. However, Xadmin's Data Publisher can be used to regenerate and redeploy .mnt files with Merchandising and Pricing data to a store if exceptional circumstances necessitate the refreshing of an Xstore database.

Use of the Data Publisher to replenish one or more types of Merchandising and Pricing data at a store will result in the purging of all existing sourced data, followed by the loading of a full set of the most recent data.

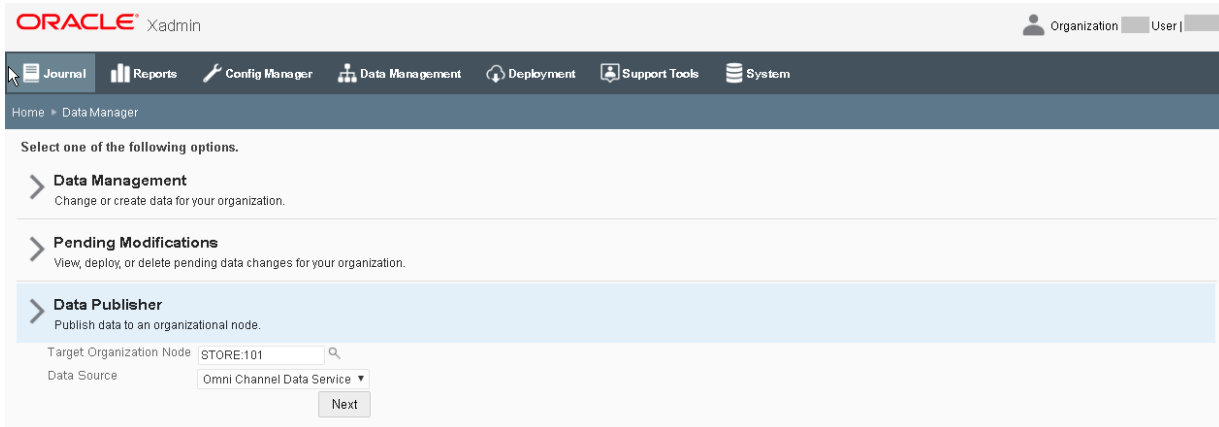
When the integration is enabled in Xstore Office, the DataManager screen's "Data Publisher" option will include a "Data Source" drop down list, which includes the list options: "Omni Channel Data Service" and "Merchandise Foundation Cloud Service".

Note: When you select the "Merchandise Foundation Cloud Service" option, Pricing is included.

To publish data to one or more stores:

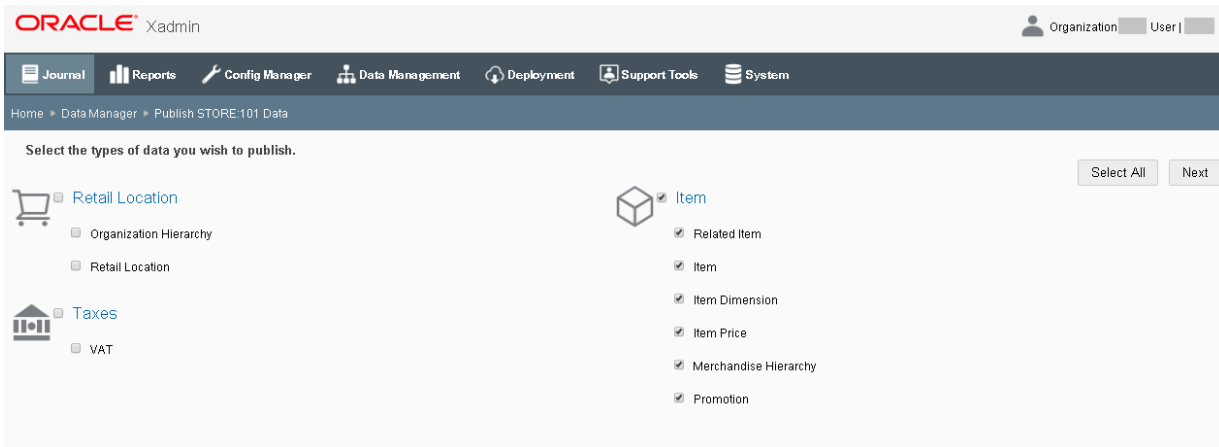
1. Choose the desired Organization Node for the target stores, select the integration, and click **Next**.

Figure 2–6 Xstore Office - Data Publisher Option

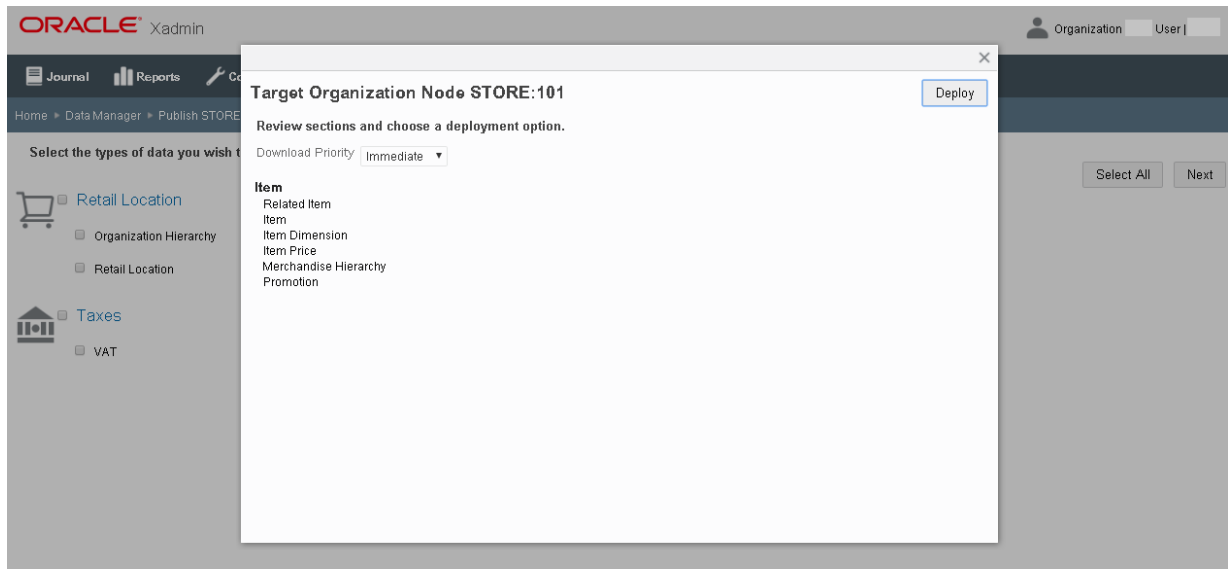


2. Choose the type of data you wish to publish, then click **Next**.

Figure 2–7 Xstore Office - Data to Publish



3. Select a Download Priority, Immediate or Store Close, and click **Deploy**.

Figure 2–8 Xstore Office - Target Organization Node Pop Up

Using Flat Files

Kill/fill (full) files should be generated using the using .ksh extract scripts, packaged in a momzip archive file and targeted to only the stores where data is to be refreshed. Inclusion of a configured momzip.properties file can be used to limit dataloading to desired databases.

Transaction Flow from Xstore to Sales Audit

Xstore is the source of Point of Sale (POS) transactions, including but not limited to the following:

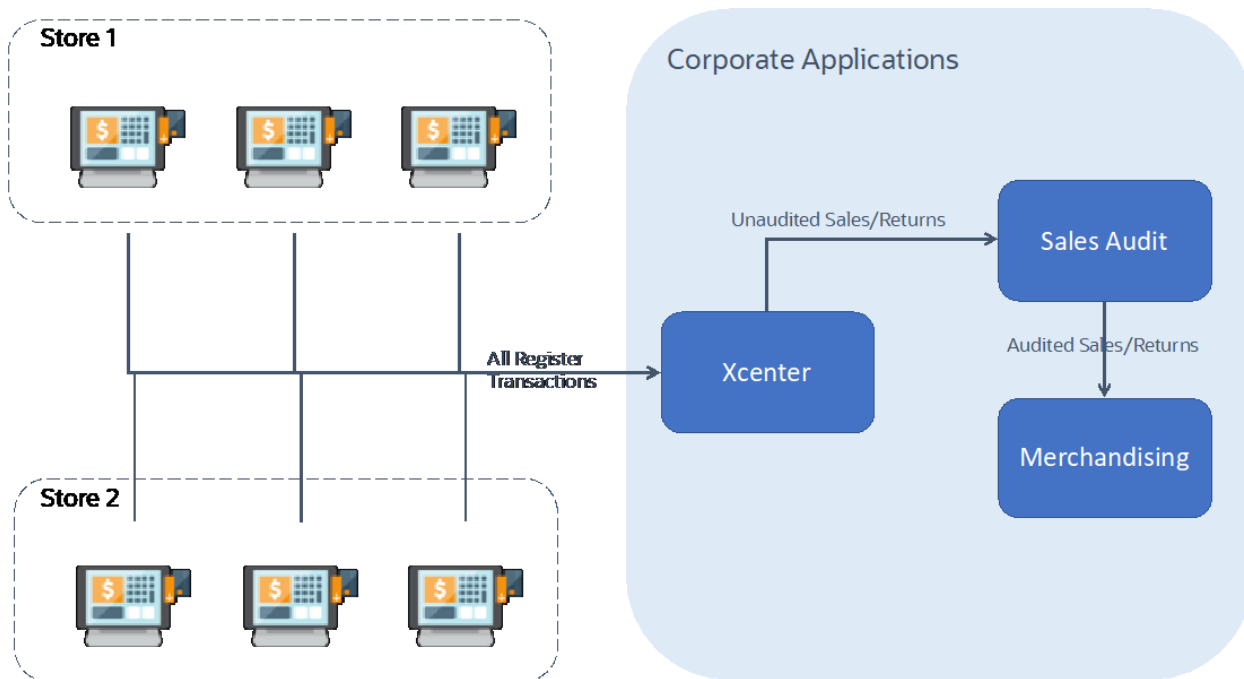
- Sales
- Returns
- Voids
- Cash management transactions
- Many store activity transactions

All transactions produced in Xstore are sent to Sales Audit. Sales Audit processing is primarily concerned with transactions that alter inventory or contain payment. Sales Audit loads other types of Xstore transactions (such as entering training mode, gift registry creation, and so on) into an OTHER transaction type for full visibility and to avoid gaps in the transactions sequence, but will not out of the box perform any audit functions on these OTHER types of transactions.

Sales Audit validates Xstore transactions that impact inventory (such as sales, returns, and customer orders) and exports the information to Merchandising to record the full financial and inventory impact.

Conceptual Data Flow

[Figure 3-1](#) illustrates the transaction flow from Xstore to Sales Audit.

Figure 3–1 Xstore to Sales Audit Transaction Flow

The following steps describe the flow shown in [Figure 3–1](#):

1. All Xstore registers replicate, or persist, all transactions to Xcenter. Note that this includes both customer related transactions (sale, return, void, and so on) and cash management/store operation transactions (paid in, no sale, change to training mode, and so on). Xcenter uses these transactions for activities such as cross location returns.

2. Xcenter broadcasts all transactions to Sales Audit in one of two ways.

In the form of RTLog files generated multiple times per day. This is the legacy approach. For more information, see [Technical Implementation - RTLog Files Integration](#).

Direct REST call to Sales Audit endpoint to post each incoming transaction. This is the new approach introduced since v21. For more information, see [Technical Implementation - Sales Audit REST Integration](#).

The broadcast system in Xcenter provides a means to transmit POSLog data to other systems. The data is transmitted just as Xcenter receives it from the registers through the replication system, which is approximately in real-time. The temporal ordering of the POSLog data is also preserved, just as it is with the replication system.

For more detailed information, see the following documents:

- Retail Reference Architecture available on My Oracle Support
 - *Oracle Retail Xstore Point of Service Technical Guide* available on My Oracle Support
 - *Oracle Retail Xstore Suite Implementation and Security Guide*
3. After successful totaling and auditing, Sales Audit sends all sale/return transactions to Merchandising, where the transactions impact perpetual inventory.

For detailed information about uploadsales_all.ksh, see *Oracle Retail Merchandising Operations Guide, Volume 1 - Batch Overviews and Designs*.

Note: Integrating sales and returns data directly to Merchandising, bypassing Sales Audit, is not a supported integration.

Technical Implementation - Sales Audit REST Integration

The technical implementation of the data from Xcenter/Xstore to Sales Audit consists of four main components:

- [Sales Audit REST Xstore Broadcaster](#)
- [Sales Audit REST Endpoints](#)
- [Scheduled Job to Query Sales Audit Rejected Transactions](#)
- [Xadmin UI to View and Republish Sales Audit Rejected Transactions](#)

Sales Audit REST Xstore Broadcaster

Through Xadmin UI, create a broadcaster of type “Oracle Retail Sales Audit, REST Service” and specify Sales Audit REST endpoint as its endpoint url.

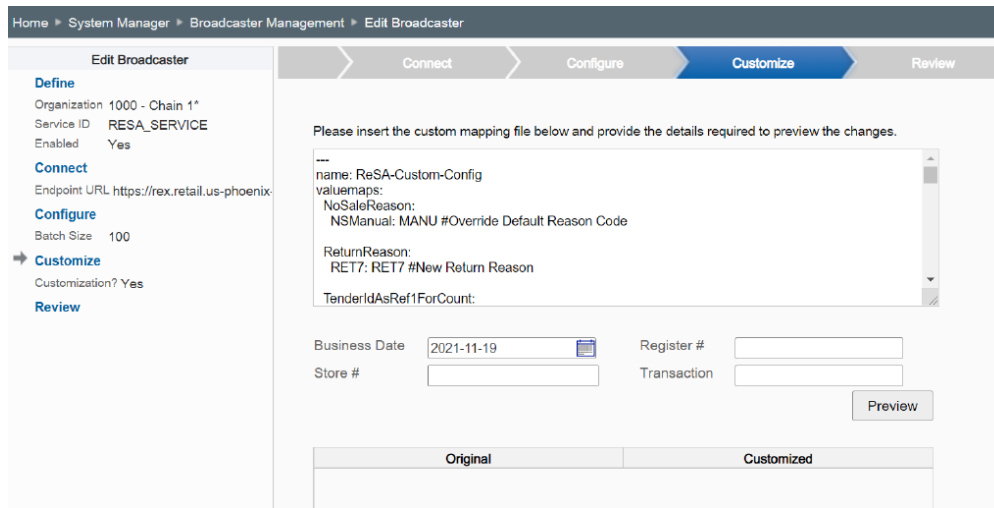
Note: The endpoint url is in the form `https://<host>:<port>/<tenancyId>`. Do not specify endpoint url with full service path like following:
`https://<host>:<port>/<tenancyId>/ResaRestServices/services/private/ReSA/salesService`

A Sales Audit REST broadcaster translates a transaction poslog xml to a JSON request, and posts it as payload to `ResaReSTServices/services/private/Resa/salesService` service. While out-of-box implementation of the translation is provided, the broadcaster gives a retailer the flexibility to customize the JSON request.

The broadcaster configuration wizard has a tab called Customize (see screen capture below). A user can provide custom mapping in yaml format to customize the JSON payload. A Preview feature is also provided to diff a transaction’s JSON request with or without the custom mapping. Once the custom mapping passes the preview, a user can save it with the broadcaster configuration. After an initial delay of about 30 minutes, Xcenter will detect the mapping change and apply it to all transactions broadcasted out to Sales Audit.

Note: For more information on how to write custom mappings in yaml, see the *Oracle Retail Xstore Sales Audit REST Extension Guidelines* (Doc ID Doc ID 1994467.1) on My Oracle Support.

Figure 3–2 Edit Broadcaster Screen - Customize Tab



Sales Audit REST Endpoints

Two Sales Audit REST services are used by Xcenter.

ResaReSTServices/services/private/Resa/salesService: A POST service called by Xcenter broadcaster to post a transaction to Sales Audit. Its payload is the JSON representation of the transaction. A success response is returned if it passes initial validation and is accepted by Sales Audit.

ResaReSTServices/services/private/Resa/getRejectedTransactions: Once a transaction is accepted into Sales Audit, it is placed in staging tables. A batch process will generate the RTLOG file and the file will get consumed by saimptlog/saimptlogi batch. While generating RTLOG, if any validation fails, then that respective transaction will be moved to reject tables. Reject tables would act as the base for this Reject Query Service.

Scheduled Job to Query Sales Audit Rejected Transactions

The Query Sales Audit Rejected Job is scheduled to run hourly in Xcenter if a broadcaster of type “Oracle Retail Sales Audit, REST Service” is enabled for at least one organization. The job iterates through each distinct Sales Audit REST endpoint configured across all organizations to query for rejected transactions through its ResaReSTServices/services/private/Resa/getRejectedTransactions service, and updates the rejected transactions’ WORK_STATUS column to “REJECTED” in Xcenter’s poslog work item queue table TRN_POSLOG_WORK_ITEM.

Xadmin UI to View and Republish Sales Audit Rejected Transactions

To view rejected transactions, a new View Error Log button is introduced in Xadmin SystemTools > Publish PosLog Data. Once clicked, it displays all rejected or otherwise errored out transactions within the range of a user specified criteria for each broadcaster (See below for a screen capture of the UI). It provides error details, and allows a user to republish all or a subset of them once the underlying problem is fixed.

Figure 3-3 POSLog Search - Sales Audit Rejected Transactions

| Trans # | Type | Register # | Store # | Date | Service Id | Last Update | Status | Error | |
|---------|-----------------|------------|---------|--------------|-------------------|--------------|---|----------------------|-------------------------------------|
| 6685 | System Open | 1 | 2324 | Nov 19, 2021 | RESA_REST_SERVICE | Nov 19, 2021 | RESA_ERROR: OR MORE TRANSACTION ARE ERROR OUT | { "status": "ONE ... | <input checked="" type="checkbox"/> |
| 6685 | System Open | 1 | 2324 | Nov 19, 2021 | XBR_1000 | Nov 19, 2021 | XBR_EXCEPTION | com.micros_reta... | <input checked="" type="checkbox"/> |
| 6686 | Session Control | 1 | 2324 | Nov 19, 2021 | RESA_REST_SERVICE | Nov 19, 2021 | RESA_ERROR: OR MORE TRANSACTION ARE ERROR OUT | { "status": "ONE ... | <input checked="" type="checkbox"/> |
| 6686 | Session Control | 1 | 2324 | Nov 19, 2021 | XBR_1000 | Nov 19, 2021 | XBR_EXCEPTION | com.micros_reta... | <input checked="" type="checkbox"/> |
| 6687 | Session Control | 1 | 2324 | Nov 19, 2021 | RESA_REST_SERVICE | Nov 19, 2021 | RESA_ERROR: OR MORE TRANSACTION | { "status": "ONE ... | <input checked="" type="checkbox"/> |

Technical Implementation - RTLog Files Integration

The technical implementation of the data from Xcenter/Xstore to Sales Audit consists of three main components:

- [RTLog File Xstore Broadcaster](#)
- [Xstore RTLog Generator](#)
- [Sales Audit saimptlog/i](#)

RTLog File Xstore Broadcaster

Through Xadmin UI, create a broadcaster of type “Oracle Retail Sales Audit, RTLog File” and specify RTLog generator url as its endpoint url.

Xstore RTLog Generator

RTLog generator is a component that collects and aggregates broadcaster transactions and transforms them to the RTLog file format.

The on premise RTLog generator is packaged with Xstore, but is generally deployed in the same file system as Sales Audit.

For more information, see [Chapter 6](#).

Sales Audit saimptlog/i

Sales Audit is the gateway for POS transactions to integrate to Oracle Retail Merchandising systems. There are two Sales Audit sub-processes that can upload POS files:

- saimptlogi.c is used when loading sales incrementally throughout the day from stores, instead of just once per day. It validates files and directly inserts the transactions into the Sales Audit tables. This includes (as necessary) creating errors for the auditors to research and correct.
- saimptlog.c is used for the once a day import of data from stores. It validates POS files and creates Sql*Loader Files. This includes (as necessary) creating errors for the auditors to research and correct. A subsequent Sql*Load process loads the transactions and errors into the Sales Audit tables.

saimptlog and saimptlogi are built with the same shared code and vary only in their approach to physically loading data into the database. The programs are collectively referred to as saimptlog/i.

There are a number of regular prerequisites in the Sales Audit batch schedule which must be completed before POS transactions can be loaded. For more information about supporting batch jobs, see *Oracle Retail Merchandising Operations Guide, Volume 1 - Batch Overviews and Designs*.

For more detailed information about saimptlog/i and the RTLog file format, see the following documents:

- *Oracle Retail Merchandising Operations Guide, Volume 1 - Batch Overviews and Designs*

Webservice Integration Configuration

This chapter describes configuration options for the service integration, where Xstore Office calls Merchandising and Pricing services directly or through OCDS. This chapter does not apply to the flat file-based integration.

Xstore Office: Direct Service Integration Communication

The direct service integration is set up through the Xadmin Integration Management feature and are stored in the Xstore Office Cloud Service database.

Note: The Integration Management feature is only available for Xstore Office Cloud Service.

Table 4–1 *RMFCS Connection Configuration*

| Fields | Description |
|---------------------------|--|
| Endpoint URL | A URL describing protocol, host name, and port of the Merchandising and Pricing Web Services. Example: <code>https://{hostname}:{port}</code> |
| Merchandising Chain ID(s) | A comma separated list of the chain ids in Merchandising containing stores in the Xstore Organization. |

For Merchandising and POM configurations required for direct service integration from Merchandising and Pricing, see the sections [Merchandising Configuration](#) and [POM Configuration](#) below.

Merchandising Configuration

To enable direct service integration, Oracle Retail Xstore Suite needs to be selected to have service based integration in the Integration section of System Options screen in Merchandising. For more information on System Options, refer to the *Oracle Retail Merchandising Foundation Cloud Service Implementation Guide*.

POM Configuration

These services are operationalized by background processes that can be enabled and configured for running frequency in the Batch Administration screen of POM. For more information on these batch jobs, refer to the *Oracle Retail Merchandising Foundation Cloud Service Operations Guide*.

Note: When these services are initialized, some services need a few hours for initial data processing before they can be used. It is recommended to enable these services a day in advance before Xstore Office starts subscribing.

Xstore Office: OCDS Integration Communication

Xstore Office requests changes to Merchandising and Pricing data in OCDS by calling the REST web services.

[ON PREMISE ONLY] The Xstore Office xcenter.properties file contains properties for the OCDS URL; service path prefix, org/chain mapping and properties.

[CLOUD ONLY] The properties for the OCDS integration are set up through the Xadmin Integration Management feature and are stored in the Xstore Office Cloud Service database.

Note: The Integration Management feature is only available for Xstore Office Cloud Service.

Table 4–2 OCDS Connection Configuration

| Fields | Description |
|---------------------|---|
| Endpoint URL | A URL describing protocol, host name, and port of the OCDS Web Services. Example: <code>https://{hostname}:{port}</code> |
| Service Path Prefix | The root service path for OCDS. Example: <code>\ords\{system}\omnichannel</code> |
| OCDS Chain ID(s) | A comma separated list of the chain ids in the Merchandising system containing stores in the Xstore Organization. |
| Username | The username used to authenticate web service communication with OCDS. |
| Password | The password used to authenticate web service communication with OCDS. |

Xstore Office: Omnichannel Data Service Configuration Options

The OCDS system settings are used to enable and configure the integration.

[ON PREMISE ONLY] The Xstore Office xcenter.properties file contains properties for the OCDS configuration options.

[CLOUD ONLY] Use the Xadmin Integration Management feature to set OCDS configuration options.

Table 4–3 OCDS Configuration Options

| Setting | Description |
|-------------------------------------|---|
| OCDS Retain Job History | The number of days to retain OCDS Job History data in the Xadmin database. |
| OCDS Scheduled Job Interval Minutes | This is the frequency with which Xadmin requests data from OCDS. The value should be greater than or equal to 30 minutes. |

Table 4–3 (Cont.) OCDS Configuration Options

| Setting | Description |
|--|--|
| OCDS On Demand Job Interval Minutes | This is the maximum amount of time that will elapse between when an on-demand job is created and is executed. The value should be greater than or equal to 5 minutes. |
| OCDS Orphan Data Protection Offset Seconds | This is the minimum amount of time data must age in OCDS before it can be visible to Xcenter. The offset helps to prevent related-data from becoming orphaned due to system latency. The value should be greater than or equal to 30 seconds. |
| OCDS Records Per Request Limit | This specifies the number of records to request in calls to OCDS. If no limit is defined then OCDS will determine the maximum number of records. |
| OCDS Retail Location: Till Accountability | This specifies if Retail Locations/Stores created from OCDS data are to use Till Accountability or not. |
| OCDS Retail Location: Default Locale | This specifies the Locale to use in Retail Locations created from OCDS data when a LANG_ISO_CODE is not defined for a location. This setting defines the value assigned to a Retail Location in the Xstore database if the store was not assigned a Locale in Merchandising |
| OCDS Item: Tax Group ID for non-taxable Items | This specifies the Tax Group ID assigned to non-taxable items created from OCDS data. |
| OCDS Item: Include future date to determine VAT code with the greatest active date | This specifies if future date should be included to determine VAT code with the greatest active date. |
| OCDS VAT Rounding Code | Specifies how the system will round when calculating the tax amount. Valid values are CEILING, DOWN, FLOOR, HALF_DOWN, HALF_EVEN, HALF_UP, and UP. For example, <code>ocds.vatRoundingCode=HALF_UP</code> |
| OCDS VAT Rounding Digits | Specifies how many decimal places to round to when calculating the tax amount. The minimum rounding digit is 0 and the maximum rounding digit is 10. For example, <code>ocds.vatRoundingDigits=2</code> |
| OCDS VAT Rounding at Transaction Level | Specifies the ability to round at transaction level when calculating the tax amount. When setting is set to <code>true</code> , the calculated tax will be rounded at the transaction level. For example, <code>ocds.vatRoundingAtTransLevel=true</code> When setting is set to <code>false</code> , the calculated tax is rounded at the line item level. |

Merchandising Chain to Xstore Organization Mapping

The Xstore Suite provides the opportunity to organize data by Organization, where centralized and store-level data can be isolated according to a retail organization structure. The Merchandising Suite supports the use of one or more Chains, where a chain can be used to group various store formats.

The Integration in the On Premise and Cloud Xstore Office requires that at least one Merchandising Chain be mapped to an Xstore Organization. This configuration must not be changed once the mapping is established.

Integration Database Tables

There are three database tables in the Xadmin database that are exclusively used for integration with Merchandising and Pricing for both integration methods described in this document. See the *Oracle® Retail Xstore Point-of-Service Software Database Dictionary* for complete details on these tables.

Table 4–4 Integration Database Tables

| Table Name | Description |
|----------------------|--|
| OCDS_JOB_HISTORY | When enabled in Xadmin, an integration job is executed by scheduled-interval or on-demand, to detect foundation data changes in integration service and extract them out into .mnt files. An entry in this table records the status, start/end time and other information of a job executed. |
| OCDS_ON_DEMAND | An entry in this table represents an on-demand job request. On-demand jobs can be system generated (as in the case of new store detection), or user generated (from the Xadmin UI). |
| OCDS_SUBTASK_DETAILS | An integration job executes a list of subtasks. Each subtask represents a foundation data area to detect and extract out changes. The table defines metadata for all subtasks. For a scheduled job, each and every active subtask is executed. For an on-demand job, a subset of subtasks specified for the job is executed. |

The only table that requires seed data is the OCDS_SUBTASK_DETAILS; the other two tables will populate during operations of the integrations.

Enabling and Disabling Data Types

Retailers can determine the types of data requested from the integration service by Xstore Office. For example, if VAT information is not needed, then the integration can be configured to prevent requests for VAT data.

In Xstore Office Cloud Service's Integration Manager, use the Configure tab when completing the wizard UI to establish the integration. In on-premise Xstore Office, set the value of `ocds.<Data Type>.properties` in `xcenter.properties` to establish which datatypes should be active for the integration.

Download: Immediate vs Store Close

The OCDS_SUBTASK_DETAILS table includes the column DOWNLOAD_TIME, which specifies when .mnt files should be downloaded for data loading at the store. The column contains null for those Subtasks that only populate the Xcenter database. The out-of-the-box seed data is configured so that only Pricing data will flow to the store immediately.

"NOT ON FILE" Items

The integration has been designed to not import items that represent an Xstore "NOT ON FILE" item. The integration compares Item Ids of item records in the Xstore Office database having the type code "NOT_ON_FILE" with item records in REST responses from the integration system so that it can skip records having the same item ID as an Xstore "NOT ON FILE" item.

Integration Considerations

This chapter provides the considerations that should be accounted for when implementing these solutions to minimize errors in data movement between solutions, as well as to call out some functional differences in the solutions that may limit the use of functionality in one or the other solutions.

Foundation Data

There are a number of basic data elements that are common between the Merchandising and store systems but which are not part of the integration. This is because they are generally a one-time set up at initial implementation with only infrequent updates afterward. However, because this data is foundational to how the solutions work, it is critical that they are set up properly. These data elements fall into a couple different categories:

- [Seed Data](#)
- [Transaction Details](#)
- [Currency Exchange Rates](#)

Seed Data

Seed data refers to data that is loaded into both solutions on implementation by Oracle Retail provided install scripts. These are coordinated between solutions as part of the base installation, but if any updates are made in one solution to add or remove items, the corresponding change should be made in the other solution. Data elements that fall into this category are:

- Currency codes
- Country codes
- Units of measure

Transaction Details

The mapping of transaction details from Xstore POSlog to Sales Audit RTLog depends on the mappings of valid values. These mappings are detailed in [Appendix A, "Appendix: Xstore to Sales Audit Mapping Details"](#). It is critical that the mappings are complete. If additional valid values are configured for Xstore in the RTLogMappingConfig.xml, they must also be configured for Sales Audit for the appropriate code types.

Similar to seed data, some initial data is provided for the data entities in this category, but this is an area that is more commonly configured for retailers based on their

specific business processes. On initial implementation, the configurations in both Xstore and Sales Audit should be made to be in synch, with any changes made post-implementation continuing to be made in both solutions. The entities in this category include:

- Transaction Types
- Tender Types
- Tender Total IDs
- Item Types
- Reason Codes
- Item Statuses
- Sales Types

See the [Appendix A, "Appendix: Xstore to Sales Audit Mapping Details"](#) for details on configuring and mapping these entities.

Multi-Line Text

Use of multi-line descriptions in Merchandising for data flowing to the Xstore Suite must be avoided because it can result in data loading failures by Xstore's DataLoader. All text fields used in the integration are expected to be single-line.

Currency Exchange Rates

Exchanges rates for currencies are not one of the things integrated between Merchandising and Xstore, as Merchandising is not considered the system of record for this information at a retailer - generally that comes from the financials solution. However, if you require currency exchange rates in Xstore, then it is expected that the same source of data used for exchange rates in Merchandising will also be used to load those rates into Xstore, in order to ensure both solutions are operating with the same information and to prevent a financial impact from occurring due to differences in the rates used. Tender exchange transactions that occur in Xstore, where a customer is given USD in exchange for CAD, will be mapped to the transaction type OTHER in Sales Audit.

Stores

By default, Xstore is configured to allow four digit store IDs, but it can be configured to hold up to 5 digit store numbers in the SequenceConfig.xml. Although Merchandising can hold up to a 10 digit store ID, when integrating with Xstore, it is strongly recommended that only four or five digit location IDs are used. Custom modifications would be required to Xstore to support larger store IDs.

Additionally, latitude and longitude information that is used by Xstore to determine nearby stores for its inventory lookup function are not available as part of the integration from Merchandising. If you wish to use this functionality in Xstore, the record type, RETAIL_LOCATION_COORDINATES, is available to DataLoader to populate the latitude and longitude of stores using the .mnt format.

All stores in Merchandising that you expect to get a sales file from Xstore should be set up to have Integrated Sales flag set to checked (Yes) as part of the store setup. Additionally, the store level attribute Unique Transaction Number should be set to Register for all Xstore integrated stores. These are both used by Sales Audit to know which stores to expect to receive files and how to audit the data in the files.

Merchandise Hierarchy

Xstore supports up to 4 levels of the merchandise hierarchy, which will be populated by the bottom four levels of the merchandise hierarchy from Merchandising - group, department, class, and subclass. In Merchandising, the class ID displayed to users is unique only when combined with its department ID. Similarly, the displayed subclass ID is only unique when combined with its department and class. However, instead of using the composite key in the integration with Xstore, the unique key that is held in the Merchandising tables for class and subclass used in the OCDS and is written into .mnt files. This unique ID is not visible to users of Merchandising.

The following is the Xstore Merchandise Hierarchy Spring bean configuration that should be used to map the bottom four Merchandising application's levels to the Xstore's four levels:

```
<bean id="merchandiseHierarchyInfo"
class="dtv.pos.common.MerchHierarchyLevelInfo">
  <property name="numberOfLevelsAvailable" value="4" />
  <property name="level1Code" value="GROUP" />
  <property name="level2Code" value="DEPARTMENT" />
  <property name="level3Code" value="CLASS" />
  <property name="level4Code" value="SUBCLASS" />
</bean>
```

Note: When overriding the default Xstore Merchandising Hierarchy levels, the default resource bundle text values should also be overridden accordingly.

Items

This section lists considerations regarding items.

Merchandise Items

Physical merchandise items should be mastered in Merchandising and use the integration described in this document to flow the data to Xstore. Xstore Office should not be used to create physical items in order to prevent errors when loading sales data into Sales Audit where the item being sold or returned cannot be identified and accounted for in Merchandising.

Non-Merchandise Items

If using non-merchandise items, such as warranties, fees, and services, in Xstore, special attributes are required that are not available in Merchandising. Therefore to configure these items, the following approach is required:

1. Create the non-merchandise item in the Xstore Office UI, specifying the required attributes to control its behavior in Xstore.
2. Create an item in Merchandising with the same ID as that created in Xcenter. The item created in Merchandising should be set up as a non-merchandise item to prevent it from being re-exported to Xstore.

The creation of the item in Merchandising will prevent any errors from occurring in the auditing process. Any maintenance on the non-merchandise items should occur in Xstore Office going forward.

Note: For more information on how non-merchandise items work between Sales Audit and Merchandising, see the *Oracle Retail Sales Audit Implementation Guide*.

To allow end users to create non-merchandise items, but be prevented from creating or editing merchandise items in Xstore, the `CFG_MERCH_ITEMS` privilege should not be granted to any users. The merch items option will still be on the screen, but it will not be accessible.

Kit/Pack Items

Kits, or pack items in Merchandising, are items that contain multiple components but are sold as a single unit. As part of the standard item integration, Xstore does not import the component level information from Merchandising, so these items will appear as standard items in Xstore and the component details will not be available.

Differentiators

Differentiators are used in Merchandising to define how a transaction level item (for example, SKU) differs from its parent (for example, style). For example, a differentiator might be a color, size, or flavor for an item. In Xstore, differentiators are called dimensions. Merchandising supports up to 4 differentiators/dimensions for items, while Xstore can support only three. It is strongly recommended that the 4th differentiator is not used when implementing Merchandising with Xstore, as it will be ignored in the integration.

Additionally, in Merchandising an item can be assigned differentiators without having a parent (style) associated with it. This could be used for hardline or grocery items to indicate the color or size of an item for reporting purposes, for example. However, in Xstore dimensions are primarily used to allow a user to determine the sellable SKU by entering a style ID and selecting the valid dimensions (usually color and size). Therefore, if an item does not have a parent, the dimensions sent from Merchandising will be ignored and will not be visible in Xstore.

Product Restrictions

Product restrictions can be set up in Merchandising to indicate limitations on certain products. For example, a restriction may be set up to limit alcohol from being sold to customers under a certain age. Product restrictions are not currently supported in the integration to Xstore. Custom integration would be required to communicate this information from Merchandising to Xstore.

Related Items

Merchandising has a concept of related items that can be used to define items that are substitutes for one another, or that could be used to cross-sell or up-sell to a customer when purchasing the main item. Substitute items from Merchandising are mapped to the Xstore substitute items to indicate items that may be substituted or offered in place of another item.

The cross-sell and up-sell types of related items are mapped to Xstore's Attached Items and configured as prompt-to-attach. Only transaction level related items are used by Xstore. Those created at the parent item level (for example, style) in Merchandising are ignored.

Other Item Attribute Notes

- Item Restocking - unlike Xstore, Merchandising does not have a flag that indicates whether an item is subject to an item restocking fee, nor the ability to define what an item's fee would be. Therefore, Xstore would not have the ability to prompt for a restocking fee during returns.
- Xstore can support prorated refunds for items, but to do so requires specific attributes sent for an item, which are not currently available in Merchandising. Therefore, this function would not be available in Xstore.
- Merchandising has the ability for retailers to extend the available item attribution by creating user defined attributes and custom flex attributes. Although included in the available data from Merchandising, these are currently not used by Xstore.
- Translated item descriptions are available from Merchandising as part of the integration but are not currently used by Xstore. Xstore uses the item level description (which is communicated in the primary Merchandising language) for Xoffice and the item/location level descriptions for the store in Xstore. If you have the requirement to send item descriptions in different languages to your stores, it is recommended that the item/location level description in Merchandising be updated to show the localized item description.

Tax

This section describes considerations regarding taxes.

Value Added Tax (VAT)

Merchandising integration includes VAT rates and the regions in which the stores have been classified for companies with operations in geographies where this type of tax is applicable. For retailers that have operations in both VAT and non-VAT regions - such as stores in the US and Canada - non-VAT regions are configured as exempt in Merchandising and communicated as such to Xstore. For more information on configuration for VAT in Xstore, see the *Oracle Retail Xstore Technical Guide*.

When Merchandising sends VAT rate updates for an item, it also includes the active date for the rate to be applicable. Retailers sometimes enter new VAT rates in advance for future planning. However, Xstore currently does not support an active date for VAT code and will ignore the active date sent, which means any new codes will go into effect immediately. Therefore, it is recommended that retailers enter the VAT code changes in Merchandising only when needed.

Note: Buying from a VAT store and returning to a non-VAT store (and vice versa) is not supported in Xstore.

US Sales Tax

Merchandising does not provide US Sales tax information to Xstore; it is assumed that product tax groups are imported into Xstore from a third-party system using Xstore Point of Service DataLoader and .mnt files.

After loading Merchandising data, the following additional steps are required to configure sales tax using the .mnt file format:

1. Set up sales tax rules. To set up a simple rate based tax rule, use existing record types TAX_LOCATION, TAX_AUTHORITY, TAX_GROUP, TAX_GROUP_RULE, and TAX_RATE_RULE to populate tax tables tax_tax_loc, tax_tax_authority, tax_

tax_group, tax_tax_group_rule, and tax_tax_rate_rule. For more details on tax rule configuration, see the TAXING section in the *Oracle Retail Xstore Point of Service Host Interface Guide* available on My Oracle Support.

2. Set up retail store and tax location mapping in table tax_rtl_loc_tax_mapping using existing record type TAX_RETAIL_LOCATION_MAPPING. For more details on this record type, see the TAXING section in the *Oracle Retail Xstore Point of Service Host Interface Guide* available on My Oracle Support.
3. ITEM_TAX_GROUP is used to update the item record in the itm_item_options table with sales tax group ID. This .mnt file has to be imported after the Merchandising data import. There is no built-in mechanism in DataLoader or Xstore Office to ensure this ordering. It has to be enforced by retailer manually.

Note: There is also a configuration of tax data that must be done in Sales Audit when using the US Sales Tax configuration. See [Appendix: Xstore to Sales Audit Mapping Details](#) for more details.

Inventory

Inventory functionality in Xstore should be disabled when implemented with Merchandising. No inventory information is integrated between Xstore and Merchandising other than sales related data and it is assumed store inventory is managed in another application, such as Oracle Retail Store Inventory Management (SIM) or Store Inventory and Operations Cloud Service (SIOCS), which is also integrated with Merchandising. Therefore, when these systems are all part of a retailer's implementation, the .sim entry in the configuration path should be used in Xstore to turn off Xstore inventory functionality. Inventory integration outside of sales and returns between Merchandising and Xstore is not supported.

Serialized Inventory

Merchandising supports the concept that an item can be a serialized item in one store, but not in another, however in Xstore, the designation for whether or not an item is serialized is held at the item level, so there is not any differentiation by store. This means that if the serialized flag actually varies by location for an item in Merchandising, the last location to be dataloaded by the integration code sets the item level serialized flag in Xstore.

Note: Merchandising does not support serialized inventory at this time. It only flags items as being serialized or not.

Customer Orders

When customer orders are initially captured in Xstore, the Xstore RTLog generator sets the Fulfillment order number in the RTLog to UNKNOWN, as the fulfillment order number is not known at the time the order is created, because information has not yet been sent to the order management system.

In-Store Orders

Orders taken in the store on behalf of a customer that do not go through an Order Management System (OMS) for fulfillment will include only a customer order number,

but not a fulfillment order number when the transactions related to it are integrated to Sales Audit.

Recognition of a Sale

For customer orders, Xstore can be configured to recognize a sale at either the time the order is placed or at the time of pickup. Integration with Merchandising requires that this configuration be time of pickup, which corresponds to when inventory is decremented from the store, in order to prevent out of sync issues between actual store inventory and what is shown in Merchandising.

In order to configure this in Xstore, the following settings should be set to false (which is the default) under both <Layaway> and <SpecialOrder> in SystemConfig.xml (whose settings are also controllable in Xadmin):

```
<Layaway>
<BookAsSaleOnSetup dtype="Boolean">false</BookAsSaleOnSetup>
<SpecialOrder>
<BookAsSaleOnSetup dtype="Boolean">false</BookAsSaleOnSetup>
```

Pricing

In both Merchandising and Pricing the data type for retail prices is NUMBER(20,4), but in Xstore, the standard is to use a data type of NUMBER(17,6). This applies to the following item prices:

- Selling Unit Retail (from Merchandising and Pricing)
- Manufacturer's Recommended Retail (from Merchandising)

If a Pricing retail value is over 17 digits, dataloading into Xstore will fail. Non-failing records from the same file will continue to be loaded.

Multi-Unit Pricing

Pricing and Xstore have different approaches to multi-unit pricing. Xstore converts multi-unit prices to single price, but cannot accurately do this without rounding information which varies from store to store. As such, Pricing's regular price update information for multiple units is not supported in this integration.

Promotions

There are certain features of promotions that are supported in Pricing that are not supported in Xstore. These features should be configured off in Pricing to prevent users from creating promotions that cannot be executed. These are the features that should be configured off:

Allow Reward List Exclusions on Transaction Offers

This Pricing system option should be unchecked, as Xstore cannot support receiving reward list exclusions for transaction offers.

Allow Supplier Site and Brand for Offer Item Selection

This Pricing system option should be unchecked, as Xstore does not have information about the brand or supplier for an item; therefore, adding or excluding items by this type of criteria on a promotion cannot be supported in the integration.

Pricing Application - Promotion Offer Distribution Rules (OFDR)

Pricing allows users to define how the discount should be distributed to items on a customer's purchase by selecting a distribution rule as part of the promotional offer creation. However, Xstore always applies the discount to the "get" items on an offer. Therefore, this code should be configured to only allow the Get Items option to be selected by users. See also the *Oracle Retail Pricing Implementation Guide* for more details.

Sales Audit

This section describes configurations that should be made in Sales Audit, as well as some limitations that exist.

First, in addition to configuring the store setup as described above, there are some Sales Audit system options that must be configured in a particular way when integrating with Xstore:

- Transaction Appended with Workstation ID – this should be set to unchecked (No); when checking for duplicate transactions from Xstore, Sales audit will look at the register field in the transaction, along with the transaction number.
- Credit Card Masking Character – this should be set to * when integrating with Xstore, as that is how Xstore masks credit card numbers.
- Balancing Level - when integrated with Xstore, Sales Audit should be configured with a balancing level of Register and Xstore will always sends the workstation ID as the register.

Store Data Configuration

Sales Audit requires configuration to be set up for each store where you expect to import data. For all Xstore stores, you'll need to include two lines for each store configuration – one for Sales Import and one for Item Level Tax. Sales Import is the normal store sales import configuration, and the Item Level Tax configuration is used in validation of the import file to look for tax at the item level instead of the transaction level. This is how tax data will always be sent from Xstore, regardless of tax type.

Register-level Balancing

Xstore workstation and Sales Audit register are equivalent concepts; however Sales Audit does not have an entity equivalent to the Xstore till, which means that Xstore cannot be configured for till-level balancing when integrated with Sales Audit.

Sales Person

In Xstore, the sales person field length can be up to 60 characters in length, but Sales Audit only allows up to 10 characters. Retailers should, as a business process, not use Xstore sales person IDs with more than 10 characters.

Additionally, Xstore allows multiple sales associates at the line item level, however Sales Audit only supports one. Therefore only the transaction level sales associate is exported to Sales Audit.

Tender Types

Xstore supports a tender type of Home Office Check, which is not supported by Sales Audit. Retailers using this integration should not use the Xstore Home Office Check

tender type. See [Appendix: Xstore to Sales Audit Mapping Details](#), for more information on tender type mapping between solutions.

Coupons

Bounce back coupon number length in Xstore can be 60 characters long, but Sales Audit only allows 40 characters. If retailers want to use the integration, they should as a business process, not use IDs with more than 40 characters.

For more Sales Audit related considerations and configurations, see [Appendix: Xstore to Sales Audit Mapping Details](#) in this document.

File Format

The RTLOG file that is sent from Xstore to Sales Audit uses a version that does not include the following components in the file format:

- Transaction Header level:
 - Reference number 28-31
- Transaction Item level:
 - Fulfillment Location Type
 - Fulfillment Location

For more information on how these fields are used in Sales Audit, see *the Oracle Retail Merchandising Operations Guide – Volume 1*.

Employee IDs

A new employee can be created using the Employee Maintenance function in Xstore. By default, the employee ID is generated automatically based on the employee.seq, for example, 0219001000009. The first four digits are the store ID, the next three digits are the register ID, and the last six digits are the sequence ID. If this new employee is selected as a sales associate, an exception will be thrown in the RTLogGenerator, since the length of a sales person ID, defined in the RTLogFormatConfig file is ten, whereas the length of the auto generated employee ID is 13. In Sales Audit, an associate ID cannot be over ten characters long.

RTLog Generator On-Premise

This chapter describes how to install, deploy, and configure the on-premise RTLog Generator application. The On Premise RTLog Generator application can be used with both the cloud or on premise Merchanding/Pricing applications.

RTLog Generator is a Java and XML based web application that exposes a Spring-JAXWS implemented SOAP web service. It is distributed as a web archive along with a configuration .zip file ready to be deployed on an Oracle WebLogic Server 14c.

This chapter uses Microsoft Windows path format as the example for paths.

Configuration

The RTLog Generator application is shipped with a configuration .zip file (rtlog-gen-config.zip) which should be used to externally configure and extend the RTLog Generator's functionality.

Note: Bounce the WebLogic server after making any configuration level changes.

Starting from release 20.0, the RTLog generator is shipped with one rtlog-generator-config.zip, which is used to integrate with Sales Audit and ReSA On-premise.

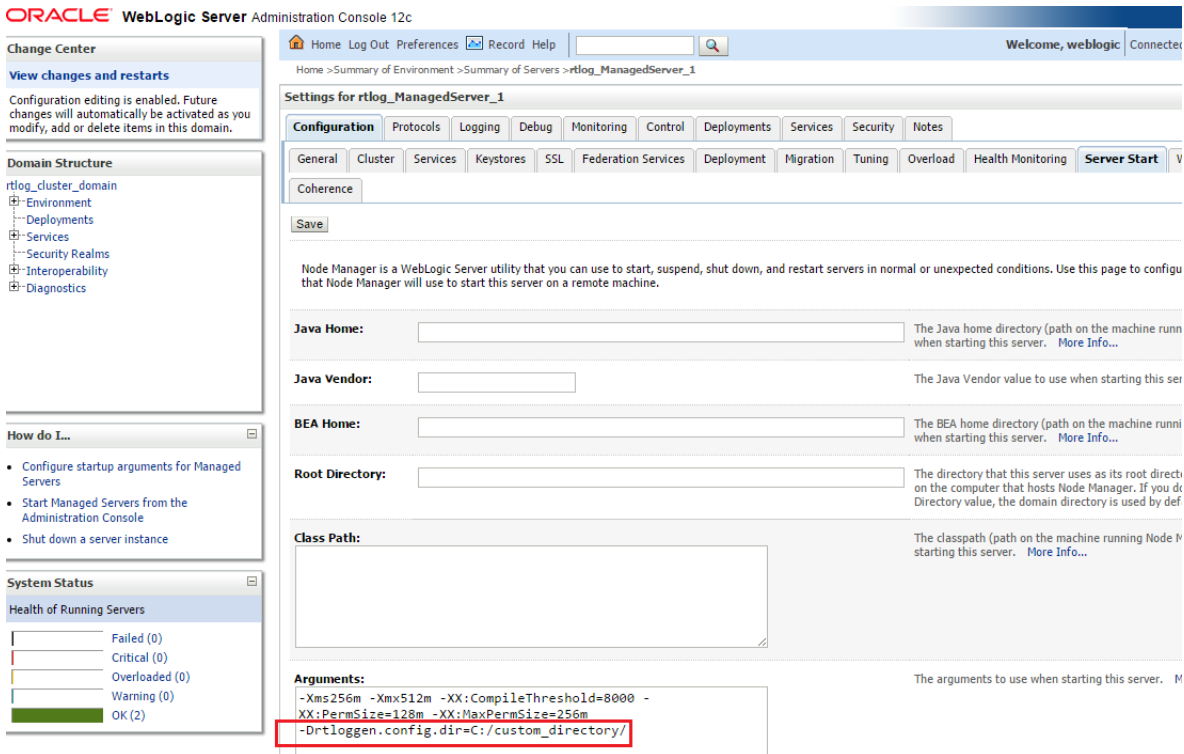
To set up the external configuration features:

1. Extract the configuration file's content into the `C:\<rtlog-generator-config>` directory if installing on Microsoft Windows or `/usr/local/<rtlog-generator-config>` on Linux OS. These directories are the default locations where the RTLog Generator application will look for the configuration files. These default locations can be overridden/changed by using one of the following ways:

- Pass a JVM argument to the server startup script and bounce the server:
`-Drtloggen.config.dir=C:/<custom_directory>/`

If the WebLogic domain is created with a Node manager, the same argument can be passed from the Administration Console in the Arguments field. See [Figure 6-1](#).

Figure 6–1 Administration Console Configuration Page



- Specify the context-param field in the RTLog Generator WAR file. This requires opening up the WAR file and making the required changes. Update the web.xml file as shown in the following example:

```
<context-param>
    <param-name>rtlog.generator.config.home</param-name>
    <param-value>C:/<custom_directory></param-value>
</context-param>
```

Figure 6–2 Example of context-param Field Update

```
1 <?xml version="1.0" encoding="UTF-8"?>
2 <web-app xmlns="http://java.sun.com/xml/ns/javaee" xmlns:web="http://java.sun.com/xml/ns/javaee/web-
  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" xsi:schemaLocation="http://java.sun.com/xml/ns
  http://java.sun.com/xml/ns/javaee/web-app_3_0.xsd" version="3.0">
3   <!--display-name>RTLOG-GENERATOR</display-name>
4   <context-param>
5     <param-name>contextConfigLocation</param-name>
6     <param-value>/WEB-INF/classes/applicationContext.xml</param-value>
7   </context-param>
8   <!-- Customizable external location for RTLog config files-->
9   <context-param>
10    <param-name>rtlog.generator.config.home</param-name>
11    <param-value>C:/custom_directory</param-value>
12  </context-param>
13  <!-- Customizable RTLog generator app_name. If not unspecified, it remains "rtlog-generator". This
14  <context-param>
15    <param-name>rtlog.generator.application.name</param-name>
16    <param-value />
17  </context-param>
18  <!-- Customizable external log4j xml file. Specify just the file name without any extension. By de
19  <context-param>
20    <param-name>rtlog.generator.config.log4j</param-name>
21    <param-value />
22  </context-param>
```

The JVM argument takes the precedence over the default location, that is, C:\<rtlog-gen-config>. If either of the two does not exist, the context parameter is used. If nothing is specified, the RTLog Generator application will fail on startup with error messages in the server logs.

2. Once the configuration file is extracted to the configured directory, verify the following files:

- `rtlogconfig.properties`:

This file contains the following properties (key value pairs):

- `processingDir`: This directory path specifies the location that RTLog Generator will use to build its RTLog files as it receives data from Xstore Office. This directory needs to be created manually.
- `resaFileDropDir`: This directory path specifies the destination for the RTLog files this system is producing. It should be configured to the location where Sales Audit is looking to receive the RTLog files. This directory needs to be created manually.
- `clusterNodeNumber`: This property should only be enabled when running in a clustered environment. For more information, see "[WebLogic Cluster Setup](#)".
- `deliveryTarget`: This property specifies the RTLog delivery target. The two possible values are `resa-cs` or `resa-onprem`. If `resa-cs` is specified, the RTLog files generated are in format compatible with Sales Audit; otherwise the files generated are in format compatible with ReSA On-premise.

Following is an example of the properties:

```
processingDir = C:/RTLOG_Weblogic/Output/Store/RTLOGS
resaFileDropDir = C:/RTLOG_Weblogic/Output/ReSA
clusterNodeNumber = 1
deliveryTarget = resa-cs
-OR-
deliveryTarget = resa-onprem
```

- Multiple `log4jx.xml` files: These files configure the logging levels for the RTLog Generator application.
 - `rtlog-generator-log4j2.xml`
 - `rtlog-generator-log4j2-loggers.xml`
 - `rtlog-generator-log4j2-appenders.xml`
- `spring-scheduler.xml`:

It is the most commonly modified file in the RTLog Generator application. It is used to configure the scheduled interval for publishing the RTLog files. In the case of trickle polling, the default interval should be 15 minutes, however, keeping a larger interval (at least greater than or equal to 15 minutes) is recommended as configuring with a smaller interval might affect the performance.

Figure 6–3 *spring-scheduler.xml Example*

```

<?xml version="1.0" encoding="UTF-8"?>
<beans xmlns="http://www.springframework.org/schema/beans" xmlns:p="http://www.springframework.org/schema/p"
       xmlns:task="http://www.springframework.org/schema/task" xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
       xsi:schemaLocation="http://www.springframework.org/schema/beans
       http://www.springframework.org/schema/beans/spring-beans-3.0.xsd
       http://www.springframework.org/schema/task
       http://www.springframework.org/schema/task/spring-task-3.0.xsd">

<task:scheduled-tasks scheduler="rtlogScheduler">

<!-- To publish files once every 10 minutes = 600000 milliseconds 15 minutes = 900000 milliseconds
     1 hour = 3600000 milliseconds in fixed-delay below.
     It is not supported if fixed-delay is less than three second (3000 milliseconds). -->
<task:scheduled ref="rtLogFilesPublisher" method="publishFilesToReSA" fixed-delay="900000" />

<!-- You can also use "cron syntax". This simplistic example publishes files once every 5 minutes -->
<!-- <task:scheduled ref="rtLogFilesPublisher" method="publishFilesToReSA" cron="0 */5 * * * ?"/> -->
</task:scheduled-tasks> <task:scheduler id="rtlogScheduler" />

<task:annotation-driven />

</beans>

```

3. Customization out of box configuration

To customize RTLogMappingConfig.xml and RTLogFormatConfig.xml, you can obtain your out-of-box version in the rtlog-generator.war file under WEB-INF\lib\config.jar\resources\resa-cs for Sales Audit integrations (deliveryTarget = resa_cs) or WEB-INF\lib\config.jar\resources\resa-onprem for ReSA On-premise integrations (deliveryTarget = resa_onprem).

Customized copies of RTLogMappingConfig.xml and RTLogFormatConfig.xml are placed in the configuration directory defaulting to C:\<rtlog-generator-config> on Microsoft Windows or /usr/local/<rtlog-generator-config> on Linux OS.

- RTLogFormatConfig.xml:

This file specifies the format of the RTLog record as specified by Sales Audit. You can modify this file.

- RTLogMappingConfig.xml:

The RTLog Generator application relies heavily on the XML-based mapping which provides extensibility and a way to maintain/upgrade features for the application. This file can be used to override all the field values for either mapping strategy:

- FieldMapperThenValueMapping:

The RecordValue attribute values as shown in the following example can be changed:

```

<MAP sourceField="tenderId" targetRecord="TransactionHeaderTotal"
     targetField="ReferenceNumber1"
     mappingStrategyOrder="FieldMapperThenValueMapping"
     fieldMapper="trnHeaderTotalMapper">
  <VALUE_MAPPINGS handleNotFound="success"> <VALUE_MAPPING
     sourceValue="GIFT_CERTIFICATE" RecordValue="GIFTCERT" />
  <VALUE_MAPPING sourceValue="HOUSE_ACCOUNT" RecordValue="HACCNT" />
  <VALUE_MAPPING sourceValue="ISSUE_STORE_CREDIT" RecordValue="ISTCRDT" />
  <VALUE_MAPPING sourceValue="ISSUE_MERCHANDISE_CREDIT_CARD"
     RecordValue="IMCCARD" />
  <VALUE_MAPPING sourceValue="ISSUE_XPAY_GIFT_CARD"
     RecordValue="IXPAYGC" />
  <!--For e.g above given value can be changed as shown here.-->

```

```

    <VALUE_MAPPING sourceValue="ISSUE_XPAY_GIFT_CARD" RecordValue="SAMPLE_
    IXPAYGC" />
    <VALUE_MAPPING sourceValue="MALL_CERTIFICATE" RecordValue="MALLCERT"
    />
    <VALUE_MAPPING sourceValue="MERCHANDISE_CREDIT_CARD"
    RecordValue="MCCARD" />
    <VALUE_MAPPING sourceValue="PAYPAL" RecordValue="PAYPAL" />
    <VALUE_MAPPING sourceValue="COUPON" RecordValue="QPON" />
    <VALUE_MAPPING sourceValue="ROOM_CHARGE" RecordValue="ROOMCHAG" />
    <VALUE_MAPPING sourceValue="RELOAD_XPAY_GIFT_CARD"
    RecordValue="RXPAYGC" />
    <VALUE_MAPPING sourceValue="RELOAD_MERCHANDISE_CREDIT_CARD"
    RecordValue="RMCCARD" />
    <VALUE_MAPPING sourceValue="STORE_CREDIT" RecordValue="STCRDT" />
    <VALUE_MAPPING sourceValue="XPAY_GIFT_CARD" RecordValue="XPAYGC" />
  </VALUE_MAPPINGS>
</MAP>

```

- No `mappingStrategyOrder` and `fieldMapper` attributes are defined.

The `RecordValue` attribute values shown in the following example can be changed or a new value can be added:

```

<MAP sourceField="reason" targetRecord="TransactionHeader"
targetField="ReasonCode">
  <VALUE_MAPPINGS handleNotFound="nextMapping">
    <VALUE_MAPPING sourceValue="PI1" RecordValue="PI1" />
    <VALUE_MAPPING sourceValue="PI2" RecordValue="PI2" />
    <VALUE_MAPPING sourceValue="PI3" RecordValue="PI3" />
    <VALUE_MAPPING sourceValue="P01" RecordValue="P01" />
    <VALUE_MAPPING sourceValue="P02" RecordValue="P02" />
    <VALUE_MAPPING sourceValue="P03" RecordValue="P03" />
    <VALUE_MAPPING sourceValue="P04" RecordValue="P04" />
    <VALUE_MAPPING sourceValue="P05" RecordValue="P05" />
    <VALUE_MAPPING sourceValue="SAMPLE" RecordValue="SAMPLE_VALUE" />
  </VALUE_MAPPINGS>
</MAP>

```

Note: For more information on how to customize the RTLog Generator, see the *Retail Xstore - RTLog Generator Extension Guidelines (Doc ID 2174095.1)* on <https://support.oracle.com>.

Deployment

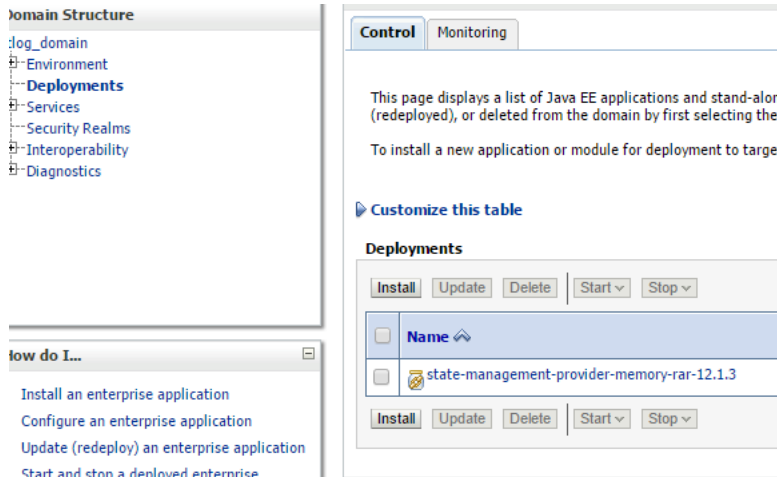
If you are deploying in a cluster, first set up a WebLogic cluster. For more information, see "[WebLogic Cluster Setup](#)".

This section covers the deployment in both a clustered and non-clustered environment.

To deploy the RTLog Generator application:

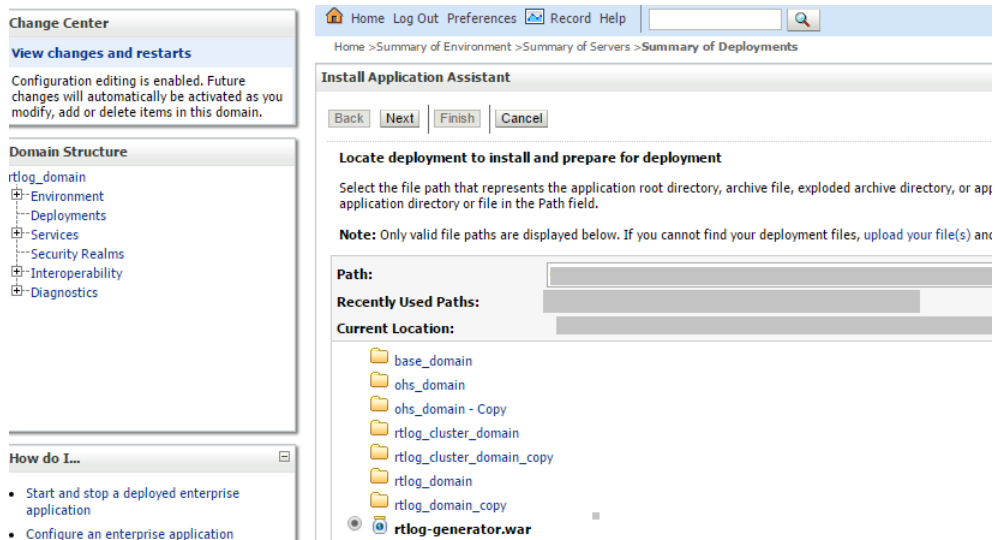
1. Log in to the WebLogic 12 Server Administration Console (<http://<hostName>:<port>/console>).
2. Click the Deployment link from the left navigation menu.
3. Click **Install**.

Figure 6–4 Administration Console Control Page



4. Navigate to the rtlog-generator.war file directory. Select the rtlog-generator.war option.

Figure 6–5 Administration Console Install Application Assistant Page



5. Click **Next** and then **Finish**. Once deployed, RTLog Generator should be listed as one of the deployed applications as shown in [Figure 6–6](#).

Figure 6–6 Administration Console Summary of Deployments

The screenshot shows the Oracle WebLogic Server Administration Console interface. The main content area is titled "Summary of Deployments" and includes a "Messages" section with two green checkmarks: "All changes have been activated. No restarts are necessary." and "The deployment has been successfully installed." Below this is a "Summary of Deployments" section with "Control" and "Monitoring" tabs. A text block explains that the page displays a list of Java EE applications and stand-alone application modules. Below this is a "Customize this table" link and a "Deployments" table with columns for Name, State, Health, and Type. The table lists two items: "rtlog-generator" (Web Application) and "state-management-provider-memory-rar-12.1.3" (Resource Adapter), both with "Active" state and "OK" health. At the bottom of the table are "Install", "Update", "Delete", "Start", and "Stop" buttons.

| Name | State | Health | Type |
|---|--------|--------|------------------|
| rtlog-generator | Active | OK | Web Application |
| state-management-provider-memory-rar-12.1.3 | Active | OK | Resource Adapter |

Once the deployment is complete, following are the next steps:

- To deploy on a cluster, see "[Deployment of the RTLog Generator Application on a Cluster](#)".
- To enable security for the RTLog Generator application, see "[Security Configuration](#)". When deploying in a non-clustered environment, continue at this section.

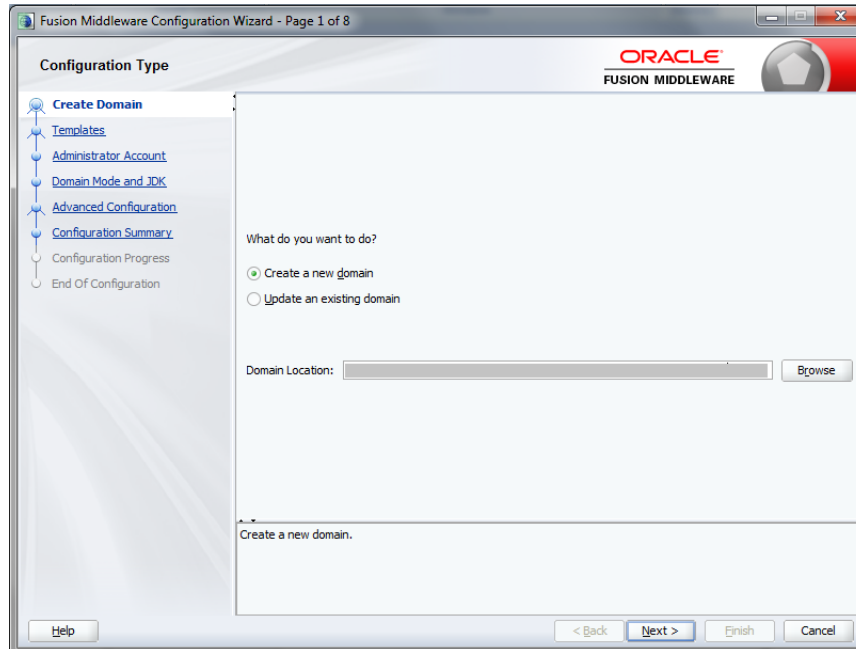
WebLogic Cluster Setup

Note: WebLogic Server 14c must be installed on all the clustered machines and the exact same installed directory location must be used on all the machines.

To set up the cluster to use RTLog Generator:

1. Start the WebLogic configuration wizard on one machine where the Administration server needs to reside.
2. On the Configuration Wizard Configuration Type page, select **Create a new domain**. Enter or browse to the location for the domain. Click **Next**.

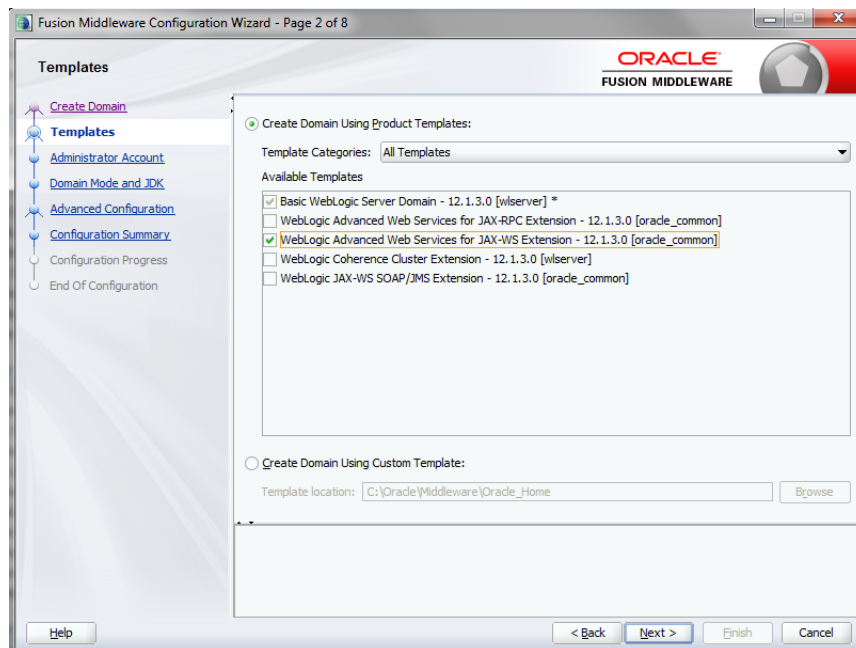
Figure 6–7 Configuration Wizard Configuration Type Page



3. On the Templates page, select the supported products and click **Next**. It is recommended to select the following:

- WebLogic Advanced Web Services for JAX-WS Extension

Figure 6–8 Configuration Wizard Templates Page



4. On the Administrator Account page, enter the Administrator user name and password. Enter the password a second time to confirm. Click **Next**.

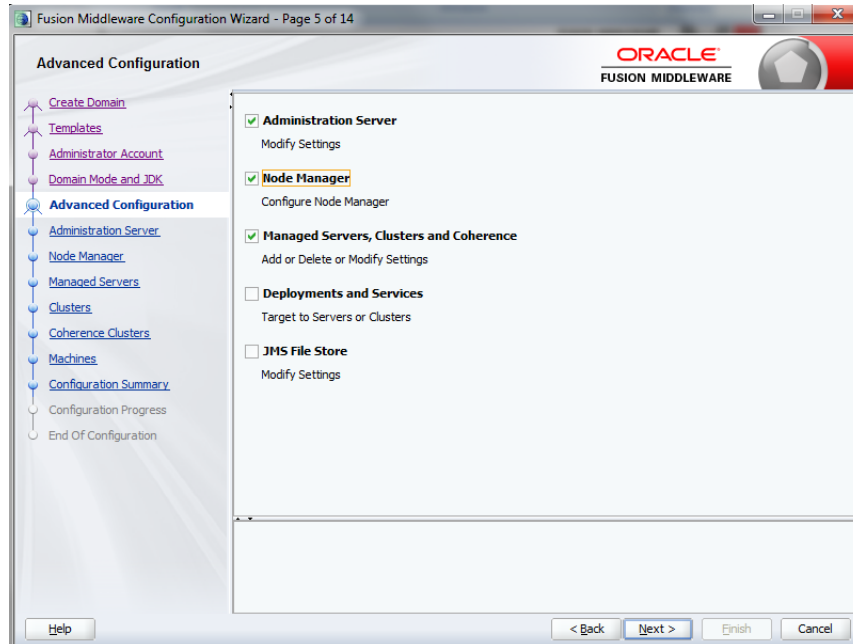
Figure 6–9 Configuration Wizard Administrator Account Page

5. On the Domain Mode and JDK page, select either Development or Production mode. For production mode, you need to manually create the boot.properties file. Click **Next**.

Figure 6–10 Configuration Wizard Domain Mode and JDK

6. On the Advanced Configuration page, select the Administration Server, Node Manager, and Managed Servers, Clusters and Coherence options. Click **Next**.

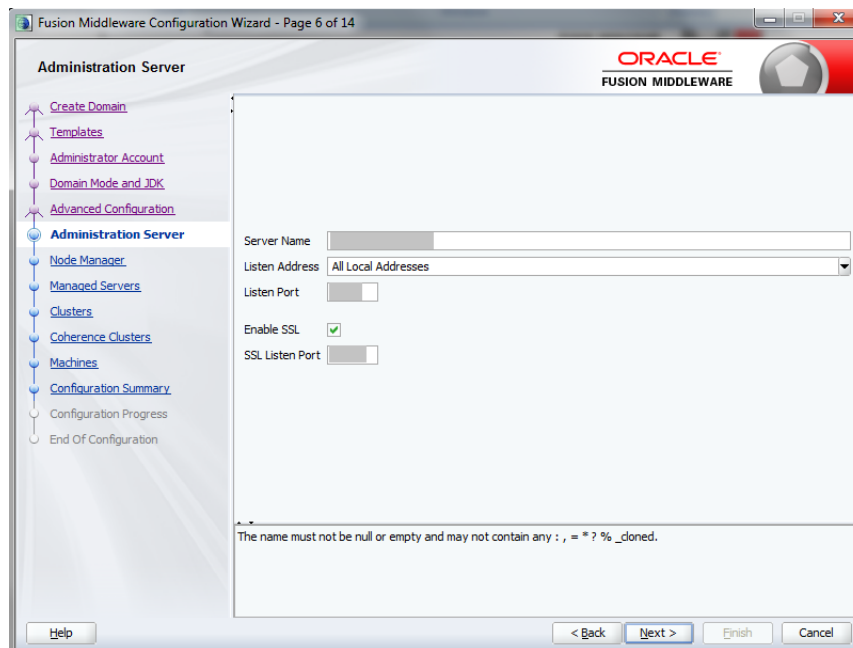
Figure 6–11 Configuration Wizard Advanced Configuration Page



7. On the Administration Server page, enter the values to configure the administration server. The administrator server controls all the managed servers that are part of the cluster.

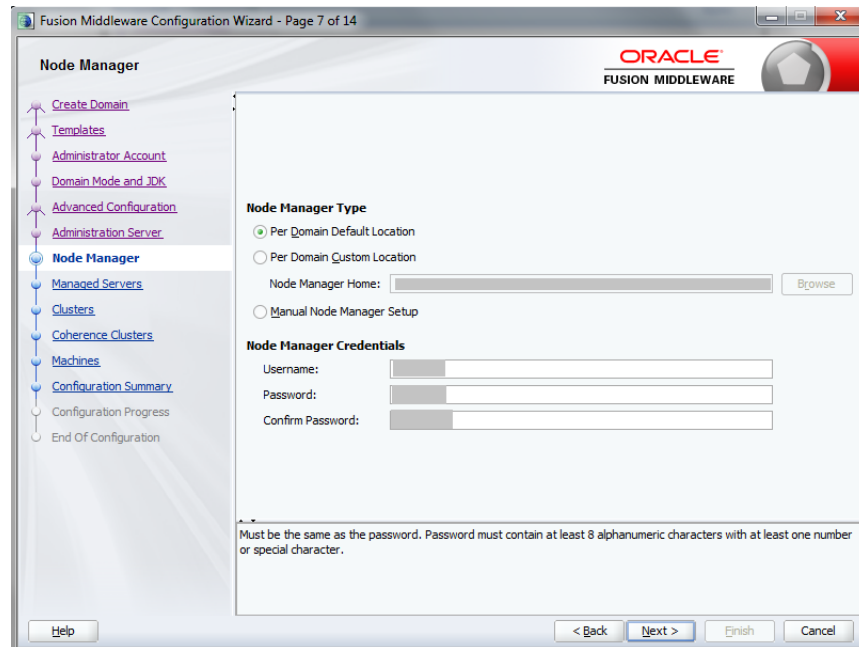
Enter the server name, select Enable SSL, and enter the listen ports. For the listen address, enter the Machine_1 IP address. Machine_1 will be part of the cluster and will have the administrator server running on it. Click **Next**.

Figure 6–12 Configuration Wizard Administration Server Page



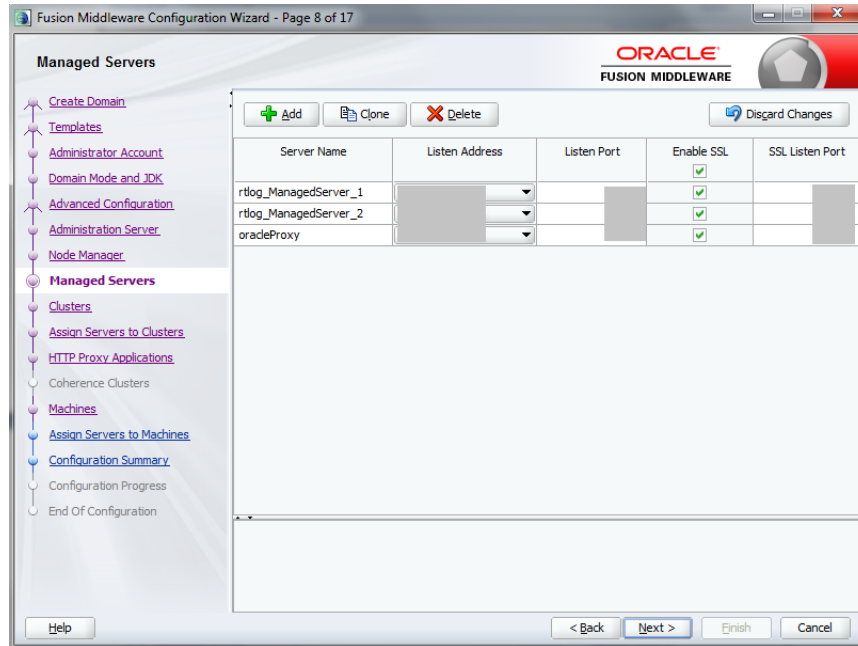
8. On the Node Manager page, do not change the default node manager settings. For the credentials, enter weblogic as the user name and enter the password. Click **Next**.

Figure 6–13 Configuration Wizard Node Manager Page



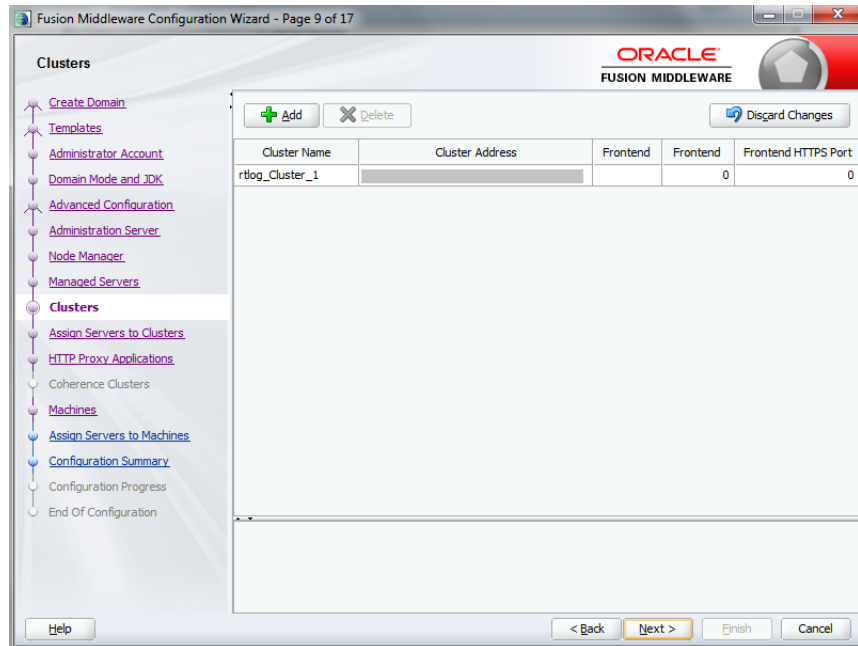
9. On the Managed Servers page, add and configure each managed server:
 - a. For the listen address, enter the IP address of the managed server. Do not select All local Addresses.
 - b. rtlog_ManagedServer_1 will be running on Machine_1 in this configuration. Enter the Machine_1 IP address for the server.
 - c. rtlog_ManagedServer_2 will be running on Machine_2 in this configuration. Enter the Machine_2 IP address for this server.
 - d. oracleProxy is running on Machine_1, but is not a part of the cluster. It is an Oracle proxy HTTP cluster servlet used for failover and load balancing purposes. Enter the Machine_1 IP address for this server.
 - e. Enable SSL for all the managed servers.
 - f. Click **Next**.

Figure 6–14 Configuration Wizard Managed Servers Page



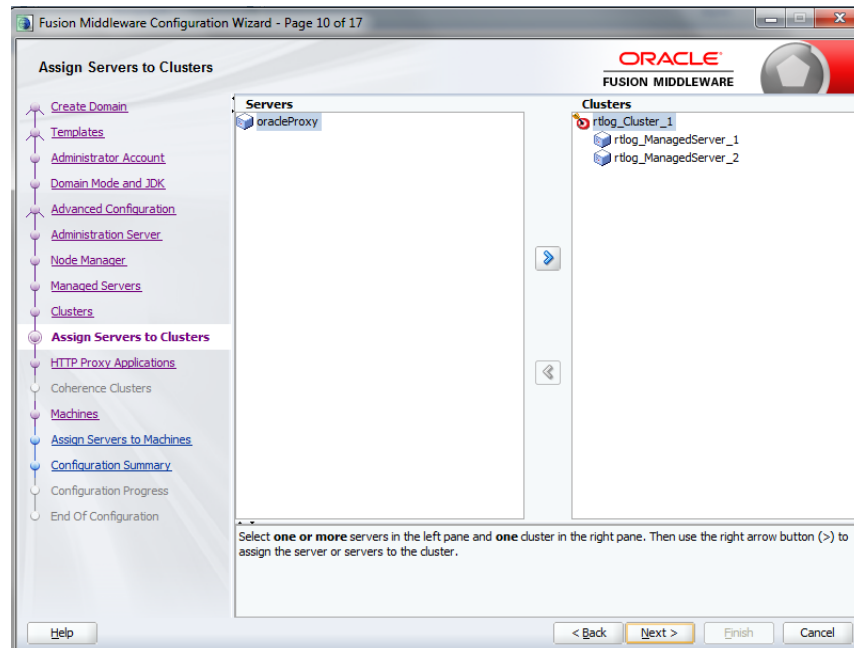
10. On the Clusters page, add and configure the cluster. Enter the cluster name followed by the cluster address, that is, IP address1:port1, IP address2:port2, so on. Click **Next**.

Figure 6–15 Configuration Wizard Clusters Page

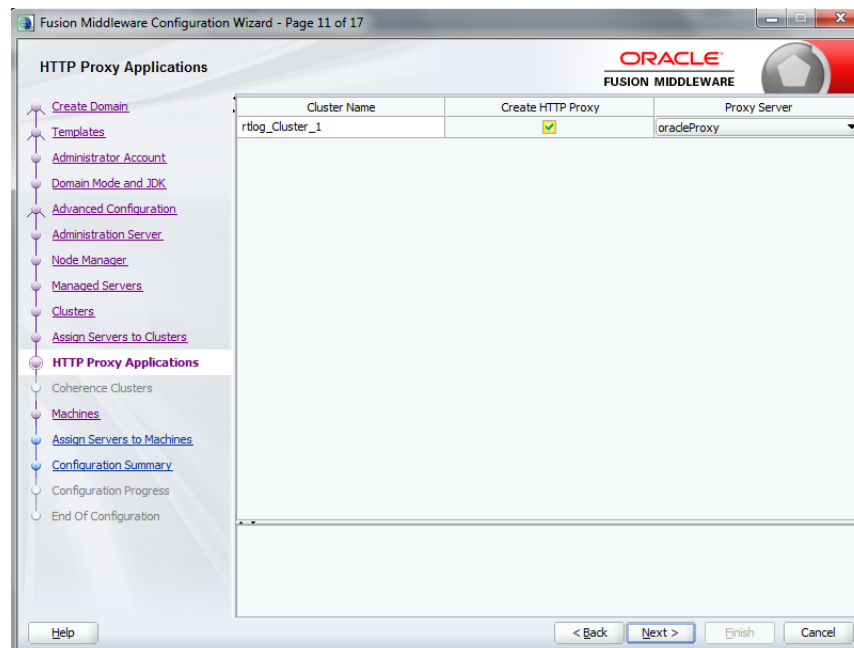


11. On the Assign Servers to Cluster page, assign the managed servers to the cluster. and click **Next**.

Note: Do not include the Oracle Proxy as part of the cluster.

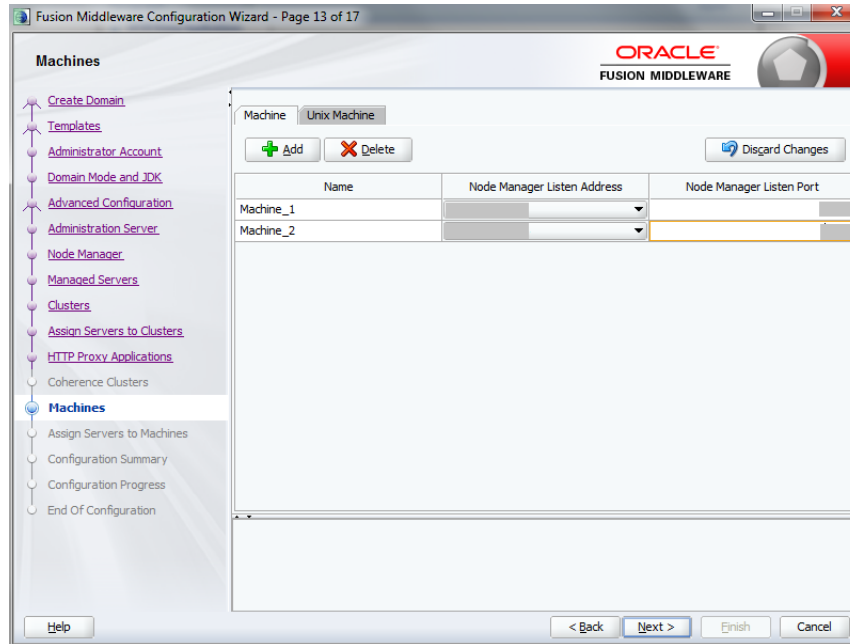
Figure 6–16 Configuration Wizard Assign Servers to Clusters Page

12. On the HTTP Proxy Applications page, select Create HTTP Proxy and then select the server from the drop-down list. By default, it should have already been selected. Click Next.

Figure 6–17 Configuration Wizard HTTP Proxy Applications Page

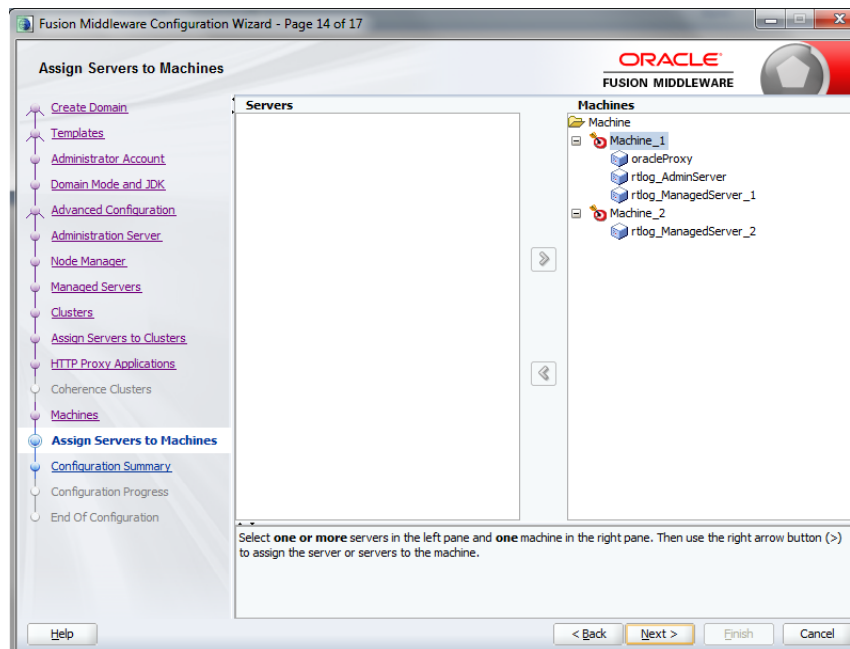
13. On the Machines page, add and configure each machine. To add Machine_1 and Machine_2, click **Add** and enter the respective IP addresses. This configuration is for setting up the Node managers on both the machines. Since these node managers are physically separated, you can select the same host. Click Next.

Figure 6–18 Configuration Wizard Machines Page

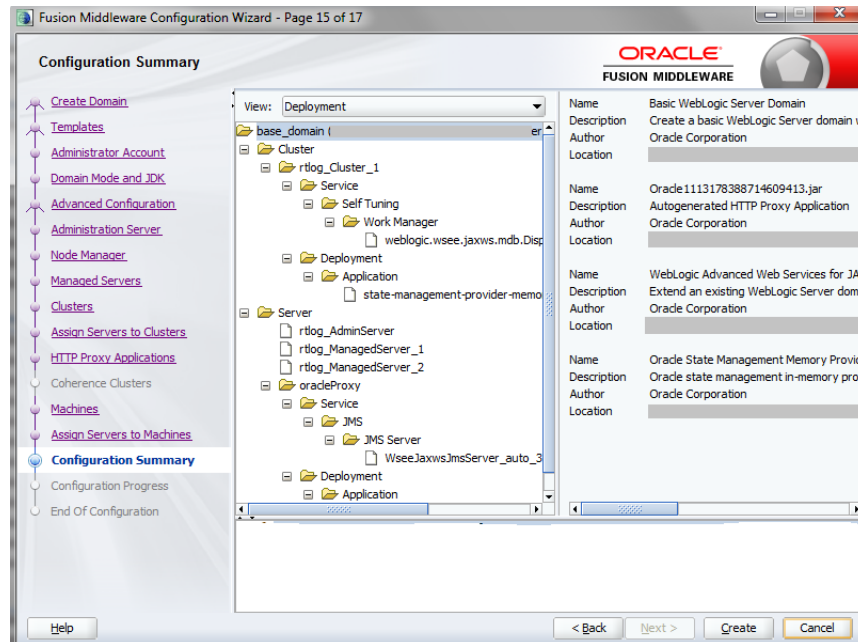


- On the Assign Servers to Machines page, assign the servers to the machines. In this example, Oracle proxy (load balancer), Administration server, and one managed server are configured on Machine_1. Another managed server is configured on Machine_2. Click Next.

Figure 6–19 Configuration Wizard Assign Servers to Machines Page



- On the Configuration Summary page, verify the selected configuration. Click **Create**. The domain is created.

Figure 6–20 Configuration Wizard Configuration Summary Page

To complete the configuration of the cluster:

1. Start and stop the node manager. You can find the start up script inside the newly created domain, that is, the `<rtlog_clust_domain>\bin` directory.
2. In the `nodemanager.properties` file, set `SecureListener=false`. This file is found in the `<rtlog_clust_domain>\nodemanager` directory.
3. Edit the `<rtlog_clust_domain>\config\config.xml` file. Use plain communication for the node managers by updating the communication type for the node managers as shown in the following example:

```
<machine>
  <name>Machine_1</name>
  <node-manager>
    <name>Machine_1</name>
    <nm-type>Plain</nm-type>
    <listen-address>xxx.x.xxx.xx</listen-address>
  </node-manager>
</machine>
<machine>
  <name>Machine_2</name>
  <node-manager>
    <name>Machine_2</name>
    <nm-type>Plain</nm-type>
    <listen-address>xxx.x.xxx.xxx</listen-address>
  </node-manager>
</machine>
```

4. If the `<rtlog_clust_domain>` is created with the production mode option:
 - a. Run `<rtlog_clust_domain>\startWeblogic.cmd` for the first time. This creates the servers folders under the domain. Enter the administration user name and password.
 - b. Create a folder named `security` under the `<rtlog_clust_domain>\servers\Admin server`.

- c. Create the boot.properties file with the following entries under the security folder:

```
password=%admin_server_password%
username=%admin_server_username%
```

%admin_server_password% and %admin_server_username% are the administrator password and user name.

- d. After making these changes, if there are any running processes, shut down all the processes.

5. Pack the created domain:

- a. Stop both the Node manager and Admin Server if not already stopped. Use the packing utility to pack the domain on the machine. This utility is found in the following location:

```
<WL_HOME>\wlsrserver\common\bin\pack.cmd
```

Run the following command:

```
pack.cmd -domain=<WL_HOME>\user_projects\domains\rtlog_cluster_
domain -template=<WL_HOME>\user_projects\domains\rtlog_cluster_
domain\rtlog_cluster_domain.jar -template_name="RTLog C domain"
```

This command creates a jar named rtlog_cluster_domain.jar by packing the complete domain into it. Copy the rtlog_cluster_domain.jar to Machine_2 and unpack it.

- b. Create a <user_templates> directory on the remote machine and copy the rtlog_cluster_domain.jar file to this location. Run the following command:

```
unpack.cmd -template=<WL_HOME>\user_projects\domains\<user_
templates>\rtlog_cluster_domain.jar -domain=<WL_HOME>\user_
projects\domains\rtlog_cluster_domain
```

- c. Start the Administration server and node manager on Machine_1.

6. To enroll the remote (Machine_2) node manager:

- a. Run the WebLogic scripting utility. This utility can be found at the following location: <WL_HOME>\wlsrserver\common\bin\as\wlst.cmd

- b. Start the node manager on this machine, in this example, Machine_2. The node managed must be started before connecting to the Machine_1 Admin server.

- c. Run the following command:

```
connect ('adminServer_username', 'adminServer_password','t3://Machine_1_
IPAddress:Admin_server_unsecured_port')
```

- d. Once the connect command shows the connection completed successfully, run the following command:

```
nmEnroll ('<WL_HOME>/user_projects/domains/<rtlog_cluster_
domain>','<WL_HOME>/user_projects/domains/<rtlog_cluster_
domain>/nodemanager')
```

- e. When the command completes successfully, run exit ().

Note: Repeat Step 6 for all the remote machines that will be in the cluster on which managed servers will be running. This step used Machine_2 as the example.

- Log in to the Administration Server console and make sure all the node managers are reachable. This can be found under Machines. Repeat this step for all the clustered machines to ensure all of them are reachable.

Figure 6–21 Administration Console Settings Page

ORACLE WebLogic Server Administration Console 12c

Home > Summary of Environment > Summary of Machines > Machine_1

Settings for Machine_1

Configuration Monitoring Notes

Node Manager Status Node Manager Log

This page allows you to view current status information for the Node Manager instance of

| | |
|-----------------|-----------|
| Status: | Reachable |
| Version: | 12.1.3 |

Change Center
View changes and restarts
Configuration editing is enabled. Future changes will automatically be activated as you modify, add or delete items in this domain.

Domain Structure
rtlog_cluster_domain
Environment
Deployments
Services
Security Realms
Interoperability
Diagnostics

How do I...
Monitor Node Manager status

- For each managed server, select the Server Start tab. In the Arguments text box, add the following if it does not already exist:

```
-Xms512m -Xmx512m -XX:CompileThreshold=8000 -XX:PermSize=512m
-XX:MaxPermSize=512m
```

Figure 6–22 Administration Console Configuration Page

Configuration editing is enabled. Future changes will automatically be activated as you modify, add or delete items in this domain.

Domain Structure
rtlog_cluster_domain
Environment
Deployments
Services
Security Realms
Interoperability
Diagnostics

How do I...
Configure startup arguments for Managed Servers
Start Managed Servers from the Administration Console
Shut down a server instance

System Status
Health of Running Servers
Failed (0)
Critical (0)
Overloaded (0)
Warning (0)
OK (4)

Settings for rtlog_ManagedServer_1

Configuration Protocols Logging Debug Monitoring Control Deployments Serv

General Cluster Services Keystores SSL Federation Services Deployment Migra

Save

Node Manager is a WebLogic Server utility that you can use to start, suspend, shut down, and resta machine.

Java Home:

Java Vendor:

BEA Home:

Root Directory:

Class Path:

Arguments:
-Xms256m -Xmx512m -XX:CompileThreshold=8000 -
XX:PermSize=128m -XX:MaxPermSize=256m

If you want to configure the non-default external RTLog configuration directory, include an additional JVM argument:

```
-Drtloggen.config.dir=C:/<rtlog-gen-config_1>/
```

Note: The server-start arguments only work when you are using a NodeManager. If you do not have a NodeManager, specify the JVM argument in the start up scripts. You can also configure the same ext directory location in the RTLog Generator WAR's context-param. For more information, see "Configuration".

9. Start all the managed servers including the Oracle proxy. Figure 6–23 shows an example of the list of managed servers.

Figure 6–23 Administration Console List of Servers

| Name | Type | Cluster | Machine | State | Health | Listen Port |
|--------------------------|------------|-----------------|-----------|---------|--------|-------------|
| loadBalancerProxy | Configured | | Machine_1 | RUNNING | OK | |
| rtlog_AdminServer(admin) | Configured | | Machine_1 | RUNNING | OK | |
| rtlog_ManagedServer_1 | Configured | rtlog_Cluster_1 | Machine_1 | RUNNING | OK | |
| rtlog_ManagedServer_2 | Configured | rtlog_Cluster_1 | Machine_2 | RUNNING | OK | |

Deployment of the RTLog Generator Application on a Cluster

To deploy the application:

1. Oracle proxy creates a web application by creating the web.xml and weblogic.xml files which can be found in the following directory:

```
<WL_HOME>\user_projects\domains\<rtlog_cluster_domain>\apps\OracleProxy4_rtlog_Cluster_1_oracleProxy\WEB-INF
```

You can modify the configurations provided in these two files and redeploy the application from the console by pointing it to this directory, that is, WEB-INF.

2. Navigate to the Administration Console home page and click Deployments in the left navigation menu. Figure 6–24 shows an example of the page before deploying the RTLog Generator application.

Figure 6–24 Administration Console Deployments Page

| Name | State | Health | Type |
|--|--------|--------|------------------|
| OracleProxy4_rtlog_Cluster_1_loadBalancerProxy | Active | OK | Web Application |
| state-management-provider-memory-rar-12.1.3 | Active | OK | Resource Adapter |

3. Click **Install**. The Install Application Assistant page appears. Select the path to the RTLog Generator WAR directory. Select the rtlog-generator.war option. Click **Next**.

Figure 6–25 Administration Console Install Application Assistant Page

Install Application Assistant

Back Next Finish Cancel

Locate deployment to install and prepare for deployment

Select the file path that represents the application root directory, archive file, exploded archive directory, or application module descriptor that you want to deploy.

Note: Only valid file paths are displayed below. If you cannot find your deployment files, upload your file(s) and/or confirm that your application contains the files.

Path:

Recently Used Paths:

Current Location:

- rtlog_cluster_domain
- rtlog_cluster_domain_copy
- rtlog_domain
- rtlog_domain_copy
- rtlogC_domain_old_no_proxy
- rtlog-generator.war**

Back Next Finish Cancel

4. Select only the managed servers and click **Next** to finish the deployment.

Figure 6–26 Install Application Assistant Select Deployment Targets Page

Install Application Assistant

Back Next Finish Cancel

Select deployment targets

Select the servers and/or clusters to which you want to deploy this application. (You can reconfigure the deployment targets after the application is deployed.)

Available targets for rtlog-generator :

Servers

- loadBalancerProxy
- rtlog_AdminServer

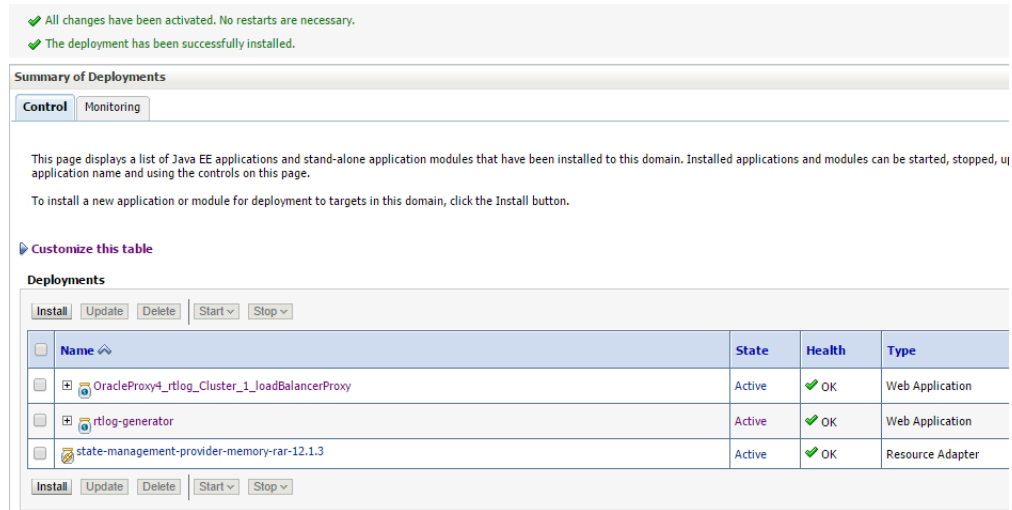
Clusters

- rtlog_Cluster_1**
 - All servers in the cluster
 - Part of the cluster
 - rtlog_ManagedServer_1
 - rtlog_ManagedServer_2

Back Next Finish Cancel

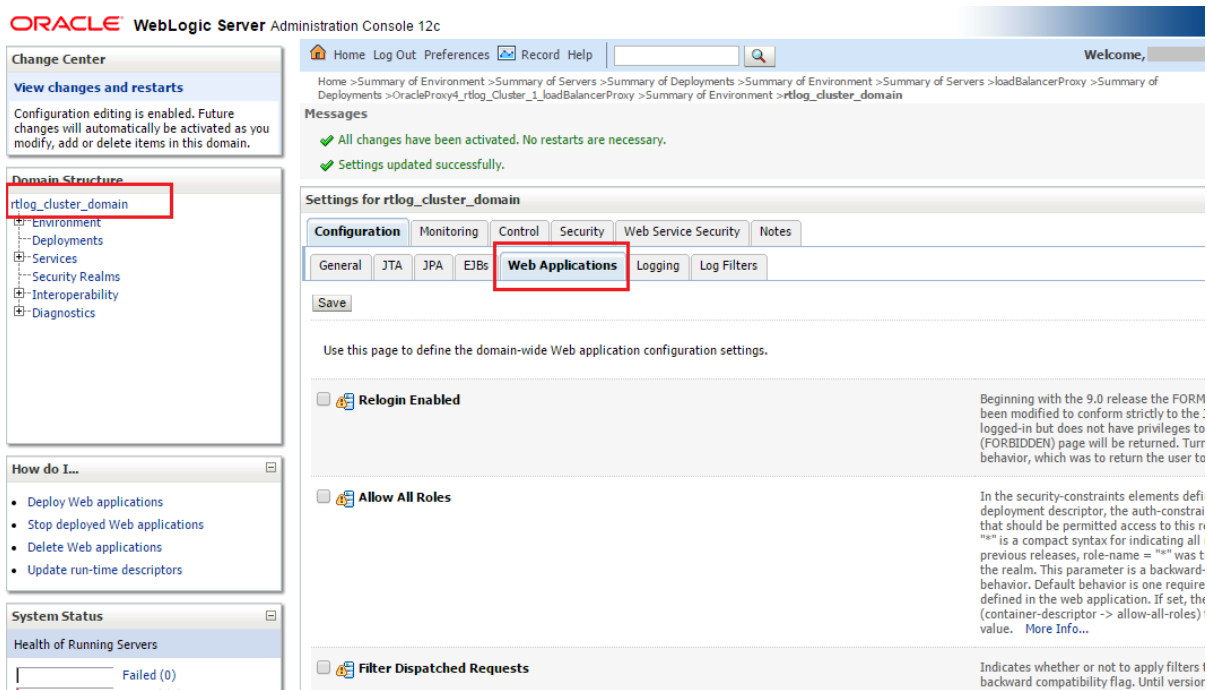
After it is successfully deployed, the RTLog Generator application appears in the Summary of Deployments page.

Figure 6–27 Summary of Deployments Page



5. To enable container and transport level security, see "Security Configuration".
6. To enable the WebLogic Plugin Enabled parameter from the cluster domain:
 - a. Click the `<rtlog_cluster_domain>` link in the left navigation menu. Navigate to the Web Application tab.

Figure 6–28 Administration Console Settings Page



- b. Scroll down the page and select WebLogic Plugin Enabled. Click **Save**.

Figure 6–29 WebLogic Plugin Enabled Parameter

| | | |
|-------------------------------------|----------------------------------|--|
| <input type="checkbox"/> | Error on Name request time value | Global property jsp:param attri to "true", the JS for the "name" for backward cc |
| <input type="checkbox"/> | Client Cert Proxy Enabled | Specifies whett with the reques |
| <input type="checkbox"/> | Http Trace Support Enabled | Returns the val |
| <input checked="" type="checkbox"/> | WebLogic Plugin Enabled | Specifies whett honored. (This Info... |

Security Configuration

The RTLog Generator application is secured by leveraging two levels of security:

- Container level security: Basic HTTP authentication by setting up the security realm in WebLogic. To configure this security, see "[Container Level Security](#)".
- Transport level security: SOAP requests are sent over the secured protocol (HTTPS) by configuring the keystore/truststore in the WebLogic domain and importing the public certificate into Xstore Office's (client) truststore. To configure this security, see "[Transport Level Security](#)".

Container Level Security

The following steps assume that a domain has been created with secure port (HTTPS) enabled. To configure container level security:

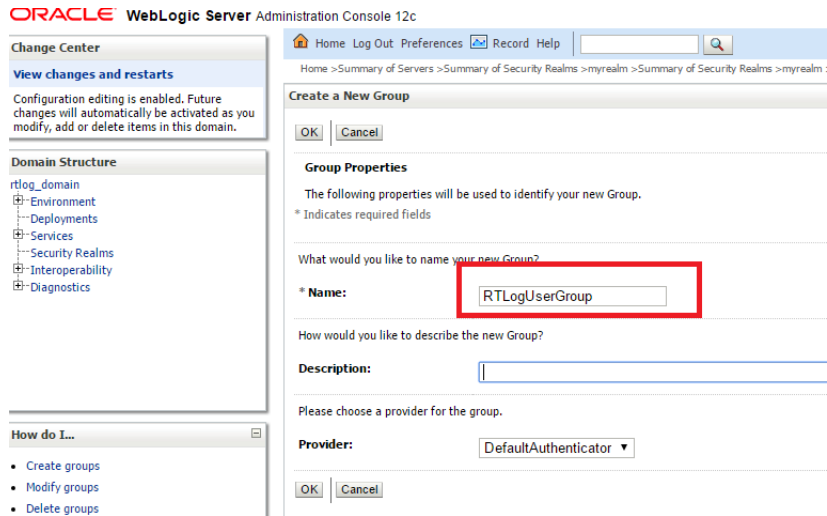
1. Start the WebLogic server and log in to Administration Console.
2. Click Security Realms in the left navigation menu.

Figure 6–30 Administration Console Summary of Security Realms Page

The screenshot shows the Oracle WebLogic Server Administration Console interface. The top navigation bar includes 'Home', 'Log Out', 'Preferences', 'Record', and 'Help'. The breadcrumb trail is 'Home > Summary of Servers > Summary of Security Realms > myrealm > Summary of Security Realms'. The main content area is titled 'Summary of Security Realms' and contains a table of configured security realms. The table has a header row with 'Name' and a red box highlights the 'myrealm' entry. Below the table are 'New' and 'Delete' buttons. The left sidebar shows the 'Domain Structure' with 'Security Realms' selected.

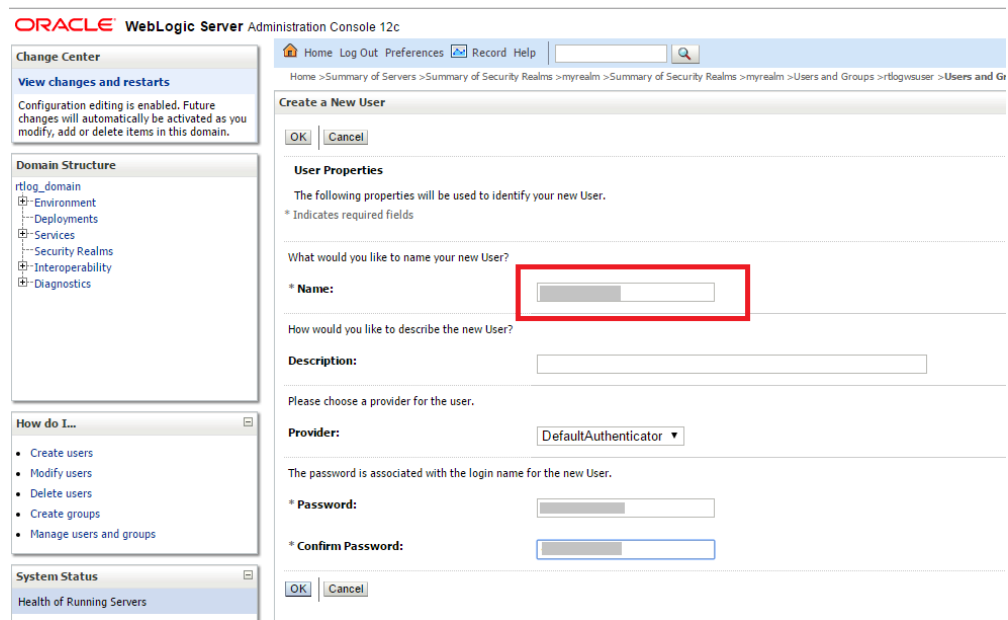
3. In the list of realms on the Summary of Security Realms page, select myrealm.
4. Select Users and Groups and then the Groups tab. To create a new group, click **New**. Enter a group name, for example RTLogUserGroup, and click **OK**.

Figure 6–31 Create a New Group Page



5. Select the Users tab and click **New**. Enter a user name and password and click **OK**.

Figure 6–32 Create a New User Page



6. In the list of users, click the newly created user.

Figure 6–33 Users Page

ORACLE WebLogic Server Administration Console 12c

Home Log Out Preferences Record Help

Home > Summary of Servers > Summary of Security Realms > myrealm > Summary of Security Realms > myrealm > Users and Groups > rtloggenuser > Users

Messages

✔ User created successfully

Settings for myrealm

Configuration Users and Groups Roles and Policies Credential Mappings Providers Migration

Users Groups

This page displays information about each user that has been configured in this security realm.

Customize this table

Users (Filtered - More Columns Exist)

| Name | Description |
|------------------|--|
| OracleSystemUser | Oracle application software system user. |
| rtloggenuser | |
| rtlogws | rtlog webservice password (rtlogws123) |
| rtlogwsuser | |
| ssss | |
| weblogic | This user is the default administrator. |

7. Select the Groups tab. Assign this user to the same group created in Step 4.

Figure 6–34 User Settings Page

ORACLE WebLogic Server Administration Console 12c

Home Log Out Preferences Record Help

Home > Summary of Security Realms > myrealm > Summary of Security Realms > myrealm > Users and Groups > rtloggenuser > Users and Groups > rtloggenuser

Settings for rtloggenuser

General Passwords Attributes Groups

Save

Use this page to configure group membership for this user.

Parent Groups:

Available:

- Administrators
- AppTesters
- CrossDomainConnectors
- Deployers
- Monitors
- Operators
- OracleSystemGroup
- RTLogGenUsers

Chosen:

- RTLogUserGroup

8. Enter the same user name and password created in Step 5 into Xstore Office's broadcaster configuration for the RTLog Generator Web service.

You should try the MrJaxWsPortProxyFactoryBean bean and create the encrypted values for the user name and password using the String Encryption Utility. For more information, see the *Oracle Retail Xstore Point of Service Implementation Guide*.

Figure 6–35 Example of MrJaxWsPortProxyFactoryBean Update

```

<bean id="ReSA_Broadcaster_jaxws_weblogic"
      class="com.micros_retail.xcenter.broadcast.MrJaxWsPortProxyFactoryBean" >

  <property name="endpointAddress" value="https://hostname:7002/rtlog-generator/service" />
  <property name="serviceInterface" value="com.micros_retail.xcenter.poslog.poslogobj.v2.PoslogObjReceiverApi" />
  <property name="wsdlDocumentUrl" value="classpath:wsdl/generic_poslog_object_v2/PoslogObjReceiverApiService.wsdl" />
  <property name="namespaceUri" value="http://v2.ws.poslog.xcenter.dtv/" />
  <property name="serviceName" value="PoslogObjReceiverApiService" />
  <property name="portName" value="PoslogObjReceiverApiPort" />
  <property name="customProperties" ref="jaxwsCustomProperties" />
  <property name="encryptedUsername" value=                />
  <property name="encryptedPassword" value=                />
</bean>

```

Transport Level Security

To configure transport level security:

1. Create keystore.jks using a keytool utility. For information on keytool utilities, see the *Oracle Retail Xstore Point of Service Implementation Guide*.
2. Export the public certificate into a truststore.jks file. These files are needed to configure the custom key and trust store for Step 3.

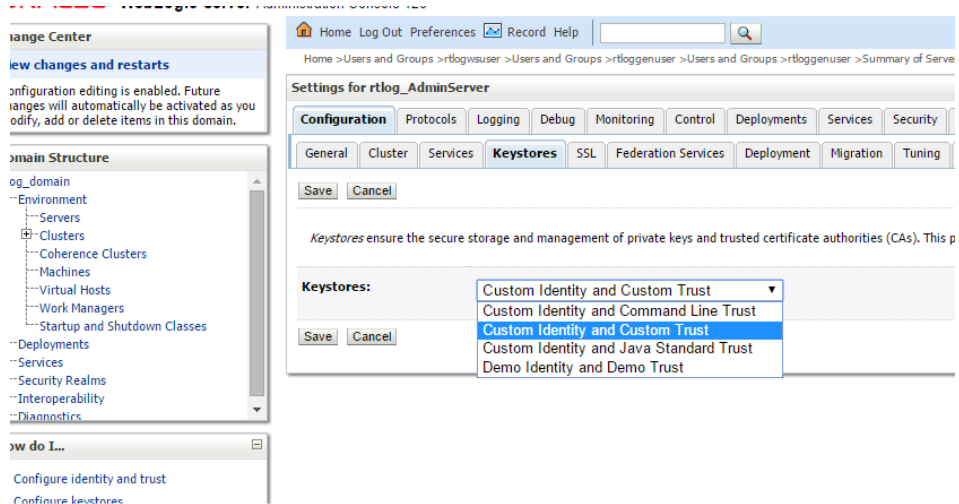
Note: In a clustered environment, import all the public certificates into one truststore file and configure all the instances of the server, including HttpClusterServlet proxy, to use the same truststore file.

3. Log in to the WebLogic console. Click Environment and then the Servers link from the left navigations menu.

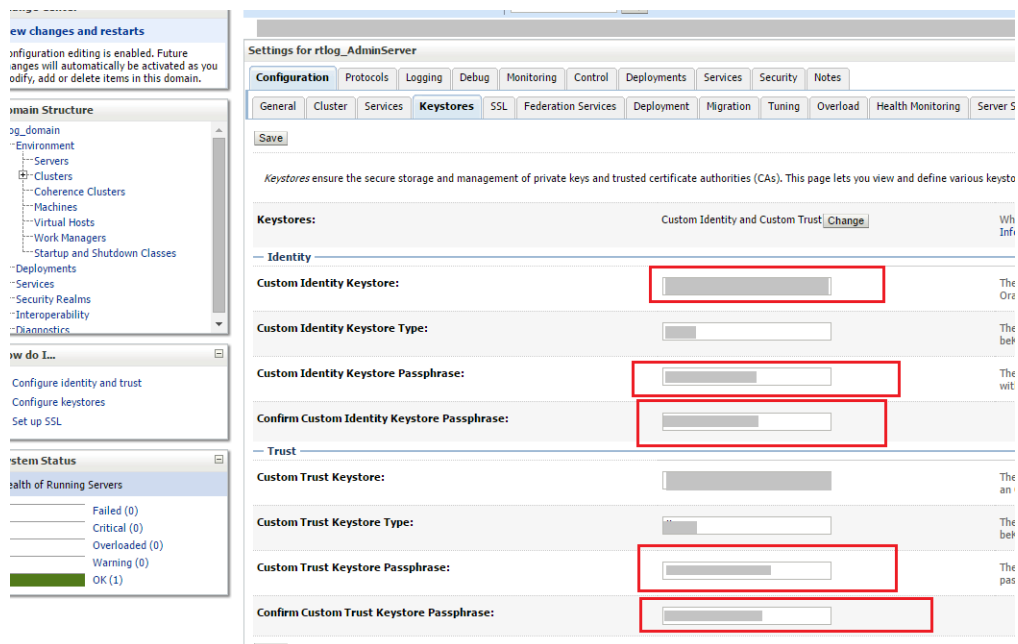
Figure 6–36 Administration Console Servers Page

The screenshot shows the Oracle WebLogic Server Administration Console. The left-hand navigation pane is expanded to 'Environment' > 'Servers'. The main content area is titled 'Summary of Servers' and includes a 'Configuration' tab. Below the title, there is a table of servers. The table has a header row with 'Name' and a dropdown arrow. One server is listed: 'rtlog_AdminServer(admin)'. Above and below the table are buttons for 'New', 'Clone', and 'Delete'.

4. Click **Change**. Select Custom Identity and Custom Trust. Click **Save**.

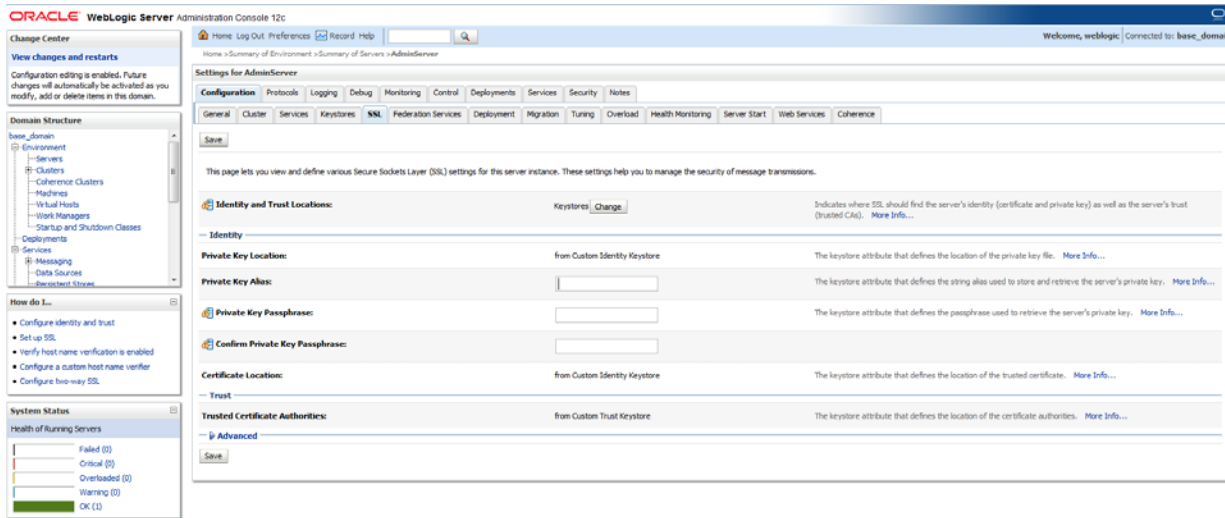
Figure 6–37 Keystores Settings

5. Click the linked name for the Administration Server. The page containing the settings for the Administration Server appears. Select the Keystores tab.

Figure 6–38 Settings for the Administration Server

6. Enter the path to keystore.jks, including the file name, and enter the custom Identity Keystore passphrase you created for the keystore. Repeat this for truststore.jks, but enter the appropriate passphrase for the truststore. For an example, see [Figure 6–38](#).
7. Switch to the SSL tab. Enter the alias name and private keyphrase as created during the certificate generation. To save the changes, click **Save**.

Figure 6–39 Save Settings for Administration Server



Note: For a clustered environment, disable the non-SSL port for the HttpClusterServlet proxy.

Complete the Security Configuration

Test both the container and transport level security using SOAPUI.

To set up the unlimited strength JCE files:

1. Download and install the correct version of the unlimited strength JCE files. For more information, see the *Oracle Retail Xstore Point of Service Implementation Guide*.
2. Configure Oracle WebLogic Server 14c with the Xstore suite of product's supported cipher suites. To configure it, update the `<domain>\<domain_name>\config\config.xml` file and add the following inside the `ssl` block:

```
<ciphersuite>TLS_RSA_WITH_AES_128_GCM_SHA256</ciphersuite>
<ciphersuite>TLS_ECDHE_ECDSA_WITH_AES_256_GCM_SHA384</ciphersuite>
<ciphersuite>TLS_ECDHE_RSA_WITH_AES_256_GCM_SHA384</ciphersuite>
<ciphersuite>TLS_ECDH_ECDSA_WITH_AES_256_GCM_SHA384</ciphersuite>
<ciphersuite>TLS_ECDH_RSA_WITH_AES_256_GCM_SHA384</ciphersuite>
<ciphersuite>TLS_RSA_WITH_AES_256_GCM_SHA384</ciphersuite>
```

3. Disable the schema validation in WebLogic by passing the JVM argument in the WebLogic startup script:
 - Dweblogic.configuration.schemaValidationEnabled=false
4. Xstore Office's RTLog Generator broadcaster end point should be configured to use the secured (HTTPS) URL for configuring the container level security section:

```
<property name="endpointAddress"
value="https://<hostname>:7002/rtlog-generator/service" />
```

The `endpointAddress` property is defined at `xcenter-spring-beans.xml` under Xcenter external configuration directory `\xcenter-config`. There are two required modifications:

- Modify `broadcasterManager` bean in the file by uncommenting the line below.


```
<ref bean="ReSA_Broadcaster" />
```

- Configure endpointAddress of the ReSA_Broadcaster_jaxws bean.

RTLog Generator Cloud

This chapter describes the RTLog Generator component of Xstore Office Cloud. The RTLog Generator Cloud application can be used with both the cloud or on premise Merchanding/Pricing applications.

RTLog Generator Cloud

This chapter describes how configure the RTLog Generator application deployed on cloud.

The RTLog Generator on cloud is a Java and XML based web application that exposes a Spring-JAXWS implemented SOAP web service and JAXRS implemented REST web services. It is usually deployed alongside the other Xstore Office Cloud applications.

Configuration

The RTLog Generator cloud application can be configured in the following way.

Customize the RTLog Generator's mapping and format configuration via REST services.

Note: For more information on how to customize the RTLog Generator, see the [Configuration](#) section in [Chapter 6, "RTLog Generator On-Premise"](#) and the *Retail Xstore - RTLog Generator Extension Guidelines (Doc ID 2174095.1)* on <https://support.oracle.com>.

Integration

This section describes the RTLog Generator Cloud integration.

Updating Mapping Configuration

RTLog Generator Cloud application provides REST services to retrieve, update and delete the `RTLogMappingConfig.xml` file. All the three services point to the URL at `https://<hostname>/rtlog-generator/rest/config/file/v1/RTLogMappingConfig`

If RTLog generator is deployed on cloud, its mapping configuration file `RTLogMappingConfig.xml` is not accessible to a user. To customize the mapping, restful APIs are provided to upload a customized `RTLogMappingConfig.xml` to override the default out-of-box one.

Table 7-1 REST Services related to the RTLogMappingConfig.xml

| HTTP Protocol | Security Protocol | Response Type | Description |
|---------------|-------------------|------------------|--|
| GET | OAuth2 | application/xml | Returns the active RTLogMappingConfig.xml file. If the customer has not uploaded a customized configuration xml file yet, provides a copy of the default mapping configuration XML file that is provided with the deployment. |
| PUT | OAuth2 | application/json | Customer submits the updated RTLogMappingConfig.xml file as the request body. Returns JSON that contains the number of bytes in the uploaded XML file. |
| DELETE | OAuth2 | No content | If the customer has uploaded a configuration XML file previously, it will be deleted and HTTP 200 status is returned. If there is no customized RTLogMappingConfig.xml file active yet, HTTP 204 status is returned. The default RTLogMappingConfig.xml that is part of the deployment will resume being the active mapping configuration. |

The examples below show how to retrieve and update the RTLogMappingConfig.xml.

Example 7-1 Get active RTLogMappingConfig.xml - Get Current RTLog Mapping Configuration

```
$ curl -H "Authorization: Bearer <token>"
https://<rlog-generator-host>/rtlog-generator/rest/config/file/v1/RTLogMappingConfig" > RTLogMappingConfig.xml
```

Example 7-2 Update RTLogMappingConfig.xml - Update the RTLog Mapping Configuration

```
$ curl -H "Authorization: Bearer <token>" -X PUT -T "/path/to/mapping/file"
https://<rlog-generator-host>/rtlog-generator/rest/config/file/v1/RTLogMappingConfig" > RTLogMappingConfig.xml
```

Similar to the example above, using the -X option with the value of DELETE will delete any customer uploaded mapping configuration XML file.

Updating Format Configuration

RTLog Generator Cloud application provides REST services to retrieve, update and delete the RTLogFormatConfig.xml file. All the three services point to the URL at:

<https://<hostname>/rtlog-generator/rest/config/file/v1/RTLogFormatConfig>

If RTLog generator is deployed on cloud, its format configuration file RTLogFormatConfig.xml is not accessible to a user. A format configuration file defines the Sales Audit RTLog format, and in most cases is not up for customizations. However RTLogFormatConfig.xml file can be customized when:

- Customization has been done to Sales Audit to accept an enhanced RTLog format with additional fields
- A retailer has to integrate with an earlier version of Sales Audit which is not supported out of box, and it accepts a slightly different older RTLog format.

Table 7-2 REST Services related to the RTLogFormatConfig.xml

| HTTP Protocol | Security Protocol | Response Type | Description |
|---------------|-------------------|------------------|---|
| GET | OAuth2 | application/xml | Returns the active RTLogFormatConfig.xml file. If the customer has not uploaded a customized configuration xml file yet, provides a copy of the default mapping configuration XML file that is provided with the deployment. |
| PUT | OAuth2 | application/json | Customer submits the updated RTLogFormatConfig.xml file as the request body. Returns JSON that contains the number of bytes in the uploaded XML file. |
| DELETE | OAuth2 | No content | If the customer has uploaded a configuration XML file previously, it will be deleted and HTTP 200 status is returned. If there is no customized RTLogFormatConfig.xml file active yet, HTTP 204 status is returned. The default RTLogMappingConfig.xml that is part of the deployment will resume being the active mapping configuration. |

The examples below show how to retrieve and update the RTLogFormatConfig.xml.

Example 7-3 Get active RTLogFormatConfig.xml - Get Current RTLog Format Configuration

```
$ curl -H "Authorization: Bearer <token>"
https://<rlog-generator-host>/rtlog-generator/rest/config/file/v1/RTLogFormatConfig" > RTLogFormatConfig.xml
```

Example 7-4 Update RTLogFormatConfig.xml - Update the RTLog Format Configuration

```
$ curl -H "Authorization: Bearer <token>" -X PUT -T "/path/to/format/file"
https://<rlog-generator-host>/rtlog-generator/rest/config/file/v1/RTLogFormatConfig"
```

Similar to the example above, using the -X option with the value of DELETE will delete any customer uploaded format configuration XML file.

Retrieving Published RTLog Files

RTLog Generator Cloud application provides three mechanisms to retrieve the published RTLog files.

- FTS (File Transfer Service) provides a way to deliver RTLog files to Sales Audit deployed as part of RMFCS v21.0 and later. By default it delivers RTLog files in compressed format (zip files). A deployment option is provided to turn off the compression. It puts RTLog files to object storage for Sales Audit to import. This service is not available for Sales Audit deployed in RMFCS v19.x.

A service request has to be filed by a retailer to setup RTLog Generator Cloud to use FTS.

FTS_ENDPOINTURL is made available to the RTLog Generator Cloud deployment team by the Merchandising team.

- SFTP service provides a way to transfer RTLog files. This method is commonly used to post RTLog files to Sales Audit deployed on RMFCS v19. By default it delivers RTLog files in compressed format (zip files). A deployment option is provided to turn off the compression.

A service request has to be filed by a retailer to setup RTLog Generator Cloud to use SFTP.

- SFTP host and path are made available to the RTLog Generator Cloud deployment team by the retailer.
- SFTP connectivity utilizes public/private key based authentication. Once the key pair is generated, the public key has to be added to the SFTP server. If the SFTP server is hosted by Merchandising Cloud, the public key will be accessible to Merchandising Cloud deployment team to add to the SFTP server; if the SFTP server is hosted on premise, the public key will be handed over to the retailer to add to the SFTP server.

- REST service provides a way to download RTLog files in compressed format (zip files). This is the default delivery method. The REST service endpoint URL is

`https://<hostname>/rtlog-generator/rest/rtlog/files/v1/published`

Table 7–3 REST Services to download RTLog Files

| HTTP Protocol | Security Protocol | Response Type | Description |
|---------------|-------------------|--------------------------|--|
| GET | OAuth2 | application/octet-stream | Returns the oldest RTLog zip file stream, if available. The content-disposition response header contains the name of the attached zip file. If no zip file is available, a HTTP 204 no content is returned. |

Example 7–5 Get Published RTLogs - OAuth2 token request

```
$ curl -O -J -H "Authorization: Bearer <token>"
"https://<rtlog-generator-host>/rtlog-generator/rest/rtlog/files/v1/published"
```

It is recommended to have a programmable approach to acquire the OAuth2 token and utilizing the token to download the available published RTLog files compared to the command line tools shown as examples above.

Security Configuration

RTLog Generator's web services are secured by requiring HTTPS protocol for transport layer security and require OAuth2 authentication for application level security. All of the Xoffice applications on cloud including the RTLog Generator have a valid OAuth Client (Application) registered with a specific tenant of the Oracle Identity Cloud Service.

OAuth2 authentication is a two-step process.

- Acquire a valid OAuth2 Bearer token using the IDCS Client Credentials.
- Provide the token value in the HTTP Authorization header for all of the web service requests until the token's validity is expired.

Acquiring IDCS Token

In order to acquire a valid IDCS token, the following information is needed beforehand.

- IDCS tenant host information to build the URL for requesting a token
 - `https://<IDCS_TENANT_HOST>/oauth2/v1/token`
- `ClientID` and `ClientSecret` for the RTLog Generator Client App (that is Sales Audit).
- A command line utility or any software that can make HTTP requests with the ability to setup specific header values
 - "curl" in Linux environments
- Access to a command/utility to encode the credentials in base64 format.
 - "base64" command in Linux environments
 - "certutil" command in Windows environments

The following example shows how to request a token using the curl command line tool in a Linux environment. Ensure to replace the `clientID`, `clientSecret` and `IDCS_TENANT_HOST` with the appropriate values.

Example 7-6 Request IDCS OAuth2 Token - OAuth2 Token Request

```
$ curl -i -H "Authorization: Basic $(echo -n clientID:clientSecret | base64)" -H  
"Content-Type: application/x-www-form-urlencoded;charset=UTF-8" https://<IDCS_  
TENANT_HOST>/oauth2/v1/token -d "grant_type=client_credentials&scope=urn:opc:ldm:  
_myscopes_"
```

You may generate Base64 encoded text of the "`clientID:clientSecret`" ahead of the request and use it directly in the curl command for the Basic Authorization header value. The following example shows the response that contains the token.

Example 7-7 IDCS OAuth2 Token Response - OAuth2 Token Response

```
{"access_token": "<oauth2_token>",  
"token_type": "Bearer",  
"expires_in": 3600 }
```

The response above shows the token value and the expiration time in seconds. Usually, the token is a sequence of random characters of varying length up to a maximum of 16K.

Provide IDCS Authentication

The following example shows how to provide the OAuth2 token while communicating with RTLog Generator REST services. The following example shows how to request the current active `RTLogMappingConfig.xml` file. Please make sure to replace the "`<token>`" with a valid OAuth2 token acquired in the last step and provide the correct RTLog Generator Host value.

Example 7-8 Provide OAuth2 Token - Provide OAuth2 Token for REST Services

```
$ curl -i -H "Authorization: Bearer <token>"  
"https://<rlog-generator-host>/rtlog-generator/rest/config/file/v1/RTLogMappingCon  
fig"
```

Appendix: Xstore to Sales Audit Mapping Details

The mapping from the Xstore POSLog format to the Sales Audit RTLog format is defined in the Xstore configuration file `RTLogMappingConfig.xml`. This appendix provides details on the following code types used by each of the applications that are used in this mapping:

- [Transaction Types](#)
- [Tender Types](#)
- [Tender Totals](#)
- [Item Types](#)
- [Sales Audit Reason Codes](#)
- [Item Status/and Sales Types](#)
- [Customer ID Types](#)
- [Sales Audit Tax Codes](#)
- [Reference Codes/Labels](#)
- [Sale Return Transaction](#)

Transaction Types

Both Xstore and Sales Audit support a number of similar transactions types, but each solution uses different codes for these transaction types. The table below shows how the transaction types in Xstore map to the transaction types and sub-transaction types used in Sales Audit. See also the *Oracle Retail Sales Audit Implementation Guide* for more details on these codes.

Table A-1 Transaction Type Mapping

| Xstore Transaction Type | Sales Audit Transaction Type TRAT | Sales Audit Sub-Transaction Type TRAS | Description |
|--------------------------------|--|--|---|
| ACCOUNT_LOOKUP | OTHER | OTHER | ACCOUNT_LOOKUP transactions are passed from Xstore to Sales Audit for full visibility audit, but not otherwise implemented in Sales Audit. |
| BALANCE_INQUIRY | OTHER | OTHER | BALANCE_INQUIRY transactions are passed from Xstore to Sales Audit for full visibility audit, but not otherwise implemented in Sales Audit. |
| CREDIT_APPLICATION | OTHER | OTHER | CREDIT_APPLICATION transactions are passed from Xstore to Sales Audit for full visibility audit, but not otherwise implemented in Sales Audit. |
| ESCROW | OTHER | OTHER | ESCROW transactions are passed from Xstore to Sales Audit for full visibility audit, but not otherwise implemented in Sales Audit. |
| EXCHANGE_RATE | OTHER | OTHER | EXCHANGE_RATE transactions are passed from Xstore to Sales Audit for full visibility audit, but not otherwise implemented in Sales Audit. |
| GNRIC | OTHER | OTHER | GNRIC transactions are passed from Xstore to Sales Audit for full visibility audit, but not otherwise implemented in Sales Audit. |
| INVENTORY_CONTROL | OTHER | OTHER | INVENTORY_CONTROL transactions are mapped from Xstore to Sales Audit for full visibility audit, but not otherwise implemented in Sales Audit. Xstore should be configured so that inventory control transactions are not generated, and therefore not sent to Sales Audit. |
| INVENTORY_SUMMARY_COUNT | OTHER | OTHER | INVENTORY_SUMMARY_COUNT transactions are mapped from Xstore to Sales Audit for full visibility audit, but not otherwise implemented in Sales Audit. Xstore should be configured so that inventory summary count transactions are not generated, and therefore not sent to Sales Audit. |
| MOVEMENT_PENDING | OTHER | OTHER | MOVEMENT_PENDING transactions are mapped from Xstore to Sales Audit for full visibility audit, but not otherwise implemented in Sales Audit. Xstore should be configured so that inventory summary count transactions are not generated, and therefore not sent to Sales Audit. |
| NO_SALE | NOSALE | NOSALE | NA |
| POST_VOID | PVOID | VOID | NA |

Table A-1 (Cont.) Transaction Type Mapping

| Xstore Transaction Type | Sales Audit Transaction Type TRAT | Sales Audit Sub-Transaction Type TRAS | Description |
|---|-----------------------------------|---------------------------------------|---|
| RETAIL_SALE (can be mapped to multiple Sales Audit transaction types depending on other conditions) | SALE | SALE | Regular transaction. |
| | NOSALE | SUSPND | Suspend transaction. |
| | VOID | CANCEL | Cancel transaction. |
| | VOID | CANCEL | Cancel orphaned transaction. |
| SESSION_CONTROL | OTHER | OTHER | Issue till. |
| | OTHER | OTHER | Assign till/assign till tender transfer. |
| | OTHER | OTHER | Attach till. |
| | OTHER | OTHER | Remove till. |
| | OTHER | OTHER | Return till. |
| SYSTEM_CLOSE | CLOSE | CSTORE | Close store. |
| SYSTEM_OPEN | OPEN | OSTORE | Open store. |
| TENDER_CONTROL (can be mapped to multiple Sales Audit transaction types depending on other conditions) | OPEN | OTILL | Begin till count. |
| | CLOSE with TOTAL /OTHER | CTILL with CTILLT /OTHER | Till closing count (register accountability/till accountability). |
| | CLOSE and TOTAL | CTILL and CTILLT | Till reconcile. Each counted tender type has a corresponding TOTAL and CTILLT as a THEAD. |
| | PAIDIN | PITILL | Pay in. |
| | PAIDOU | POTILL | Pay out. |
| | OTHER | AUDIT | Till audit. |
| | PULL | PUTILL | Mid-day deposit. Place funds in store bank. |
| | OTHER | BANK | Bank deposit. |
| | LOAN | LOTILL | Till loan (cash transfer). |
| | PULL | PUTILL | Pick up till (cash pickup). |
| | OTHER | OTHER | Open store bank. |
| OTHER | OTHER | Store bank reconcile. | |
| TENDER_EXCHANGE | PAIDIN | PITILL | NA |
| TILL_CONTROL | OTHER | OTHER | NA |
| TIMECLOCK | OTHER | OTHER | Employee clock in. |
| | OTHER | OTHER | Employee clock out. |
| TRAINING_MODE_ENTRY | OTHER | NTRAIN | NA |
| TRAINING_MODE_EXIT | OTHER | XTRAIN | NA |
| WORKSTATION_CLOSE | CLOSE | CREG | NA |

Table A-1 (Cont.) Transaction Type Mapping

| Xstore Transaction Type | Sales Audit Transaction Type TRAT | Sales Audit Sub-Transaction Type TRAS | Description |
|-----------------------------------|--|--|--|
| WORKSTATION_COMPLETE_REMOTE_CLOSE | CLOSE | CRGRC | NA |
| WORKSTATION_OPEN | OPEN | OREG | NA |
| WORKSTATION_START_REMOTE_CLOSE | OTHER | CRGRC | NA |
| GIFT_REGISTRY | OTHER | OTHER | Assign gift registry (register operation) |
| | OTHER | OTHER | Reissue gift registry (register operation) |
| RAIN_CHECK | OTHER | OTHER | Redeem rain check. |
| BATCH_CLOSE | OTHER | OTHER | Credit and debit settlement. |
| REOPEN | REOPEN | NA | Used to reopen a previously closed store day in Sales Audit. |

Tender Types

In order to communicate the tender type used on transactions, a specific mapping is used between Xstore and Sales Audit. The details below outline how the tenders in Xstore map to the Sales Audit tender groups and types. See also the *Oracle Retail Sales Audit Implementation Guide* for information on configuration of tender types and tender type groups.

Table A-2 Tender Type Mapping

| Xstore | | Xstore POS Log Tender Group Type | | Sales Audit RTLog | |
|-------------------------|----------------------|---|------------------|--------------------------|-------------------------------------|
| Tender Type Code | Tender Type D | Tender Type | Tender ID | Tender Type Group | Tender Type ID |
| CURRENCY | USD_CURRENCY | Cash | USD_CURRENCY | CASH | If primary 1000, if alternate 1010. |
| | AUD_CURRENCY | Cash | AUD_CURRENCY | CASH | If primary 1000, if alternate 1010. |
| | CAD_CURRENCY | Cash | CAD_CURRENCY | CASH | If primary 1000, if alternate 1010. |
| | EUR_CURRENCY | Cash | EUR_CURRENCY | CASH | If primary 1000, if alternate 1010. |
| | GBP_CURRENCY | Cash | GBP_CURRENCY | CASH | If primary 1000, if alternate 1010. |

Table A-2 (Cont.) Tender Type Mapping

| Xstore | | Xstore POS Log Tender Group Type | | Sales Audit RTLog | |
|------------------|---------------------------|----------------------------------|---------------------------|-------------------|-----------------------------------|
| Tender Type Code | Tender Type D | Tender Type | Tender ID | Tender Type Group | Tender Type ID |
| CREDIT_CARD | VISA | CreditDebit | VISA | CCARD | 3000 |
| | MASTERCARD | CreditDebit | MASTERCARD | CCARD | 3010 |
| | AMERICAN_EXPRESS | CreditDebit | AMERICAN_EXPRESS | CCARD | 3020 |
| | DINERS_CLUB | CreditDebit | DINERS_CLUB | CCARD | 3040 |
| | DISCOVER | CreditDebit | DISCOVER | CCARD | 3030 |
| | JCB | CreditDebit | JCB | CCARD | 3090 |
| | DEBITCARD | CreditDebit | DEBITCARD | DCARD | 8000 |
| ACCOUNT | HOUSE_ACCOUNT | dtv:Account | HOUSE_ACCOUNT | CCARD | 3120 |
| | A new type of credit card | CreditDebit | A new type of credit card | CCARD | Map to UNKNW. |
| CHECK | CHECK | Check | CHECK | CHECK | If primary 2000, if foreign 2050. |
| TRAVELERS_CHECK | USD_TRAVELERS_CHECK | dtv:TravelersCheck | USD_TRAVELERS_CHECK | CHECK | If primary 2020, if foreign 2060. |
| | CAD_TRAVELERS_CHECK | dtv:TravelersCheck | CAD_TRAVELERS_CHECK | CHECK | If primary 2020, if foreign 2060. |

Table A-2 (Cont.) Tender Type Mapping

| Xstore | | Xstore POS Log Tender Group Type | | Sales Audit RTLog | |
|-------------------|--------------------------------|----------------------------------|--------------------------------|--|-----------------------------------|
| Tender Type Code | Tender Type D | Tender Type | Tender ID | Tender Type Group | Tender Type ID |
| VOUCHER | GIFT_CERTIFICATE | Voucher | GIFT_CERTIFICATE | VOUCH | If primary 4030, if foreign 4100. |
| | ISSUE_GIFT_CERTIFICATE | Voucher | ISSUE_GIFT_CERTIFICATE | VOUCH | If primary 4030, if foreign 4100. |
| | ISSUE_MERCHANDISE_CREDIT_CARD | Voucher | ISSUE_MERCHANDISE_CREDIT_CARD | VOUCH | 4050 |
| | ISSUE_STORE_CREDIT | Voucher | ISSUE_STORE_CREDIT | VOUCH | 4050 |
| | ISSUE_XPAY_GIFT_CARD | Voucher | ISSUE_XPAY_GIFT_CARD | VOUCH | 4040 |
| | MALL_CERTIFICATE | Voucher | MALL_CERTIFICATE | VOUCH | 4060 |
| | MERCHANDISE_CREDIT_CARD | Voucher | MERCHANDISE_CREDIT_CARD | VOUCH | 4050 |
| | RELOAD_MERCHANDISE_CREDIT_CARD | Voucher | RELOAD_MERCHANDISE_CREDIT_CARD | VOUCH | 4050 |
| | RELOAD_XPAY_GIFT_CARD | Voucher | RELOAD_XPAY_GIFT_CARD | VOUCH | 4040 |
| | STORE_CREDIT | Voucher | STORE_CREDIT | VOUCH | If primary 4050, if foreign 4090. |
| | XPAY_GIFT_CARD | Voucher | XPAY_GIFT_CARD | VOUCH | 4040 |
| COUPON | COUPON | Manufacturer Coupon | COUPON | QPON | 5000 |
| | ROOM_CHARGE | CreditDebit | ROOM_CHARGE | VOUCH | 4050 |
| CREDIT_CARD | PAYPAL | TBD | PAYPAL | PAYPAL | 3075 |
| HOME_OFFICE_CHECK | HOME_OFFICE_CHECK | NA | NA | Not supported in this solution. Home office check tenders should not be used in Xstore if it is integrated with Sales Audit. | |

Tender Totals

Sales Audit uses tender totals to compare the total amount of a tender reported from the store (Accounted For) with the amount that it should have reported (Accountable For) based on store activity in a day. Totals are fully customer configured in Sales Audit. However, when integrating with Xstore, a few specific totals are expected to be created to total tenders. The totals that should be created and their ID are outlined in the table below, along with the Xstore tender type used for the total.

Table A-3 Total Tender ID Mapping

| Xstore | | Sales Audit RTLog |
|-------------------|--------------------------------|--------------------------|
| TenderType | TenderID | Total ID |
| CURRENCY | USD_CURRENCY | CASH |
| | AUD_CURRENCY | CASHAC |
| | CAD_CURRENCY | CASHAC |
| | EUR_CURRENCY | CASHAC |
| | GBP_CURRENCY | CASHAC |
| TRAVELERS_CHECK | USD_TRAVELERS_CHECK | TCHECK |
| | AUD_TRAVELERS_CHECK | TCHECKAC |
| | CAD_TRAVELERS_CHECK | TCHECKAC |
| | EUR_TRAVELERS_CHECK | TCHECKAC |
| | GBP_TRAVELERS_CHECK | TCHECKAC |
| | MXN_TRAVELERS_CHECK | TCHECKAC |
| CREDIT_CARD | CREDIT_CARD | CCARD |
| VOUCHER | GIFT_CERTIFICATE | GIFTCERT |
| | MALL_CERTIFICATE | MALLCERT |
| | MERCHANDISE_CREDIT_CARD | MCCARD |
| | RELOAD_MERCHANDISE_CREDIT_CARD | RMCCARD |
| | RELOAD_XPAY_GIFT_CARD | RXPAYGC |
| | STORE_CREDIT | STCRDT |
| | XPAY_GIFT_CARD | XPAYGC |
| | ISSUE_XPAY_GIFT_CARD | IXPAYGC |
| | ISSUE_STORE_CREDIT | ISTCRDT |
| | ISSUE_MERCHANDISE_CREDIT_CARD | IMCCARD |
| ACCOUNT | HOUSE_ACCOUNT | HACCNT |
| COUPON | COUPON | COUPON |

Item Types

There are a number of different item types that are used in Xstore, and these all need to be mapped to item types used by Sales Audit. The table below outlines how the types in Xstore map to the Sales Audit item types.

Table A-4 Item Type Mapping

| Xstore Item Type | Sales Audit Item Type | Description |
|-------------------------|------------------------------|----------------------|
| Alteration | NMITEM | Non-Merchandise Item |
| Deposit | NMITEM | Non-Merchandise Item |
| dtv:GiftCertificate | GCN | Voucher |
| dtv:NonMerchandise | NMITEM | Non-Merchandise Item |

Table A-4 (Cont.) Item Type Mapping

| Xstore Item Type | Sales Audit Item Type | Description |
|-------------------------|------------------------------|----------------------|
| dtv:Payment | NMITEM | Non-Merchandise Item |
| Fee | NMITEM | Non-Merchandise Item |
| ItemCollection | ITEM | Item |
| Service | NMITEM | Non-Merchandise Item |
| Stock | ITEM | Item |
| Warranty | NMITEM | Non-Merchandise Item |

Sales Audit Reason Codes

Xstore has a single set of reason codes, used both for reason codes, price override codes, and other modifications. Sales Audit separates these concepts into individual sets. Because reason codes can be mixed coming out of Xstore, Sales Audit has mapped some code values to multiple code types to avoid the possibility of errors.

For more information on each of these categories of reason codes, see the *Oracle Retail Sales Audit Implementation Guide*.

Reason Codes

First is a general reason code that may be entered for specific transaction types to provide more information about the context of the transaction. This type of reason code is mapped to Xstore miscellaneous reason codes.

Table A-5 Sales Audit Reason Codes

| Xstore Reason Code | Sales Audit Reason Code | Description |
|---------------------------|--------------------------------|------------------------|
| PV1 | PV1 | Cashier Error |
| PV2 | PV2 | Supervisors Discretion |
| PV3 | PV3 | Customer Satisfaction |
| NS1 | NS1 | Making Change |
| NS2 | NS2 | Employee Check Cashed |
| NS3 | NS3 | Petty Cash In |
| NS4 | NS4 | Petty Cash Out |
| NS5 | NS5 | Spiff/Bonus Out 1 |
| CF1 | CF1 | Holiday Adjustment |
| CF2 | CF2 | Register Down |
| PAID_IN | PI1 | Change from Paid Out |
| PAID_IN | PI2 | Found Money |
| PAID_IN | PI3 | Drawer Loan 1 |
| PAID_IN | TENDEX | Tender exchange |
| PAID_OUT | PO1 | Stocks |
| PAID_OUT | PO2 | Delivery |

Table A-5 (Cont.) Sales Audit Reason Codes

| Xstore Reason Code | Sales Audit Reason Code | Description |
|---------------------------|--------------------------------|---------------------|
| PAID_OUT | PO3 | Postage |
| PAID_OUT | PO4 | Contractor Services |
| PAID_OUT | PO5 | Store Incentives |

Return Reason Codes

Sales Audit return reason codes are used to provide the context of the return in Sales Audit. These map to the Xstore reason codes as follows:

Table A-6 Return Reason Codes

| Xstore Reason Code | Sales Audit Reason Code | Description |
|---------------------------|--------------------------------|-----------------------------|
| RET1 | RET1 | Did not like |
| RET2 | RET2 | Better price somewhere else |
| RET3 | RET3 | Did not fit |
| RET4 | RET4 | Damaged |
| RET5 | RET5 | Exchange |
| RET6 | RET6 | Poor quality |
| RET41 | RET41 | Open box |
| RET42 | RET42 | Unusable |
| RET43 | RET43 | Repairable |

Discount Reason Codes

Used in Sales Audit to indicate the valid discount types for sales and return transaction, this table shows how these are mapped to the Xstore reason codes used for returns:

Table A-7 Discount Reason Codes

| Xstore Reason Code | Sales Audit Reason Code | Description |
|---------------------------|--------------------------------|---------------------|
| DC1 | S | Incorrect Label |
| DC2 | MS | Manager Discretion |
| DC3 | CP | Price Guarantee |
| DC4 | D | Damage Adjustment |
| NEW_PRICE_RULE | NEWPRC | New Price Rule |
| DOCUMENT | DOC | Document |
| MANUFACTURER_COUPON | MCOUP | Manufacturer Coupon |
| REFUND_PRORATION | REFUND | Refund Proration |
| CALCULATED_WARRANTY_PRICE | CALWAR | Warranty Price |

Item Price Override Reason Codes

This grouping of reason codes is used to hold the valid price override reason codes that are expected by Sales Audit. These map to the reason codes used in Xstore as follows:

Table A-8 Item Price Override Reason Codes

| Xstore Reason Code | Sales Audit Reason Code | Description |
|---------------------|-------------------------|-------------------------|
| AR_PR_1 | AR_PR_1 | Insufficient Funds |
| AR_PR_2 | AR_PR_2 | Wrong Amount |
| AR_PR_3 | AR_PR_3 | Wrong Amount |
| AR_PR_4 | AR_PR_4 | Wrong Invoice |
| COMMENT | NEWPRC | Other - Enter Comments |
| PC1 | S | Incorrect Label |
| PC2 | MS | Supervisors Discretion |
| PC3 | CP | Competitive Price Match |
| PC4 | D | Damage Adjustment |
| BASE_PRICE_RULE | BSPRC | Base Price Rule |
| PROMPT_PRICE_CHANGE | PROMPT | Price Prompt |
| AUTHORIZED_AMOUNT | AUTHMT | Authorized Amount |

Item Status/and Sales Types

Valid values for item status in Sales Audit are:

| | |
|-----|------------------|
| V | Voided |
| S | Sale |
| R | Return |
| O | Other |
| ORI | Order Initiate |
| ORC | Order Cancel |
| ORD | Order Complete |
| LIN | Layaway Initiate |
| LCA | Layaway Cancel |
| LCO | Layaway Complete |

Valid values for sales type in Sales Audit are:

| | |
|---|-------------------------|
| R | Regular |
| I | In-Store Customer Order |
| E | External Customer Order |

These two sets of codes are then mapped to the actions in Xstore as follows:

Table A-9 Item Status/and Sales Type Mapping

| Xstore Item | Xstore Action | Sales Audit Item Status | Sales Audit Sales Type |
|--------------------|----------------------------|--------------------------------|-------------------------------|
| Regular Sale | Sale | S | R |
| | Return | R | R |
| | Void | S and V (two lines) | R |
| Layaway Item | Init | LIN | I |
| | Cancel | LCA | I |
| | Pickup | LCO | I |
| | Void | S and V (two lines) | I |
| Locate Order | Init | ORI | E |
| | Cancel | ORC | E |
| | Pickup | ORD | E |
| | Void when update or pickup | ORC | E |
| | Void when Init | S and V (two lines) | E |
| Special Order | Init | ORI | E |
| | Cancel | ORC | E |
| | Pickup | ORD | E |
| | Void when update or pickup | ORC | E |
| | Void when Init | S and V (two lines) | E |
| Work Order | Init | ORI | I |
| | Cancel | ORC | I |
| | Pickup | ORD | I |
| | Void when update or pickup | ORC | I |
| | Void when Init | S and V (two lines) | I |
| Pre-Sale | Init | ORI | I |
| | Cancel | ORC | E |
| | Pickup | ORD | E |
| | Void when update or pickup | ORC | E |
| | Void when Init | S and V (two lines) | E |
| On Hold | Init | ORI | I |
| | Cancel | ORC | I |
| | Pickup | ORD | I |
| | Void when update or pickup | ORC | I |
| | Void when Init | S and V (two lines) | I |
| Send Sale | Init | S | R |
| | Void when Init | S and V (two lines) | R |

Customer ID Types

Sales Audit has a set of codes that it uses for validating the various forms of identification that can be presented at the store for a transaction. However, Xstore always sends just one type - Customer ID or CUSTID. This customer ID type is part of the default Sales Audit implementation and should be configured to "Used" when implementing with Xstore.

Sales Audit Tax Codes

When implementing using US Sales Tax for some or all stores, the tax code sent by Xstore will always be sent as TOTAX. This will be validated against the list of valid non-VAT tax codes in Sales Audit. This code is included in the initial Sales Audit configuration, but care should be taken not to remove or configure that code type off when integrating with Xstore.

Reference Codes/Labels

Sales Audit has a flexible method of taking in reference fields from a POS at various levels of the transaction. These can be configured differently by transaction and sub-transaction type within Sales Audit. In general, these are not used in the base Xstore implementation without customization with a couple of exceptions. The reference fields that Xstore uses are as follows:

- Reference Number 1 (Transaction Header) - for the Day Close (DCLOSE) transaction type, this field will contain the file counter for the end of day
- Reference Number 1 (Transaction Header) - for the Tender Total (TOTAL) transaction type, contains the tender ID/type
- Reference Number 3 (Transaction Header) - includes the employee ID if the sale was to an employee

For more on configuring reference fields in Sales Audit, see the *Oracle Retail Sales Audit Implementation Guide*.

Sale Return Transaction

This section describes sale return transaction mappings.

Transaction Header Mapping

This section describes mappings for the transaction header of sale return transactions. The transaction header format is defined at TransactionHeader RECODE_FORMAT in the RTLogFormatConfig.xml file.

A sale return transaction can be mixed with a sale line item, return line item, special order pickup line item, special order payment, work order line item, and so on. In this case, the sale and return transaction type and sub transaction type are defined as SALE. The actual types of line lines are specified at line item level.

The field Salesperson in Transaction Header is matched to -1. It means there is no Salesperson available at transaction level. Instead the field Salesperson in line item is populated, since it is possible to have different sales people for different line items in the same transaction.

OriginalTransactionNumber and Orig_reg_no are only populated for post void transactions. Mapper PostVoidOriginalTransactionInfoMapper indicates the mapping logic.

Line Item (TITEM) Mapping

This section describes mappings for line items of sale return transactions. The line item format is defined at TransactionItem RECODE_FORMAT in the RTLogFormatConfig.xml file.

The line item will not be exported for suspended, cancelled and post void transactions. It is disabled in retailTrnDetailExportabilityMapper.

It would be one or more line item records in a sale return transaction.

There are two kinds of item types, physical/merchandise item (ITEM) or non physical/non merchandise item (NMITEM). Non physical items are service fee, payment, work order item, and so on.

The Item id of a physical item is set to Item field; the Item id for a non physical item is set to NonMerchandiseItem field of the RTLog record.

The default value of the ItemStatus field is S, which is set in the RTLogFormatConfig.xml file. If the line item is return, the value is set to R. Return reason code mapping can be found in /retailTransaction/lineItems/return source field VALUE_MAPPINGS.

The default value of SalesType for sale return line item is I (internal sale).

ItemVoidStatusMapper sets the void flag of the record to true if the line is voided. If a sale return line item record is voided, a clone of the sale return line item record will be created, the item status for the new record is set to "V". In this case the status of the two lines are "S" and "V" or "R" and "V". Sales Audit balances off a sale/return line with its voided line to zero.

Item Discount (IDISC) Mapping and Round off Discount Mapping

This section describes mappings for item level discounts of sale return transactions. The item level discount format is defined at ItemDiscount RECODE_FORMAT in the RTLogFormatConfig.xml file.

The item level discount will not be exported for suspended, cancelled and post void transactions. It is disabled in retailTrnDetailExportabilityMapper.

It would be zero or more item discount records for a line item.

If it is a Deal discount, the RMSPromotionType is mapped to 9999, otherwise it is mapped to 1004.

The discount type of ItemDiscount is mapped from reasonCode and discountReasonCode of RetailPriceModifierType Poslog object.

The values of DiscountReferenceNumber, PromoComponent and RMSPromotionType are mapped from PromotionIDMapper. If the retail price modifier is voided, DiscountReferenceNumber and PromoComponent are set to zero. If the retail price modifier is not voided and it is a deal promotion, it is mapped based on the following conditions:

- If it is an RPM promotion, DiscountReferenceNumber is set to RPM promotion id and PromoComponent is set to promotion component id.

- If it is not an RPM promotion, `RMSPromotionType` is set to 2000, `DiscountReferenceNumber` and `PromoComponent` are set to zero.

`PromotionIDMapper` implements this logic.

`DiscountAmountMapper` calculates unit discount amount for the `ItemDiscount` RTLog IDISC record and also sets `roundedOffAmount` to the RTLog record. The value of unit discount amount is the total discount amount divided by item quantity rounding by half up with unit discount scale. The value of `roundedOffAmount` is the difference between total discount amount and unit discount amount multiplies item quantity. Note that the `roundedOffAmount` is not a field of IDISC, therefore it is not going to be exported.

A round off discount record (`ItemRoundedOffDiscount`) is always created after a discount record (`ItemDiscount`) if the `roundedOffAmount` is not equal to zero. In a discount record, if the `roundedOffAmount` exists, a round off discount record will be cloned from the discount record, otherwise a new round off discount record will be created. The value of discount amount of a round off discount record is the round off amount of the corresponding discount record. The value of the quantity in a round off discount record is always set to zero. `RoundedOffDiscountAmountMapper` is specially used for this mapping.

The format of `ItemRoundedOffDiscount` record and `ItemDiscount` record are exactly the same, that is IDISC. The reason for adding a round off discount record is that Poslog uses `discountAmount` and RTLog uses `UnitDiscountAmount`. If a quantity of a line item is not one, such as 3, the `UnitDiscountAmount` times 3 is not equal to `discountAmount` in Poslog. To balance it off, an extra discount line is introduced to set the round off amount to unit discount amount. It guaranties the total discount value is equal to sum of the two IDISC records.

Item Level Tax Mapping

This section describes mappings for item level tax of sale return transactions. The item level tax format is defined at `ItemGlobalTax RECODE_FORMAT` in the `RTLogFormatConfig.xml` file.

Tax lines will not be exported for suspended, cancelled and post void transactions. It is disabled in `retailTrnDetailExportabilityMapper`.

It would be one or more tax lines for one sale return line item. It depends on how many tax authorities are applied on this item.

RTLog requires four digits decimal for `TaxRate` in `ItemGlobalTax` record format. `AmountMapper` object sets the scale of tax rate to four to fit the need.

RTLog requires four digits decimal for `TaxAmount` in `ItemGlobalTax` record format. `AmountMapper` object sets the scale of tax amount to four to fit the need.

The value of `TaxAmountSign` field is P when tax amount is positive, N when tax amount is negative. It is done in `SignMapper`.

The RTLog Generator application supports either VAT or non VAT. It does not support both of them in the application instance. VAT and non VAT configurable settings can be found at the `roundedOffTaxAmountMapper` spring bean in the `RTLogMappingBean.xml` file.

`TaxCode` mappings are different for VAT and non VAT systems. If the RTLog Generator application is set to VAT, the tax group id maps to the tax code. If the RTLog Generator application is set to non VAT, the tax code is always "TOTTAX". `ItemTaxCodeMapper` implements this logic.

The value of TotalTaxAmount is blank if the tax is voided. Otherwise, the value of TotalTaxAmount is the sum of the item taxes from all item tax authorities. It is implemented at ItemTotalTaxAmountMapper.

The value of TaxableIndicator of ItemGlobalTax is a flag to indicate the item has tax or not. If the TaxAmountSign amount sign is zero, the value of TaxableIndicator is set to N, otherwise it is set to Y.

Tender Line Mapping

This section describes mappings for transaction tender of sale return transactions. The transaction tender format is defined at TransactionTender RECODE_FORMAT in the RTLogFormatConfig.xml file.

The tender line will not be exported for suspended, cancelled and post void transactions. It is disabled in retailTrnTenderExportabilityMapper.

The actual mappings of transaction tender type group and tender type id (sub type) are listed on sourcePath "/transaction/lineItems/tender" in the RTLogMappingConfig.xml file.

There may be zero or more tender line items within one sale return transaction.

The tender type id is different for base currency and foreign currency and is also different for base traveler's check and foreign traveler's check. Object tenderTypeIDMapper is responsible for doing the actual currency and traveler's check mapping. If the tender type is currency or traveler's check, mapping value will be found from the mapper, the corresponding VALUE_MAPPINGS will be skipped. Otherwise, it will do the value mapping in the VALUE_MAPPINGS.

RTLog requires four digits decimal for TenderAmount in TransactionTender record format. Mapper amountMapper sets the scale of tender amount to four to fit the need.

The value of TenderSign field will be P when tender amount is positive, N when tender amount is negative. It is done in SignMapper.

There is a special tender line which is used for transaction rounded off amount. For detail information, refer to the next section.

Rounded Off Amount

Transaction rounded amount is used to balance off a transaction. The amount comes from the following areas.

- Rounded amount from roundedTotal@RetailTransactionType Poslog object.
- Total tax amount from the sum of amount@TaxType of each tax LineItem of RetailTransactionType Poslog object.
- Total sale return line item tax amount from the sum of amount@TaxType of SaleType of LineItem of RetailTransactionType Poslog object.

Transaction rounded off amount = value of 1 + (value of 2 - value of 3).

RTLog exports line item taxes rather than exports transaction level taxes. The reason is that the payment amount in an order transaction includes a certain percentage of item price and tax amount. When doing a pickup, the transaction level tax is not zero, and the payment already includes the part of the tax. The total tax will be more than the value is supposed to be. The transaction will not balance off.

A special RECORD_FORMAT TransactionTenderRoundedOff is created for this purchase. The syntax of the format is exactly the same as the one for

TransactionTender. The TenderTypeGroup is always set to CASH and TenderTypeID is always 1020. TenderAmount of TransactionTenderRoundedOff is the transaction rounded of amount, TenderSign is the transaction rounded of amount sign. RoundedOffAmountMapper accumulates the rounded amounts based on the algorithm above and sets the value and sign to TenderAmount and TenderSign fields.

This special tender line will not be exported for suspended, cancelled and post void transactions. It is disabled in retailTrnDetailExportabilityMapper. It will not be exported when the total rounded amount is zero.

Appendix: Flat File Mapping

This appendix describes the mapping from the RMS and RPM flat file format to the Xstore database format. Details are provided for the following mappings:

- [RMS Diff Group Detail](#)
- [RMS Diff Group Header](#)
- [RMS Diffs](#)
- [RMS Item Header](#)
- [RMS Item Location](#)
- [RMS Merchandise Hierarchy](#)
- [RMS Organizational Hierarchy](#)
- [RMS Related Item Detail](#)
- [RMS Related Item Header](#)
- [RMS Store](#)
- [RMS Store Address](#)
- [RMS VAT](#)
- [RMS VAT Item](#)
- [RPM Clearance Price](#)
- [RPM Regular Price](#)
- [RPM Promotions](#)

RMS Diff Group Detail

[Table B-1](#) describes the RMS Diff Group Detail mapping.

Table B-1 RMS Diff Group Detail Mapping

| P o s | RMS Field Name | RMS Description | RMS Field Type | RMS Max Width | Xstore Table | Xstore Column | Xstore Data Type | Xstore Default Value | Xstore Comment |
|-------------|----------------------|--|----------------------|---------------------|-----------------|------------------|------------------------|----------------------------|-------------------|
| 0 | Family | Value is always DIFFGRPDTL. Data will always be present in this field. | Character | 20 | NA | NA | NA | NA | Ignored. |

Table B-1 (Cont.) RMS Diff Group Detail Mapping

| P o s | RMS Field Name | RMS Description | RMS Field Type | RMS Max Width | Xstore Table | Xstore Column | Xstore Data Type | Xstore Default Value | Xstore Comment |
|-------------|----------------------|--|----------------------|---------------------|-----------------------|------------------|------------------------|----------------------------|--------------------|
| 1 | Type | Value is always FULL when the program is run in full mode. If the program is run in delta mode, the type will vary based on the type of change (CRE, MOD, DEL). Valid values are: DIFFGRPDTLCRE - Creation of diff group group header information. DIFFGRPDTLMOD - Modification of diff group header information. DIFFGRPDTLDEL - Deletion of diff group header information. Data will always be present in this field. | Character | 15 | NA | NA | NA | NA | Determines action. |
| 2 | DiffGroupId | This field holds the ID of the diff group. Data will always be present in this field. | Character | 10 | rms_diff_group_detail | DIFF_GROUP_ID | VARCHAR2(10 CHAR) | NA | NA |
| 3 | DiffId | This field holds a unique number ID for the diff. Data will always be present in this field. | Character | 10 | rms_diff_group_detail | DIFF_ID | VARCHAR2(10 CHAR) | NA | NA |
| 4 | Display Seq | Optional sequence to describe the order in which diffs within the diff group should be displayed in user interfaces. Data is optional in this field. | Character | 4 | rms_diff_group_detail | DISPLAY_SEQ | NUMBER(4) | NA | NA |
| NA | NA | NA | NA | NA | rms_diff_group_detail | CREATE_DATE | TIMESTAMP(6) | now() | NA |
| NA | NA | NA | NA | NA | rms_diff_group_detail | UPDATE_DATE | TIMESTAMP(6) | now() | NA |

RMS Diff Group Header

Table B-2 describes the RMS Diff Group Header mapping.

Table B-2 RMS Diff Group Header Mapping

| P o s | RMS Field Name | RMS Description | RMS Field Type | RMS Max Width | Xstore Table | Xstore Column | Xstore Data Type | Xstore Default Value | Xstore Comment |
|----------------------|-------------------------------|---|-------------------------------|------------------------------|-------------------------|--------------------------|---------------------------------|-------------------------------------|---------------------------|
| 0 | Family | Value is always DIFFGRPHDR. Data will always be present in this field. | Character | 20 | NA | NA | NA | NA | Ignored. |
| 1 | Type | Value is always FULL when the program is run in full mode. If the program is run in delta mode, the type will vary based on the type of change (CRE, MOD, DEL). Valid values are: DIFFGRPHDRCRE - Creation of diff group group header information. DIFFGRPHDRMOD - Modification of diff group header information. DIFFGRPHDRDEL - Deletion of diff group header information. Data will always be present in this field. | Character | 15 | NA | NA | NA | NA | Determines action. |
| 2 | DiffGroupId | This field holds a unique number ID for the differentiator group. As primary ID, DiffGroupId cannot be modified. Data will always be present in this field. | Character | 10 | rms_diff_group_head | DIFF_GROUP_ID | VARCHAR2(10 CHAR) | NA | NA |

Table B-2 (Cont.) RMS Diff Group Header Mapping

| P o s | RMS Field Name | RMS Description | RMS Field Type | RMS Max Width | Xstore Table | Xstore Column | Xstore Data Type | Xstore Default Value | Xstore Comment |
|----------------------|-------------------------------|--|-------------------------------|------------------------------|-------------------------|--------------------------|---------------------------------|-------------------------------------|---|
| 3 | DiffGroupDesc | Description of the differentiator group (for example: Men's Shirt Sizes, Women's Shoe Sizes, Girls Dress Sizes, Shower Gel Scents, Yogurt Flavors, and so on). Data will always be present in this field in the DIFFGRPHDRCRE and DIFFGRPHDRMOD records. | Character | 120 | rms_diff_group_head | DIFF_GROUP_DESC | VARCHAR2(120 CHAR) | NA | NA |
| 4 | DiffTypeId | This field holds a value of the types of differentiators contained in this differentiator group including, but not limited to: S - size C - color F - flavor E - scent P - pattern Data will always be present in this field in the DIFFGRPHDRCRE and DIFFGRPHDRMOD records. | Character | 6 | rms_diff_group_head | DIFF_TYPE | VARCHAR2(6 CHAR) | NA | NA |
| 5 | DiffTypeDesc | Contains the description of the differentiator type. Data will always be present in this field in the DIFFGRPHDRCRE and DIFFGRPHDRMOD records. | Character | 120 | NA | NA | NA | NA | Ignored. It is already captured in the diff id tab. |
| NA | NA | NA | NA | NA | rms_diff_group_head | CREATE_DATE | TIMESTAMP(6) | NA | NA |
| NA | NA | NA | NA | NA | rms_diff_group_head | UPDATE_DATE | TIMESTAMP(6) | NA | NA |

RMS DiffS

Table B-3 describes the RMS DiffS mapping.

Table B-3 RMS DiffS Mapping

| P o s | RMS Field Name | RMS Description | RMS Field Type | RMS Max Width | Xstore Table | Xstore Column | Xstore Data Type | Xstore Default Value | Xstore Comment |
|-------------|----------------------|---|----------------------|---------------------|-----------------|------------------|------------------------|----------------------------|--|
| 0 | Family | Value is always DIFFS. Data will always be present in this field. | Character | 20 | NA | NA | NA | NA | Ignored. |
| 1 | Type | Value is always FULL when the program is run in full mode. If the program is run in delta mode, the type will vary based on the type of change (CRE, MOD, DEL). Valid values are: DIFFCRE - Creation of diff information. DIFFMOD - Modification of diff information. DIFFDEL - Deletion of diff information. Data will always be present in this field. | Character | 15 | NA | NA | NA | | Determines action. |
| 2 | DiffId | Contains the unique ID of the diff. Data will always be present in this field. | Character | 10 | rms_diff_ids | DIFF_ID | VARCHAR2(10 CHAR) | NA | Also used to populate rms_diff_group_detail for the default Diff Group membership. |
| 3 | DiffDesc | Contains the text description of the diff. Data will always exist in this field for DIFFCRE and DIFFMOD. | Character | 120 | rms_diff_ids | DIFF_DESC | VARCHAR2(120 CHAR) | NA | NA |
| 4 | DiffType | Contains the type code for the diff. All diffs belong to one and only one type. Data will always exist in this field for DIFFCRE and DIFFMOD. | Character | 6 | rms_diff_ids | DIFF_TYPE | VARCHAR2(6 CHAR) | NA | Also used to populate rms_diff_group_head for the default Diff Group. |

Table B-3 (Cont.) RMS Diffs Mapping

| P o s | RMS Field Name | RMS Description | RMS Field Type | RMS Max Width | Xstore Table | Xstore Column | Xstore Data Type | Xstore Default Value | Xstore Comment |
|----------------------|-------------------------------|---|-------------------------------|------------------------------|-------------------------|--------------------------|---------------------------------|-------------------------------------|---|
| 5 | DiffTypeDesc | Contains the description of the diff. Data will always exist in this field for DIFFCRE and DIFFMOD. | Character | 120 | rms_diff_ids | DIFF_TYPE_DESC | VARCHAR2(120 CHAR) | NA | Also used to populate rms_diff_group_head for the default Diff Group. |
| 6 | IndustryCode | Can optionally hold the unique code used by industry standards to identify the differentiator. For example, in the US, the National Retail Federation defines a standard Color and Size Codes that gives retailers, vendors, and manufacturers a common language for product color and size identification for EDI purposes. This supplements the information conveyed by UPC codes. For example, mens pants size combination 32x32 has an NRF code number 10492. | Character | 10 | NA | NA | NA | NA | Ignored. |
| 7 | IndustrySubgroup | Can optionally hold a sub-grouping code used by industry standards to further identify the differentiator. For example, in the US, the National Retail Federation uses a subgroup for colors (for example, purple is defined as 500; dark purple represents a range from 501 to 509, medium purple represents a range from 510 to 519, bright purple represents a range from 520 to 529, and so on). | Character | 10 | NA | NA | NA | NA | Ignored. |
| NA | NA | NA | NA | NA | rms_diff_ids | CREATE_DATE | TIMESTAMP(6) | now() | NA |

Table B-3 (Cont.) RMS Diffs Mapping

| P o s | RMS Field Name | RMS Description | RMS Field Type | RMS Max Width | Xstore Table | Xstore Column | Xstore Data Type | Xstore Default Value | Xstore Comment |
|-------------|----------------------|--------------------|----------------------|---------------------|------------------|------------------|------------------------|----------------------------|-------------------|
| NA | NA | NA | NA | NA | rms_diff_ ids | UPDATE _DATE | TIMESTA MP(6) | now() | NA |

RMS Item Header

Table B-4 describes the RMS Item Header mapping.

Table B-4 RMS Item Header Mapping

| Po s | RMS Field Name | RMS Description | RMS Field Type | RMS Max Width | Xstore Table | Xstore Column | Xstore Data Type | Xstore Default Value | Xstore Comment |
|---------|----------------------|--|----------------------|---------------------|---|--------------------------------------|------------------------|----------------------------|---|
| 0 | Family | Value is always ITEMS. Data will always be present in this field. | Character | 20 | NA | NA | NA | NA | Ignored. |
| 1 | Type | Value is always FULLHDR when the program is run in full mode. In delta mode, valid values are: ITEMHDCRE ITEMHDRMOD ITEMHDRDEL Data will always be present in this field. | Character | 15 | NA | NA | NA | NA | Determines action. |
| 2 | Location | For corporate level files, this field holds the string CORPORATE. In location specific files, this field holds the numeric ID of the store of WH. | Character | 10 | itm_item, itm_ item_ dimensio n_value, itm_ item_ dimensio n_type | ORG_ CODE, ORG_ VALUE | VARCHA R(30) | "*" | Dimension Value and Dimension Type are only used for Style items. |
| 3 | Item | ID of the item. This field will always have data. | Character | 25 | itm_item, itm_ item_ dimensio n_value, itm_ item_ dimensio n_type | ITEM_ID, DIMENSI ON_ SYSTEM | VARCHA R(60) | NA | DIMENSION_SYSTEM is populated with ItemID for Style items. |

Table B-4 (Cont.) RMS Item Header Mapping

| Po s | RMS Field Name | RMS Description | RMS Field Type | RMS Max Width | Xstore Table | Xstore Column | Xstore Data Type | Xstore Default Value | Xstore Comment |
|---------|----------------------|---|----------------------|---------------------|-------------------------------------|-------------------------|------------------------|----------------------------|---|
| 4 | ItemParent | ID of the parent of the item. ItemParents can be used as a mechanism to group items together. The ItemParent will also exist as an Item in another row. Data is optional in this field for many items. | Character | 25 | itm_item, itm_item_cross_ reference | PARENT_ITEM_ID, ITEM_ID | VARCHAR(60) | NA | If the record represents a UPC item, an itm_item_cross_ reference record is created. Otherwise, the parent is stored in itm_item.parent_item_id. If a UPC item, ItemParent is set in itm_item_cross_ reference.item_id, and the item is stored in itm_item_cross_ reference.manufacturer_upc. |
| 5 | ItemGrandparent | ID of the grandparent of the Item. ItemGrandparents can be used as a mechanism to group items together. The ItemGrandparent will also exist as an Item in another row. Data is optional in this field for many items. | Character | 25 | NA | NA | NA | NA | Ignored. |

Table B-4 (Cont.) RMS Item Header Mapping

| Pos | RMS Field Name | RMS Description | RMS Field Type | RMS Max Width | Xstore Table | Xstore Column | Xstore Data Type | Xstore Default Value | Xstore Comment |
|------------|-----------------------|---|-----------------------|----------------------|---------------------|----------------------|-------------------------|-----------------------------|-----------------------|
| 6 | PackInd | <p>This field indicates if the item is a pack. A pack is a type of item that acts as a collection of other items. Packs have a Y value. Regular items have an N value.</p> <p>This field will always have data for the ITEMHDRCRE and ITEMHDRMOD records.</p> | Character | 1 | NA | NA | NA | NA | Ignored. |
| 7 | SimplePackInd | <p>This field indicates if the pack item is a simple pack. A simple pack is a collection that consists of multiple of the same item (for example, 10 pack of the same item). Only simple packs have a Y value. Both complex packs (which consist of a mixture of different items) and regular items have an N value.</p> <p>This field will always have data for the ITEMHDRCRE and ITEMHDRMOD records.</p> | Character | 1 | NA | NA | NA | NA | Ignored. |

Table B-4 (Cont.) RMS Item Header Mapping

| Pos | RMS Field Name | RMS Description | RMS Field Type | RMS Max Width | Xstore Table | Xstore Column | Xstore Data Type | Xstore Default Value | Xstore Comment |
|------------|-----------------------|--|-----------------------|----------------------|---------------------|----------------------|-------------------------|-----------------------------|---|
| 8 | ItemLevel | <p>Number indicating in which of the three levels the item resides. The item level determines if the item stands alone or if it is part of a family of related items.</p> <p>The concept is best explained with the following typical (although not exhaustive) examples.</p> <p>Staple items generally have a item level = 1. UPCs for staple items generally have an item level = 2 (and the staple item is the UPC's parent item).</p> <p>Fashion styles generally have an item level = 1. Fashion skus generally have an item level = 2. UPCs for fashion skus generally have an item level = 3.</p> <p>Valid values are 1, 2, and 3.</p> <p>This field will always have data.</p> | Number | 1 | NA | NA | NA | NA | Used to distinguish Item from Style, and UPC from Item. |

Table B-4 (Cont.) RMS Item Header Mapping

| Po s | RMS Field Name | RMS Description | RMS Field Type | RMS Max Width | Xstore Table | Xstore Column | Xstore Data Type | Xstore Default Value | Xstore Comment |
|---------|----------------------|---|----------------------|---------------------|-----------------|------------------|------------------------|----------------------------|---|
| 9 | TranLevel | <p>Number indicating which of the three levels of transactions occur for the item's group. The transaction level is the level at which the items inventory is tracked in the system. The transaction level item will be inventoried, counted, transferred, shipped, and so on. Only one level of the hierarchy of an item family may contain transaction level items.</p> <p>The concept is best explained with the following typical (although not exhaustive) examples.</p> <p>Staple items generally have a TranLevel = 1. UPCs for staple items generally have a TranLevel = 1 (inventory transactions occur at the staple sku level; sales of the item roll up to the parent staple sku).</p> <p>Fashion styles generally have a TranLevel = 2 (the style itself is not sold/inventoried).</p> <p>Fashion skus generally have an TranLevel = 2 (the fashion sku is sold/inventoried).</p> <p>UPCs for fashion skus generally have a TranLevel = 2 (the fashion sku is sold/inventoried).</p> | Number | 1 | NA | NA | NA | NA | Used to distinguish Item from Style, and UPC from Item. |

Table B-4 (Cont.) RMS Item Header Mapping

| PoS | RMS Field Name | RMS Description | RMS Field Type | RMS Max Width | Xstore Table | Xstore Column | Xstore Data Type | Xstore Default Value | Xstore Comment |
|-----|--------------------------|---|----------------|---------------|--------------|----------------------|------------------|----------------------|------------------------------|
| | TranLevel (Continued) | There are some rare cases in vendor-managed inventory where the TranLevel equals 3. Valid values are 1, 2, and 3. This field will always have data. | | | | | | | |
| 10 | InventoryInd | This indicator is used to determine if inventory is held for the item/item family. Inventory is not held (value = N) in some cases, such as: Concession items that are sold by independent in location concessions. Consignment items that are not owned by the retailer; financial and inventory processing occurs after the item is sold to a consumer. Containers sold/returned for deposit. Some items that are transformed for sale. Valid values are Y and N. This field will always have data for the ITEMHDRCRE and ITEMHDRMOD records. | Character | 1 | itm_item | NOT_INVENTORIED_FLAG | NUMBER(1,0) | NA | When N, then 1; otherwise 0. |

Table B-4 (Cont.) RMS Item Header Mapping

| Po s | RMS Field Name | RMS Description | RMS Field Type | RMS Max Width | Xstore Table | Xstore Column | Xstore Data Type | Xstore Default Value | Xstore Comment |
|---------|----------------------|---|----------------------|---------------------|---------------------------------|------------------|------------------------|----------------------------|--|
| 11 | Diff1Level | <p>This field describes whether the Diff1 information represents a Diff Group or Diff ID.</p> <p>A Diff Group is a collection of possible Diff IDs. When assigned to a parent item, the diffs in the diff group limit the possible diff IDs that can be assigned to a child of the parent item.</p> <p>Valid Values are GROUP and ID.</p> <p>Data is optional in this field for many items.</p> | Character | 6 | NA | NA | NA | NA | Used to identify Style Item. When GROUP, DataLoader fetches RMS DIFF Group Details to create Dimension System information. When ID, DataLoader fetches the Default DIFF Group for the Diff1Type. |
| 12 | Diff1Type | <p>This field contains the type of the Diff1 information. Valid values can be configured by the retailer. Common examples include:</p> <p>C - Color</p> <p>WS - Waist Sizes</p> <p>F - Flavor</p> <p>SC - Scent</p> <p>Note that these examples are not consistent or required. They are presented here simply to help explain the field.</p> <p>Data is optional in this field for many items.</p> | Character | 6 | itm_ item_ dimension_type | DIMENSION | VARCHAR2(30 CHAR) | NA | Only used when Diff1Level is provided. |

Table B-4 (Cont.) RMS Item Header Mapping

| Pos | RMS Field Name | RMS Description | RMS Field Type | RMS Max Width | Xstore Table | Xstore Column | Xstore Data Type | Xstore Default Value | Xstore Comment |
|------------|-----------------------|--|-----------------------|----------------------|---------------------|----------------------|-------------------------|-----------------------------|--|
| 13 | Diff1 | This field contains the ID of the Diff1 information. If the Diff1Level is GROUP, this field contains a Diff Group ID. If the Diff1Level is ID, it contains the ID of a diff (a size, color, and so on.) This field is optional. | Character | 10 | itm_item | DIMENSION1 | VARCHAR2(60 CHAR) | NA | NA |
| 14 | Diff2Level | This field describes whether the Diff2 information represents a Diff Group or Diff ID. A Diff Group is a collection of possible Diff IDs. When assigned to a parent item, the diffs in the diff group limit the possible diff IDs that can be assigned to a child of the parent item. Valid values are GROUP and ID. Data is optional in this field for many items. | Character | 6 | NA | NA | NA | NA | Used to identify Style Item. When GROUP, DataLoader fetches RMS DIFF Group Details to create Dimension System information. When ID, DataLoader fetches the Default DIFF Group for the Diff2Type. |

Table B-4 (Cont.) RMS Item Header Mapping

| Po s | RMS Field Name | RMS Description | RMS Field Type | RMS Max Width | Xstore Table | Xstore Column | Xstore Data Type | Xstore Default Value | Xstore Comment |
|---------|----------------------|--|----------------------|---------------------|-------------------------|------------------|------------------------|----------------------------|--|
| 15 | Diff2Type | <p>This field contains the type of the Diff2 information. Valid values can be configured by the retailer. Common examples include:</p> <p>C - Color WS - Waist Sizes F - Flavor SC - Scent</p> <p>Note that these examples are not consistent or required. They are presented here simply to help explain the field.</p> <p>Data is optional in this field for many items.</p> | Character | 6 | itm_item_dimension_type | DIMENSION | VARCHAR2(30 CHAR) | NA | Only used when Diff2Level is provided. |
| 16 | Diff2 | <p>This field contains the ID of the Diff2 information. If the Diff1Level is GROUP, this field contains a Diff Group ID. If the Diff1Level is ID, it contains the ID of a diff (a size, color, and so on).</p> <p>Data is optional in this field for many items.</p> | Character | 10 | itm_item | DIMENSION2 | VARCHAR2(60 CHAR) | NA | NA |

Table B-4 (Cont.) RMS Item Header Mapping

| Pos | RMS Field Name | RMS Description | RMS Field Type | RMS Max Width | Xstore Table | Xstore Column | Xstore Data Type | Xstore Default Value | Xstore Comment |
|------------|-----------------------|---|-----------------------|----------------------|-------------------------|----------------------|-------------------------|-----------------------------|--|
| 17 | Diff3Level | <p>This field describes whether the Diff3 information represents a Diff Group or Diff ID.</p> <p>A Diff Group is a collection of possible Diff IDs. When assigned to a parent item, the diffs in the diff group limit the possible diff IDs that can be assigned to a child of the parent item.</p> <p>Valid values are GROUP and ID.</p> <p>Data is optional in this field for many items.</p> | Character | 6 | NA | NA | NA | NA | Used to identify Style Item. When GROUP, DataLoader fetches the RMS DIFF Group Details to create the Dimension System information. When ID, DataLoader fetches the Default DIFF Group for the Diff3Type. |
| 18 | Diff3Type | <p>This field contains the type of the Diff3 information. Valid values can be configured by the retailer. Common examples include:</p> <p>C - Color WS - Waist Sizes F - Flavor SC - Scent</p> <p>Note that these examples are not consistent or required. They are presented here simply to help explain the field.</p> <p>Data is optional in this field for many items.</p> | Character | 6 | itm_item_dimension_type | DIMENSION | VARCHAR2(30 CHAR) | NA | Only used when Diff3Level is provided. |

Table B-4 (Cont.) RMS Item Header Mapping

| Pos | RMS Field Name | RMS Description | RMS Field Type | RMS Max Width | Xstore Table | Xstore Column | Xstore Data Type | Xstore Default Value | Xstore Comment |
|------------|-----------------------|--|-----------------------|----------------------|---------------------|----------------------|-------------------------|-----------------------------|-----------------------|
| 19 | Diff3 | This field contains the ID of the Diff3 information. If the Diff1Level is GROUP, this field contains a Diff Group ID. If the Diff1Level is ID, it contains the ID of a diff (a size, color, and so on). Data is optional in this field for many items. | Character | 10 | itm_item | DIMENSION3 | VARCHAR2(60 CHAR) | NA | NA |
| 20 | Diff4Level | This field describes whether the Diff4 information represents a Diff Group or Diff ID. A Diff Group is a collection of possible Diff IDs. When assigned to a parent item, the diffs in the diff group limit the possible diff IDs that can be assigned to a child of the parent item. Valid values are GROUP and ID. Data is optional in this field for many items. | Character | 6 | NA | NA | NA | NA | Ignored. |

Table B-4 (Cont.) RMS Item Header Mapping

| Po s | RMS Field Name | RMS Description | RMS Field Type | RMS Max Width | Xstore Table | Xstore Column | Xstore Data Type | Xstore Default Value | Xstore Comment |
|---------|----------------------|--|----------------------|---------------------|-----------------|-------------------------------------|------------------------|----------------------------|--|
| 21 | Diff4Type | This field contains the type of the Diff4 information. Valid values can be configured by the retailer. Common examples include: C - Color WS - Waist Sizes F - Flavor SC - Scent Note that these examples are not consistent or required. They are presented here simply to help explain the field. Data is optional in this field for many items. | Character | 6 | NA | NA | NA | NA | Ignored. |
| 22 | Diff4 | This field contains the ID of the Diff4 information. If the Diff1Level is GROUP, this field contains a Diff Group ID. If the Diff1Level is ID, it contains the ID of a diff (a size, color, and so on). Data is optional in this field for many items. | Character | 10 | NA | NA | NA | NA | Ignored. |
| 23 | Dept | Number identifying the department in the merchandise hierarchy to which the item belongs. If the item has a parent, the item's department will be the same as that of its parent. This field will always have data for the ITEMHDRCRE and ITEMHDRMOD records. | Number | 4 | itm_item | MERCH_LEVEL_2 (or configured level) | VARCHAR2(60 CHAR) | NA | D is always appended to the value. The value is used to look up the parent from the loc_org_hierarchy to populate merch_level_1. |

Table B-4 (Cont.) RMS Item Header Mapping

| Pos | RMS Field Name | RMS Description | RMS Field Type | RMS Max Width | Xstore Table | Xstore Column | Xstore Data Type | Xstore Default Value | Xstore Comment |
|------------|-----------------------|--|-----------------------|----------------------|---------------------|-------------------------------------|-------------------------|-----------------------------|------------------------------------|
| 24 | Class | <p>Number identifying the class in the merchandise hierarchy to which the item belongs.</p> <p>If the item has a parent, the item's class will be the same as that of its parent.</p> <p>This field will always have data for the ITEMHDRCRE and ITEMHDRMOD records.</p> | Number | 4 | NA | NA | NA | NA | Ignored. |
| 25 | Unique Class | <p>Number uniquely identifying the class node to which the item belongs.</p> <p>Class ID is not unique in the level of the merchandise hierarchy. The combination of Dept/Class is unique, but requires the use of a composite key.</p> <p>This field will always have data for the ITEMHDRCRE and ITEMHDRMOD records.</p> | Number | 10 | itm_item | MERCH_LEVEL_3 (or configured level) | VARCHAR2(60 CHAR) | NA | C is always appended to the value. |

Table B-4 (Cont.) RMS Item Header Mapping

| Po s | RMS Field Name | RMS Description | RMS Field Type | RMS Max Width | Xstore Table | Xstore Column | Xstore Data Type | Xstore Default Value | Xstore Comment |
|---------|------------------------|---|----------------------|---------------------|-----------------|-------------------------------------|--------------------------|----------------------------|------------------------------------|
| 26 | Subclas s | <p>Number identifying the subclass in the merchandise hierarchy to which the item belongs.</p> <p>Subclass ID is not unique in the level of the merchandise hierarchy. The combination of Dept/Class/Subclass is unique.</p> <p>If the item has a parent, the item's class will be the same as that of its parent.</p> <p>This field will always have data for the ITEMHDRCRE and ITEMHDRMOD records.</p> | Number | 4 | NA | NA | NA | NA | Ignored. |
| 27 | Unique Subclas s | <p>Number uniquely identifying the subclass node to which the item belongs.</p> <p>Subclass ID is not unique in the level of the merchandise hierarchy. The combination of Dept/Class/Subclass is unique, but requires the use of a composite key.</p> <p>This field will always have data for the ITEMHDRCRE and ITEMHDRMOD records</p> | Character | 10 | itm_item | MERCH_LEVEL_4 (or configured level) | VARCHA R2(60 CHAR) | NA | S is always appended to the value. |

Table B-4 (Cont.) RMS Item Header Mapping

| Po s | RMS Field Name | RMS Description | RMS Field Type | RMS Max Width | Xstore Table | Xstore Column | Xstore Data Type | Xstore Default Value | Xstore Comment |
|---------|----------------------|---|----------------------|---------------------|-------------------|------------------|------------------------|----------------------------|---|
| 28 | Status | Item life cycle status of the item. For this release, this field always contains the character A as only approved items are exported. This field will always have data for the ITEMHDRCRE and ITEMHDRMOD records. | Character | 1 | NA | NA | NA | NA | Ignored. |
| 29 | ItemDesc | Long description of the item. This field will always have data for the ITEMHDRCRE and ITEMHDRMOD records. | Character | 250 | com_ translations | TRANSLATION | VARCHAR2(4000 CHAR) | NA | If Location is CORPORATE, the com_ translations record is created using ItemDesc. |
| 30 | SecondaryItem Desc | Secondary description of the item. Data is optional in this field. | Character | 250 | NA | NA | NA | NA | Ignored. |
| 31 | ShortDescription | Shortened item description. This field will always have data for the ITEMHDRCRE and ITEMHDRMOD records. | Character | 120 | NA | NA | NA | NA | Ignored. |
| 32 | BrandName | This field contains the brand associated to an item. Data is optional in this field. | Character | 30 | NA | NA | NA | NA | Ignored. |

Table B-4 (Cont.) RMS Item Header Mapping

| Pos | RMS Field Name | RMS Description | RMS Field Type | RMS Max Width | Xstore Table | Xstore Column | Xstore Data Type | Xstore Default Value | Xstore Comment |
|------------|-----------------------|--|-----------------------|----------------------|---------------------|----------------------|-------------------------|-----------------------------|--|
| 33 | MerchandiseInd | <p>Indicates if the item is a merchandise item (Y, N).</p> <p>Merchandise items are generally physical items (things that must be shipped/received and of which there is an inventory). Non-merchandise items are often items which do not have inventory. Common examples include extra fees for service (extended warranties, alterations) or endlessly available items (downloads, in-app purchases of bonus content, subscriptions).</p> <p>All items, both merchandise and non-merchandise, are exported from RMS.</p> <p>This field will always have data for all records.</p> | Character | 1 | NA | NA | NA | NA | If N, then NON_PHYSICAL item so the record is skipped. |

Table B-4 (Cont.) RMS Item Header Mapping

| Po s | RMS Field Name | RMS Description | RMS Field Type | RMS Max Width | Xstore Table | Xstore Column | Xstore Data Type | Xstore Default Value | Xstore Comment |
|---------|---------------------------|---|----------------------|---------------------|-----------------|------------------|------------------------|----------------------------|-------------------|
| 34 | Primary RefItem Ind | Indicates if the sub-transaction level item is designated as the primary sub-transaction level item. For transaction level items and above, the value in this field will be N. For sub-transaction level items, this field may be either Y (if the current record is a primary UPC) or N (if the current record is not the primary UPC). This field will always have data for the ITEMHDRCRE and ITEMHDRMOD records. | Character | 1 | NA | NA | NA | NA | Ignored. |
| 35 | CostZo neGrou pId | Cost zone group associated with the item for ELC calculations. Data is optional in this field. | Number | 4 | NA | NA | NA | NA | Ignored. |
| 36 | Standar dUOM | Unit of measure in which stock of the item is tracked at a corporate level. Unit of measure may have to be manually synchronized between the systems as this is foundation data that is not currently bulk integrated out of Merchandising. This field will always have data for the ITEMHDRCRE and ITEMHDRMOD records. | Character | 4 | NA | NA | NA | NA | Ignored. |

Table B-4 (Cont.) RMS Item Header Mapping

| Po s | RMS Field Name | RMS Description | RMS Field Type | RMS Max Width | Xstore Table | Xstore Column | Xstore Data Type | Xstore Default Value | Xstore Comment |
|---------|-----------------------|---|----------------------|---------------------|-----------------|------------------|------------------------|----------------------------|-------------------|
| 37 | UOMC onvFact or | Conversion factor between an Each and the standard_uom when the standard_uom is not in the quantity class (such as if standard_uom = lb and 1 lb = 10 eaches, this factor will be 10). This factor is used to convert sales and stock data when an item is retailed in eaches, but does not have eaches as its standard unit of measure. Data is optional in this field. | Number | 20 | NA | NA | NA | NA | Ignored. |
| 38 | Package Size | Holds the size of the product printed on any packaging (for example, 24 ounces). This field can be used for reporting purposes and to determine same sized and different sized items. Data is optional in this field. | Number | 12 | NA | NA | NA | NA | Ignored. |
| 39 | Package UOM | Holds the unit of measure associated with the package size. This field can be used for reporting purposes and to determine same sized and different sized items. Data is optional in this field. | Character | 4 | NA | NA | NA | NA | Ignored. |

Table B-4 (Cont.) RMS Item Header Mapping

| Po s | RMS Field Name | RMS Description | RMS Field Type | RMS Max Width | Xstore Table | Xstore Column | Xstore Data Type | Xstore Default Value | Xstore Comment |
|---------|--|---|----------------------|---------------------|-----------------|------------------|------------------------|----------------------------|-------------------|
| 40 | StoreOr dMulti | Merchandise shipped from the warehouses to the stores must be specified in this unit type. Valid values are: C = Cases; I = Inner; E = Eaches. This field will always have data for the ITEMHDRCRE and ITEMHDRMOD records. | Character | 1 | NA | NA | NA | NA | Ignored. |
| 41 | Forecast Ind | Indicates if sales forecasts will be produced for this item. Valid values are: Y, N. This field will always have data for the ITEMHDRCRE and ITEMHDRMOD records. | Character | 1 | NA | NA | NA | NA | Ignored. |
| 42 | Original Retail | The original retail price of the item per unit. This field is stored in the primary currency of the Merchandising systems. Data is optional in this field. | Number | 20 | NA | NA | NA | NA | Ignored. |
| 43 | Original RetailC urrency Code | The currency code of the OriginalRetail. Data is optional in this field. | Character | 3 | NA | NA | NA | NA | Ignored. |
| 44 | MfgRec Retail | Data is optional in this field. | Number | 20 | itm_item | LIST_ PRICE | NUMBE R(17,6) | NA | NA |
| 45 | MfgRec RetailC urrency Code | The currency code of the MfgRecRetail. Data is optional in this field. | Character | 3 | NA | NA | NA | NA | Ignored. |

Table B-4 (Cont.) RMS Item Header Mapping

| Pos | RMS Field Name | RMS Description | RMS Field Type | RMS Max Width | Xstore Table | Xstore Column | Xstore Data Type | Xstore Default Value | Xstore Comment |
|------------|-----------------------|--|-----------------------|----------------------|---------------------|----------------------|-------------------------|-----------------------------|-----------------------|
| 46 | RetailLabelType | This field indicates any special label type associated with an item (that is, pre-priced or cents off). This field is used in RMS for reporting purposes only. Data is optional in this field. | Character | 6 | NA | NA | NA | NA | Ignored. |
| 47 | RetailLabelValue | This field represents the value associated with the RetailLabelType. Data is optional in this field. | Number | 20 | NA | NA | NA | NA | Ignored. |
| 48 | ItemAggregated | This field holds an indicator of whether the retailer wants to aggregate inventory and sales for the item with the inventory and sales of other child items. For staple items, this indicator is N. For fashion items, this indicator may be Y. If this indicator is Y, the retailer may also define which diff positions should be aggregated into item/diff combinations. Aggregated sales and inventory planning data can be used for inventory decision making and other reporting. This field will always have data for the ITEMHDRCRE and ITEMHDRMOD records. | Character | 1 | NA | NA | NA | NA | Ignored. |

Table B-4 (Cont.) RMS Item Header Mapping

| Po s | RMS Field Name | RMS Description | RMS Field Type | RMS Max Width | Xstore Table | Xstore Column | Xstore Data Type | Xstore Default Value | Xstore Comment |
|---------|---------------------------|---|----------------------|---------------------|-----------------|------------------|------------------------|----------------------------|-------------------|
| 49 | Diff1Ag gregatel nd | <p>This field holds an indicator (Y/N) of whether the retailer wants to aggregate inventory and sales for the item by the first Diff defined for the item (such as Style/Color).</p> <p>Aggregated sales and inventory planning data can be used for inventory decision making and other reporting.</p> <p>This field will always have data for the ITEMHDRCRE and ITEMHDRMOD records.</p> | Character | 1 | NA | NA | NA | NA | Ignored. |
| 50 | Diff2Ag gregatel nd | <p>This field holds an indicator (Y/N) of whether the retailer wants to aggregate inventory and sales for the item by the first Diff defined for the item (such as Style/Waist Size).</p> <p>Aggregated sales and inventory planning data can be used for inventory decision making and other reporting.</p> <p>This field will always have data for the ITEMHDRCRE and ITEMHDRMOD records.</p> | Character | 1 | NA | NA | NA | NA | Ignored. |

Table B-4 (Cont.) RMS Item Header Mapping

| Pos | RMS Field Name | RMS Description | RMS Field Type | RMS Max Width | Xstore Table | Xstore Column | Xstore Data Type | Xstore Default Value | Xstore Comment |
|------------|-----------------------|---|-----------------------|----------------------|---------------------|----------------------|-------------------------|-----------------------------|-----------------------|
| 51 | Diff3Aggregateland | <p>This field holds an indicator (Y/N) of whether the retailer wants to aggregate inventory and sales for the item by the first Diff defined for the item (such as Style/Inseam).</p> <p>Aggregated sales and inventory planning data can be used for inventory decision making and other reporting.</p> <p>This field will always have data for the ITEMHDCRE and ITEMHDRMOD records.</p> | Character | 1 | NA | NA | NA | NA | Ignored. |
| 52 | Diff4Aggregateland | <p>This field holds an indicator (Y/N) of whether the retailer wants to aggregate inventory and sales for the item by the first Diff defined for the item (such as Style/Pattern).</p> <p>Aggregated sales and inventory planning data can be used for inventory decision making and other reporting.</p> <p>This field will always have data for the ITEMHDCRE and ITEMHDRMOD records.</p> | Character | 1 | NA | NA | NA | NA | Ignored. |

Table B-4 (Cont.) RMS Item Header Mapping

| Po s | RMS Field Name | RMS Description | RMS Field Type | RMS Max Width | Xstore Table | Xstore Column | Xstore Data Type | Xstore Default Value | Xstore Comment |
|---------|------------------------|---|----------------------|---------------------|-----------------|------------------|------------------------|----------------------------|-------------------|
| 53 | ItemNu mberTy pe | <p>This field holds a code descriptor of the type of item number. Valid values for this field include, but are not limited to:</p> <p>ITEM - Oracle Retail Item Number</p> <p>UPC-A - UCC12</p> <p>UPC-AS - UCC12 with Supplement</p> <p>UPC-E - UCC8</p> <p>UPC-ES - UCC8 with Supplement</p> <p>EAN8 - EAN/UCC-8</p> <p>EAN13 - EAN/UCC-13</p> <p>EAN13S - EAN/UCC-13 with Supplement</p> <p>ISBN10 - SBN-10</p> <p>ISBN13 - ISBN-13</p> <p>NDC - NDC/NHRIC - National Drug Code</p> <p>PLU - PLU</p> <p>VPLU - Variable Weight PLU</p> <p>SSCC - SSCC Shipper Carton</p> <p>UCC14 - EAN/UCC-14</p> <p>MANL - Manual</p> <p>This field will always have data for the ITEMHDRCRE and ITEMHDRMOD records.</p> | Character | 6 | NA | NA | NA | NA | Ignored. |

Table B-4 (Cont.) RMS Item Header Mapping

| Pos | RMS Field Name | RMS Description | RMS Field Type | RMS Max Width | Xstore Table | Xstore Column | Xstore Data Type | Xstore Default Value | Xstore Comment |
|------------|-----------------------|--|-----------------------|----------------------|---------------------|----------------------|-------------------------|-----------------------------|-----------------------|
| 54 | FormatID | <p>This field holds the format ID that corresponds to the item's variable UPC. This value is only used for items with variable weight PLU/variable weight UPCs (ItemNumberType = VPLU). Valid values include, but are not limited to:</p> <p>A 1-4-6-1 B 1-5-5-1 C 1-4-6-1 D 1-5-5-1 E 2-4-5-1 F 2-4-5-1 G 2-5-4-1 H 2-5-5-1 I 2-4-6-1 J 2-4-6-1 K 2-5-5-1</p> <p>In the case of a type A, this means the total barcode will be at least 6 digits. The prefix of the barcode will be 1 digit in the first position. The item ID is 4 digits and will be in digits 2 to 5 of the barcode. The variable weight will start in digit 6 of the barcode and be at least 1 (possibly more digits).</p> <p>Data is optional in this field, and can only exist for the ItemNumberType = VPLU.</p> | Character | 1 | NA | NA | NA | NA | Ignored. |

Table B-4 (Cont.) RMS Item Header Mapping

| Po s | RMS Field Name | RMS Description | RMS Field Type | RMS Max Width | Xstore Table | Xstore Column | Xstore Data Type | Xstore Default Value | Xstore Comment |
|---------|-------------------------|---|----------------------|---------------------|-----------------|------------------|------------------------|----------------------------|-------------------|
| 55 | Prefix | This column holds the prefix for variable weight UPCs. The size of the prefix (1 or 2 digits) depends on the FormatID. The prefix is used to standardize variable weight UPCs. Data is optional in this field, and can only exist for the ItemNumberType = VPLU. | Number | 2 | NA | NA | NA | NA | Ignored. |
| 56 | RecHan dlingTem p | Holds the temperature information associated with the item. Valid values include, but are not limited to: FRIDGE - Keep Refrigerated at 35°F/1.6°C FROZEN - Keep Frozen at 0°F/-18C ROOM - Keep at Room Temperature Data is optional in this field. | Character | 6 | NA | NA | NA | NA | Ignored. |
| 57 | RecHan dlingSe ns | Holds the sensitivity information associated with the item. Valid values include, but are not limited to: AERO - Aerosol Container - flammable COMPUS - Combustible EXPLOD - Explosive FRAG - Fragile TOXIC - Toxic Data is optional in this field. | Character | 6 | NA | NA | NA | NA | Ignored. |

Table B-4 (Cont.) RMS Item Header Mapping

| PoS | RMS Field Name | RMS Description | RMS Field Type | RMS Max Width | Xstore Table | Xstore Column | Xstore Data Type | Xstore Default Value | Xstore Comment |
|-----|----------------|--|----------------|---------------|--------------|---------------|------------------|----------------------|----------------|
| 58 | PerishableInd | Grocery item attribute used to indicate whether an item is perishable. Valid values are Y and N. This field will always have data for the ITEMHDRCRE and ITEMHDRMOD records. | Character | 1 | NA | NA | NA | NA | Ignored. |
| 59 | WasteType | Identifies the wastage type as either sales or spoilage wastage. Sales wastage occurs during processes that make an item saleable (for example, fat is trimmed off at customer request). Spoilage wastage occurs during the products shelf life (for example, evaporation causes the product to weigh less after a period of time). Valid values are: SP - Spoilage SL - Sales Wastage is not applicable to pack items. Data is optional in this field. | Character | 6 | NA | NA | NA | NA | Ignored. |
| 60 | WastePct | Average percent of wastage for the item over its shelf life. Used in inflating the retail price for wastage items. Data is optional in this field. | Number | 12 | NA | NA | NA | NA | Ignored. |

Table B-4 (Cont.) RMS Item Header Mapping

| Po s | RMS Field Name | RMS Description | RMS Field Type | RMS Max Width | Xstore Table | Xstore Column | Xstore Data Type | Xstore Default Value | Xstore Comment |
|---------|--------------------------|--|----------------------|---------------------|-----------------|------------------|------------------------|----------------------------|-------------------|
| 61 | Default WastePc t | Default daily wastage percent for spoilage type wastage items. This value defaults to all item locations and represents the average amount of wastage that occurs on a daily basis. Data is optional in this field. | Number | 12 | NA | NA | NA | NA | Ignored. |
| 62 | Constan tDimIn d | This field will always have data. | Character | 1 | NA | NA | NA | NA | Ignored. |
| 63 | Contain sInnerI nd | Indicates that the dimensions of the product are always the same, regardless of the supplier. Valid values are Y and N. This field will always have data for the ITEMHDRCRE and ITEMHDRMOD records. | Character | 1 | NA | NA | NA | NA | Ignored. |
| 64 | Sellable Ind | Indicates if the item may be sold as a unit. If it is Y, the item will have its own unique unit retail. If it is N, the item will not have a unit retail. This field will always have data for the ITEMHDRCRE and ITEMHDRMOD records. | Character | 1 | NA | NA | NA | NA | Ignored. |

Table B-4 (Cont.) RMS Item Header Mapping

| Po s | RMS Field Name | RMS Description | RMS Field Type | RMS Max Width | Xstore Table | Xstore Column | Xstore Data Type | Xstore Default Value | Xstore Comment |
|---------|----------------------|--|----------------------|---------------------|-----------------|------------------|------------------------|----------------------------|-------------------|
| 65 | OrderableInd | <p>Indicates if the item may be ordered as a unit.</p> <p>Valid values are:</p> <p>Y - Yes, this item/pack may be ordered from a single supplier.</p> <p>N - No, this item/pack may not be ordered from a single supplier.</p> <p>Non-orderable items might be transformed or produced (in-house bakery items for example) by the retailer. Examples might include, but are not limited to:</p> <p>Ground beef item is not orderable because the retailer orders a side of beef and transforms it into ground beef.</p> <p>In-house bakery items are not orderable because the retailer produces them in house.</p> <p>This field will always have data for the ITEMHDRCRE and ITEMHDRMOD records.</p> | Character | 1 | NA | NA | NA | NA | Ignored. |

Table B-4 (Cont.) RMS Item Header Mapping

| Pos | RMS Field Name | RMS Description | RMS Field Type | RMS Max Width | Xstore Table | Xstore Column | Xstore Data Type | Xstore Default Value | Xstore Comment |
|------------|-----------------------|--|-----------------------|----------------------|---------------------|----------------------|-------------------------|-----------------------------|-----------------------|
| 66 | PackType | <p>Indicates if a pack item is a vendor pack or a buyer pack.</p> <p>A vendor pack is a pack that the vendor or supplier recognizes and sells to the retailer. If the pack item is a vendor pack, communication with the supplier will use the vendor pack number.</p> <p>A buyer pack is a pack that a buyer has created for ease of use within the retail business. If the pack item is a buyer pack, communication with the supplier will explode the pack out to its component items.</p> <p>This field will only contain data if the item is a pack item. If the pack item is not orderable, this field must be NULL. Valid values are: V - Vendor, B - Buyer.</p> <p>Data is optional in this field.</p> | Character | 1 | NA | NA | NA | NA | Ignored. |

Table B-4 (Cont.) RMS Item Header Mapping

| PoS | RMS Field Name | RMS Description | RMS Field Type | RMS Max Width | Xstore Table | Xstore Column | Xstore Data Type | Xstore Default Value | Xstore Comment |
|-----|------------------|---|----------------|---------------|--------------|---------------|------------------|----------------------|----------------|
| 67 | OrderAsType | Indicates if a pack item is receivable at the component level or at the pack level (for a buyer pack only). This field is required if a pack item is an orderable buyer pack. This field must be NULL if the pack is sellable only or a vendor pack. This field will only be available if the item is a pack item. Valid values are: E - Eaches (component level) P - Pack (buyer pack only) Data is optional in this field. | Character | 1 | NA | NA | NA | NA | Ignored. |
| 68 | ItemServiceLevel | Default shipping option for consumers. Valid values include, but are not limited to: 2DAY - Second Day GRND - Ground OVRNT - Overnight POVRNT - Priority Overnight Data is optional in this field. | Character | 6 | NA | NA | NA | NA | Ignored. |
| 69 | GiftWrapInd | This field contains a value of Y if the item is eligible to be gift wrapped. If not explicitly defined, this field defaults to N. This field will always have data for the ITEMHDRCRE and ITEMHDRMOD records. | Character | 1 | NA | NA | NA | NA | Ignored. |

Table B-4 (Cont.) RMS Item Header Mapping

| Po s | RMS Field Name | RMS Description | RMS Field Type | RMS Max Width | Xstore Table | Xstore Column | Xstore Data Type | Xstore Default Value | Xstore Comment |
|---------|------------------------|--|----------------------|---------------------|-----------------|------------------|------------------------|----------------------------|-------------------|
| 70 | ShipAlo neInd | This field contains a value of Y if the item must be shipped alone to consumers. If not explicitly defined, this field defaults to N. This field will always have data for the ITEMHDRCRE and ITEMHDRMOD records. | Character | 1 | NA | NA | NA | NA | Ignored. |
| 71 | ItemXfo rmInd | This field contains a value of Y if the item is associated with an item transformation. The item will either be the sellable item or orderable item in the transformation process. Transformations include a side of beef (orderable transformation item) cut into steaks/ground beef and so on (sellable transformation items). This field will always have data for the ITEMHDRCRE and ITEMHDRMOD records. | Character | 1 | NA | NA | NA | NA | Ignored. |
| 72 | CatchW eightIn d | Indicates whether the item should be weighed when it arrives at a location. Valid values for this field are Y and N. This field will always have data for the ITEMHDRCRE and ITEMHDRMOD records. | Character | 1 | itm_item | MEASURE_REQ_FLAG | NUMBER(1,0) | NA | If Y, then 1. |

Table B-4 (Cont.) RMS Item Header Mapping

| Po s | RMS Field Name | RMS Description | RMS Field Type | RMS Max Width | Xstore Table | Xstore Column | Xstore Data Type | Xstore Default Value | Xstore Comment |
|---------|------------------------------|--|----------------------|---------------------|-----------------|------------------|------------------------|----------------------------|-------------------|
| 73 | CatchW eightTy pe | This field holds the catch weight type for a simple pack catch weight item. The value is based on the component items order_type and sale_type. 2 - order_type = Variable Weight, sale_type = Loose Weight 4 - order_type = Variable Weight, sale_type = Variable Weight Each Data is optional in this field and will only exist for catch weight, simple pack items. | Character | 1 | NA | NA | NA | NA | Ignored. |
| 74 | CatchW eightOr derType | This field holds a code that indicates how catch weight items are ordered. Valid values are: F - Fixed weight V - Variable weight Data is optional in this field and will only exist for catch weight items. | Character | 6 | NA | NA | NA | NA | Ignored. |
| 75 | CatchW eightSal eType | This field holds a code that indicates how catch weight items are sold in store locations. Valid values are: V - variable weight each L - Loose weight Data is optional in this field and will only exist for catch weight items. | Character | 6 | NA | NA | NA | NA | Ignored. |
| 76 | CatchW eightU OM | UOM for Catchweight Items. Data is optional in this field and will only exist for catch weight items. | Character | 4 | NA | NA | NA | NA | Ignored. |

Table B-4 (Cont.) RMS Item Header Mapping

| Po s | RMS Field Name | RMS Description | RMS Field Type | RMS Max Width | Xstore Table | Xstore Column | Xstore Data Type | Xstore Default Value | Xstore Comment |
|---------|-------------------------|--|----------------------|---------------------|-----------------|------------------|------------------------|----------------------------|-------------------|
| 77 | Deposit ItemTyp e | <p>This is the deposit item component type.</p> <p>A NULL value in this field indicates that this item is not part of a deposit item relationship. Valid values include:</p> <p>E - Contents A - Container Z - Crate T - Returned Item (Empty bottle) P - Complex pack (with deposit items)</p> <p>The Returned Item is flagged only to enable these items to be mapped to a separate GL account if required.</p> <p>Data is optional in this field.</p> | Character | 6 | NA | NA | NA | NA | Ignored. |
| 78 | Contain erItem | <p>This field holds the container item number for a contents item. This field is only populated and required if the DepositItemType is E. It will hold an item number that has a DepositItemType of A.</p> <p>Data is optional in this field.</p> | Character | 25 | NA | NA | NA | NA | Ignored. |

Table B-4 (Cont.) RMS Item Header Mapping

| Po s | RMS Field Name | RMS Description | RMS Field Type | RMS Max Width | Xstore Table | Xstore Column | Xstore Data Type | Xstore Default Value | Xstore Comment |
|---------|----------------------------------|--|----------------------|---------------------|-----------------|------------------|------------------------|----------------------------|-------------------|
| 79 | Deposit InPrice PerOU M | This field indicates if the deposit amount is included in the price per UOM calculation for a contents item ticket. This value is only required if the DEPOSIT_ITEM_TYPE = E. Valid values are: I - Includes deposit amount E - Excludes deposit amount Data is optional in this field. | Character | 6 | NA | NA | NA | NA | Ignored. |
| 80 | SOHInq uiryAtP ackInd | This field indicates if stock on hand inquiries from downstream systems should be allowed at the pack level for this item. It defaults to N for non-pack items. This field will always have data for the ITEMHDRCRE and ITEMHDRMOD records. | Character | 1 | NA | NA | NA | NA | Ignored. |
| 81 | Notiona IPackIn d | If this indicator is Y, SIM and other downstream systems should track pack inventory at the pack level. If the indicator is N, SIM and other downstream systems track inventory at the component level. It defaults to N for non-pack items. This field will always have data for the ITEMHDRCRE and ITEMHDRMOD records. | Character | 1 | NA | NA | NA | NA | Ignored. |

Table B-4 (Cont.) RMS Item Header Mapping

| Po s | RMS Field Name | RMS Description | RMS Field Type | RMS Max Width | Xstore Table | Xstore Column | Xstore Data Type | Xstore Default Value | Xstore Comment |
|---------|----------------------|---|----------------------|---------------------|-----------------|------------------|---------------------------|----------------------------|---|
| 82 | Comments | Holds any free-form comments associated with the item. Data is optional in this field. | Character | 2000 | NA | NA | NA | NA | Ignored. |
| NA | NA | NA | NA | NA | itm_item | MERCH_LEVEL_1 | VARCHA R2(60 CHAR) | NA | Populated by looking up the parent of the merch_level_2 value in loc_org_hierarchy. |
| NA | NA | NA | NA | NA | itm_item | ITEM_TYPCODE | VARCHA R2(30 CHAR) | STANDARD | NA |
| NA | NA | NA | NA | NA | itm_item | DESCRIPTION | VARCHA R2(254 CHAR) | NA | Contains synthesized Translation key following the pattern: "+item:[ITEM ID]:description". Key maps to the com_translation record populated from RMS ItemLoc. |
| NA | NA | NA | NA | NA | itm_item | NAME | VARCHA R2(254 CHAR) | NA | Contains synthesized Translation key following pattern: "+item:[ITEM ID]:description". Key maps to com_translation record populated from RMS ItemLoc. |

Table B-4 (Cont.) RMS Item Header Mapping

| PoS | RMS Field Name | RMS Description | RMS Field Type | RMS Max Width | Xstore Table | Xstore Column | Xstore Data Type | Xstore Default Value | Xstore Comment |
|-----|----------------|-----------------|----------------|---------------|---|-----------------|-------------------|---------------------------------|--------------------------------|
| NA | NA | NA | NA | NA | itm_item, itm_item_dimension_value, itm_item_dimension_type | ORGANIZATION_ID | NUMBER(10,0) | \$(dtv.location.organizationId) | Always uses default value. |
| NA | NA | NA | NA | NA | itm_item, itm_item_dimension_value, itm_item_dimension_type | CREATE_DATE | TIMESTAMP(6) | now() | Not set when Type =ITEMHDRMOD. |
| NA | NA | NA | NA | NA | itm_item, itm_item_dimension_value, itm_item_dimension_type | CREATE_USER_ID | VARCHAR2(30 CHAR) | DATALOADER | Not set when Type =ITEMHDRMOD. |
| NA | NA | NA | NA | NA | itm_item, itm_item_dimension_value, itm_item_dimension_type | UPDATE_DATE | TIMESTAMP(6) | now() | Not set when Type =FULLHDR. |
| NA | NA | NA | NA | NA | itm_item, itm_item_dimension_value, itm_item_dimension_type | UPDATE_USER_ID | VARCHAR2(30 CHAR) | DATALOADER | Not set when Type =FULLHDR. |

RMS Item Location

Table B-5 describes the RMS Item Location mapping.

Table B-5 RMS Item Location Mapping

| Pos | RMS Field Name | RMS Description | RMS Field Type | RMS Max Width | Xstore Table | Xstore Column | Xstore Data Type | Xstore Default Value | Xstore Comment |
|------------|-----------------------|---|-----------------------|----------------------|---------------------|----------------------|-------------------------|-----------------------------|--------------------------------------|
| 0 | Family | Value is always ITEMLOC. Data will always be present in this field. | Character | 20 | NA | NA | NA | NA | Ignored. |
| 1 | Type | Value is always FULLITEMLOC when the program is run in full mode. In delta mode, valid values are: ITEMLOCCRE ITEMLOCMOD ITEMLOCDEL Data will always be present in this field. | Character | 15 | NA | NA | NA | NA | Determines action. |
| 2 | LocType | This field indicates whether the location is a Store or Warehouse. Valid values are: S - Store W - Warehouse E - External Finisher Data will always be present in this field. | Character | 1 | itm_item_options | LEVEL_CODE | VARCHAR2(30 CHAR) | STORE | If not S, entire record is ignored. |
| 3 | Location | This field holds the numeric ID of the store of WH. Data will always be present in this field. | Number | 10 | itm_item_options | LEVEL_VALUE | VARCHAR2(30 CHAR) | NA | NA |
| 4 | Item | ID of the item. Data will always be present in this field. | Character | 25 | itm_item_options | ITEM_ID | VARCHAR2(60 CHAR) | NA | NA |
| 5 | ItemParent | ID of the parent of the item. Item parent does not vary by location, but is included in this feed as a denormalized convenience for downstream systems. Data is optional in this field for many items. | Character | 25 | NA | NA | NA | NA | Ignored, obtained from ItemHdr feed. |

Table B-5 (Cont.) RMS Item Location Mapping

| Pos | RMS Field Name | RMS Description | RMS Field Type | RMS Max Width | Xstore Table | Xstore Column | Xstore Data Type | Xstore Default Value | Xstore Comment |
|------------|-----------------------|--|-----------------------|----------------------|---------------------|----------------------|-------------------------|-----------------------------|--------------------------------------|
| 6 | ItemGrandparent | ID of the grandparent of Item. Item grandparent does not vary by location, but is included in this feed as a denormalized convenience for downstream systems. Data is optional in this field for many items. | Character | 25 | NA | NA | NA | NA | Ignored, obtained from ItemHdr feed. |
| 7 | InitialUnitRetail | Initial Unit Retail of the item in the item's standard unit of measure. Data is optional in this field for some type of items. Data is only included in this field for the ITEMLOCCRE and FULLITEMLOC records. After initial item creation, unit retail and other pricing information should come from RPM. If this program is run in full mode and produces FULLITEMLOC records, a full extract from RPM of pricing should also be performed to update all price records. | NA | NA | NA | NA | NA | NA | Ignored. |

Table B-5 (Cont.) RMS Item Location Mapping

| Pos | RMS Field Name | RMS Description | RMS Field Type | RMS Max Width | Xstore Table | Xstore Column | Xstore Data Type | Xstore Default Value | Xstore Comment |
|-----|--------------------|---|----------------|---------------|------------------|----------------------|-------------------|----------------------|---|
| 8 | Selling UnitRetail | Data is optional in this field for some type of items. Data is only included in this field for the ITEMLOCCRE and FULLITEMLOC records. After initial item creation, unit retail and other pricing information should come from RPM. If this program is run in full mode and produces FULLITEMLOC records, a full extract from RPM of pricing should also be performed to update all price records. | Number | 20 | itm_item_prices | PRICE | NUMBER (17,6) | NA | When non-null, a Regular Price Event record will be created in itm_item_prices. |
| 9 | CurrencyCode | Currency Code of retail values. This field should be populated for the ITEMLOCCRE and FULLITEMLOC records that have InitialUnitRetail and SellingUnitRetail values. | NA | NA | NA | NA | NA | NA | Ignored. |
| 10 | Selling UOM | Data is optional in this field for some type of items. Data is only included in this field for the ITEMLOCCRE and FULLITEMLOC records. | Character | 4 | itm_item_options | UNIT_OF_MEASURE_CODE | VARCHAR2(30 CHAR) | NA | NA |
| 11 | TaxableInd | Indicates if the item is taxable at the location. This field will always have data for FULLITEMLOC, ITEMLOCCRE, and ITEMLOCCMOD. | Character | 1 | itm_item_options | TAX_GROUP_ID | VARCHAR2(60 CHAR) | 0 | If N, then 0, or configured value, (dataloader-beans.xml) |

Table B-5 (Cont.) RMS Item Location Mapping

| Pos | RMS Field Name | RMS Description | RMS Field Type | RMS Max Width | Xstore Table | Xstore Column | Xstore Data Type | Xstore Default Value | Xstore Comment |
|------------|-----------------------|---|-----------------------|----------------------|---------------------|----------------------|-------------------------|-----------------------------|--|
| 12 | LocalItemDesc | <p>Contains the local description of the item. This may be the same as the primary description of the item, a regional description of the item (jimmies versus sprinkles in the US or roll versus bap versus cob versus bun in the UK), or a value in a local language (Overlay dress - true black knit at US stores versus Lagenkleid - Strick, tiefschwarz at stores in Germany).</p> <p>The intent is that this string is appropriate to print this description on signage/receipts at this location.</p> <p>This field will always have data.</p> | Character | 250 | com_translations | TRANSLATION | VARCHAR2(4000 CHAR) | NA | If translated, ItemDescriptionsEnabled is true (see dataloader-beans.xml), and then com_translations record is created using ItemDesc. It is recommended that translated ItemDescriptionsEnabled be set to false when loading Xcenter because non-local descriptions are used in translation records when loading ItemHdr. |
| 13 | LocalShorDesc | <p>May contain an abbreviated local description of the item. This may be the same as the primary abbreviated description of the item, an abbreviation of regional description of the item, or an abbreviation in a local language.</p> <p>The intent is that this string is appropriate to print this description on signage/receipts at this location.</p> <p>Data is optional in this field.</p> | Character | 120 | NA | NA | NA | NA | Ignored. |

Table B-5 (Cont.) RMS Item Location Mapping

| P os | RMS Field Name | RMS Description | RMS Field Type | RMS Max Width | Xstore Table | Xstore Column | Xstore Data Type | Xstore Default Value | Xstore Comment |
|---------|----------------------------|---|----------------------|---------------------|-----------------|------------------|------------------------|----------------------------|-------------------|
| 14 | Ti | Number of shipping units (cases) that make up one tier of a pallet of this item for this location. A location may use a local pallet configuration if there are special space or receiving configurations that make typical pallets impractical. Multiply TI by HI to get the total number of cases for a pallet. Data is optional in this field. | Number | 12 | NA | NA | NA | NA | Ignored. |
| 15 | Hi | Number of tiers that make up a complete pallet (height). A location may use a local pallet configuration if there are special space or receiving configurations that make typical pallets impractical. Multiply TI by HI to get the total number of cases for a pallet. Data is optional in this field. | Number | 12 | NA | NA | NA | NA | Ignored. |
| 16 | StoreOr derMulti ple | Contains the multiple in which the item needs to be shipped from a warehouse to the location. This field will always have data for FULLITEMLOC, ITEMLOCCRE, and ITEMLOCMOD. | Character | 1 | NA | NA | NA | NA | Ignored. |

Table B-5 (Cont.) RMS Item Location Mapping

| Pos | RMS Field Name | RMS Description | RMS Field Type | RMS Max Width | Xstore Table | Xstore Column | Xstore Data Type | Xstore Default Value | Xstore Comment |
|------------|-----------------------|---|-----------------------|----------------------|---------------------|--------------------------------------|-------------------------------------|-----------------------------|--|
| 17 | Status | Current status of item at the location. Valid values are: A - Active, item is valid and can be ordered and sold. I - Inactive, item is valid, but cannot be ordered or sold. C - Discontinued, item is valid and sellable, but no longer orderable . D - Delete, item is pending delete and cannot be ordered or sold. This field will always have data for FULLITEMLOC, ITEMLOCCRE, and ITEMLOCMOD. | Character | 1 | itm_item_options | ITEM_AVAILABILITY_CODE, STOCK_STATUS | VARCHAR2(30 CHAR, VARCHAR2(60 CHAR) | NA | If itemLoc:status=A or C, the item_availability_code is Available. If itemLoc:status=I or D, the item_availability_code is NA. If itemLoc:status=C, the stock_status is DISCONTINUED. Otherwise, stock_status is null. |
| 18 | DailyWastePct | Average percentage lost from inventory on a daily basis due to natural wastage. Data is optional in this field. | Number | 12 | NA | NA | NA | NA | Ignored. |
| 19 | MeasureOfEach | Size of an each in terms of the uom_of_price, for example, 12 oz. This information can be used in ticketing and display to consumers. Data is optional in this field. | Number | 12 | NA | NA | NA | NA | Ignored. |
| 20 | MeasureOfPrice | Size to be used on the ticket in terms of the uom_of_price. For example, if the user wants the ticket to have the label print the price per ounce, this value would be 1. If the user wants the price per 100 grams, this value would be 100. Used in ticketing. Data is optional in this field. | Number | 12 | NA | NA | NA | NA | Ignored. |

Table B-5 (Cont.) RMS Item Location Mapping

| Pos | RMS Field Name | RMS Description | RMS Field Type | RMS Max Width | Xstore Table | Xstore Column | Xstore Data Type | Xstore Default Value | Xstore Comment |
|------------|-----------------------|---|-----------------------|----------------------|---------------------|----------------------|-------------------------|-----------------------------|-----------------------|
| 21 | UomOf Price | Unit of measure that is used on the ticket for this item. Data is optional in this field. | Character | 4 | NA | NA | NA | NA | Ignored. |
| 22 | Primary Variant | This field is used to address sales of PLUs (that is, above transaction level items) when inventory is tracked at a lower level (that is, UPC). This field only contains a value for items one level higher than the transaction level. Valid choices are any transaction level item that is a child of this item. In order to select a transaction level item as the primary variant, the item/location relationship must exist at the transaction level. When a transaction level item is specified as a primary variant for an item higher than the transaction level, an extra pos_mods record will be written. Both the transaction level item (that is, UPC) and the higher than transaction level item (that is, PLU) will be sent to the POS to allow the store to sell the PLU. The information sent for the PLU will be the same information sent for the transaction level item (that is, UPC). Data is optional in this field. | Character | 25 | NA | NA | NA | NA | Ignored. |

Table B-5 (Cont.) RMS Item Location Mapping

| Pos | RMS Field Name | RMS Description | RMS Field Type | RMS Max Width | Xstore Table | Xstore Column | Xstore Data Type | Xstore Default Value | Xstore Comment |
|------------|-----------------------|---|-----------------------|----------------------|---------------------|----------------------|-------------------------|-----------------------------|-----------------------|
| 23 | Primary CostPack | This field contains an item number that is a simple pack containing the item in the item column for this record. If populated, the cost of the future cost table will be driven from the simple pack and the deals and cost changes for the simple pack. Data is optional in this field. | Character | 25 | NA | NA | NA | NA | Ignored. |
| 24 | Primary Supplier | Numeric identifier of the supplier who will be considered the primary supplier for the specified item/loc. The supplier/origin country combination will determine the value of the unit cost field on item_loc. If the supplier is changed and ELC = N, the unit cost field on item_loc will be updated with the new supplier's cost. Data is optional in this field. | Number | 10 | NA | NA | NA | NA | Ignored. |
| 25 | Primary OriginCountry | Contains the identifier of the origin country which will be considered the primary country for the specified item/location. Data is optional in this field. | Character | 3 | NA | NA | NA | NA | Ignored. |

Table B-5 (Cont.) RMS Item Location Mapping

| Pos | RMS Field Name | RMS Description | RMS Field Type | RMS Max Width | Xstore Table | Xstore Column | Xstore Data Type | Xstore Default Value | Xstore Comment |
|------------|-----------------------|--|-----------------------|----------------------|---------------------|----------------------|-------------------------|-----------------------------|-----------------------|
| 26 | Receive AsType | This column determines whether the stock on hand for a pack component item or the buyer pack itself will be updated when a buyer pack is received at a warehouse. Valid values are Each or Pack. Data is optional in this field. | Character | 1 | NA | NA | NA | NA | Ignored. |
| 27 | InboundHandlingDays | This field indicates the number of inbound handling days for an item at a warehouse type location. Data is optional in this field. | Number | 2 | NA | NA | NA | NA | Ignored. |
| 28 | Source Method | This value is used to specify how the ad-hoc PO/TSF creation process should source the item/location request. If the value is Warehouse, the process will attempt to fill the request by creating a transfer from the warehouse in the source_wh field. If this warehouse does not have enough inventory to fill the request, a purchase order will be created for the item/location's primary supplier. For warehouses, it is used by Oracle Retail Allocation to determine the valid sources and destinations for warehouse to warehouse allocations. Data is optional in this field. | Character | 1 | NA | NA | NA | NA | Ignored. |

Table B-5 (Cont.) RMS Item Location Mapping

| Pos | RMS Field Name | RMS Description | RMS Field Type | RMS Max Width | Xstore Table | Xstore Column | Xstore Data Type | Xstore Default Value | Xstore Comment |
|------------|-----------------------|--|-----------------------|----------------------|---------------------|----------------------|-------------------------|-----------------------------|---|
| 29 | Source Wh | This value is used by the ad-hoc PO/Transfer creation process to determine from which warehouse to fill the stores request. It is also used by the Allocation process to support warehouse to warehouse allocations. A value is required in this field if the sourcing method is Warehouse. Data is optional in this field. | Number | 10 | NA | NA | NA | NA | Ignored. |
| 30 | UinType | This column contains the unique identification number (UIN) used to identify the instances of the item at the location. Data is optional in this field. | Character | 6 | NA | NA | NA | NA | Used to determine a Serialized Item. If empty, itm_serialized_item_flag is set to 0. Otherwise, itm_serialized_item_flag is set to 1. |
| 31 | UinLabel | This column contains the label for the UIN when displayed in SIM. Data is optional in this field. | Character | 6 | NA | NA | NA | NA | Ignored. |
| 32 | Capture TimeIn Proc | This column indicates when the UIN should be captured for an item during transaction processing. Data is optional in this field. | Character | 6 | NA | NA | NA | NA | Ignored. |

Table B-5 (Cont.) RMS Item Location Mapping

| Pos | RMS Field Name | RMS Description | RMS Field Type | RMS Max Width | Xstore Table | Xstore Column | Xstore Data Type | Xstore Default Value | Xstore Comment |
|------------|------------------------|---|-----------------------|----------------------|---------------------|----------------------|-------------------------|-----------------------------|-----------------------|
| 33 | ExtUinInd | This Yes/No indicator indicates if the UIN is being generated in the external system. This field will always have data for FULLITEMLOC, ITEMLOCCRE, and ITEMLOCMOD. | Character | 1 | NA | NA | NA | NA | Ignored. |
| 34 | IntentionallyRangedInd | Determines if the location is ranged intentionally by the user for replenishment/selling (Y) or incidentally ranged (N) by the RMS programs when the item is not ranged to a specific location on the transaction. This field will always have data for FULLITEMLOC, ITEMLOCCRE, and ITEMLOCMOD. | Character | 1 | NA | NA | NA | NA | Ignored. |
| 35 | CostingLocation | Numeric identifier of the costing location for the franchise store. This field may contain a store or a warehouse. Data is optional in this field. | Number | 10 | NA | NA | NA | NA | Ignored. |
| 36 | CostingLocType | This field holds the type of costing location in the CostingLocation field. Data is optional in this field. | Character | 1 | NA | NA | NA | NA | Ignored. |
| 37 | LaunchDate | Holds the date that the item should first be sold at the location; in the human readable format DD-MON-YYYY. Data is optional in this field. | Character | 12 | NA | NA | NA | NA | Ignored. |

Table B-5 (Cont.) RMS Item Location Mapping

| Pos | RMS Field Name | RMS Description | RMS Field Type | RMS Max Width | Xstore Table | Xstore Column | Xstore Data Type | Xstore Default Value | Xstore Comment |
|------------|-----------------------|---|-----------------------|----------------------|---------------------|--|-------------------------|-----------------------------|--|
| 38 | QtyKey Options | Determines whether the qty key on a POS should be used for this item at the location. Valid values include, but are not limited to: R - Required P - Prohibited O - Optional Data is optional in this field. | Character | 6 | itm_item_options | PROMPT_FOR_QUANTITY_FLAG, FORCE_QUANTITY_OF_ONE_FLAG | NUMBER (1,0) | NA | PROMPT_FOR_QUANTITY_FLAG - 1 if R, otherwise 0. FORCE_QUANTITY_OF_ONE_FLAG - 1 if P, otherwise 0. |
| 39 | Manual PriceEntry | Determines whether the price can/should be entered manually on a POS for this item at the location. Valid values include, but are not limited to: R - Required P - Prohibited O - Optional Data is optional in this field. | Character | 6 | itm_item_options | PROMPT_FOR_PRICE_FLAG | NUMBER (1,0) | NA | 1 if R, otherwise 0. |
| 40 | Deposit Code | Indicates if a deposit is associated with this item at the location. Deposits are not subtracted from the retail of an item uploaded to RMS, and so on. This kind of processing is the responsibility of the client and should occur before sales are sent to Sales Audit and RMS. Data is optional in this field. | Character | 6 | NA | NA | NA | NA | Ignored. |

Table B-5 (Cont.) RMS Item Location Mapping

| Pos | RMS Field Name | RMS Description | RMS Field Type | RMS Max Width | Xstore Table | Xstore Column | Xstore Data Type | Xstore Default Value | Xstore Comment |
|-----|---------------------|--|----------------|---------------|------------------|-------------------------|------------------|----------------------|----------------------|
| 41 | FoodStampInd | Indicates whether the item is approved for food stamps at the location. This indicator does not impact processing in Sales Audit and RMS. Data is optional in this field. | Character | 1 | itm_item_options | FOODSTAMP_ELIGIBLE_FLAG | NUMBER (1,0) | NA | 1 if Y, otherwise 0. |
| 42 | WicInd | Indicates whether the item is approved for WIC at the location. This indicator does not impact processing in Sales Audit and RMS. Data is optional in this field. | Character | 1 | NA | NA | NA | NA | Ignored. |
| 43 | ProportionalTarePct | Holds the value associated for the packaging in items sold by weight at the location. The proportional tare is the proportion of the total weight of a unit of an item that is packaging (that is, if the tare item is bulk candy, this is the proportional of the total weight of one piece of candy that is the candy wrapper). Data is optional in this field. | Number | 12 | NA | NA | NA | NA | Ignored. |

Table B-5 (Cont.) RMS Item Location Mapping

| Pos | RMS Field Name | RMS Description | RMS Field Type | RMS Max Width | Xstore Table | Xstore Column | Xstore Data Type | Xstore Default Value | Xstore Comment |
|------------|-----------------------|---|-----------------------|----------------------|---------------------|----------------------|-------------------------|-----------------------------|-----------------------|
| 44 | FixedTareValue | Holds the value associated for the packaging in items sold by weight at the location. Fixed tare is the tare of the packaging. This value is fixed, regardless of how much of the item is sold (that is, if the tare item is bulk candy, this is weight of the bag and twist tie). Data is optional in this field. | Number | 12 | NA | NA | NA | NA | Ignored. |
| 45 | FixedTareUom | Holds the unit of measure value associated with the tare value. Data is optional in this field. | Character | 4 | NA | NA | NA | NA | Ignored. |
| 46 | RewardEligibleItem | Holds whether the item is legally valid for various types of bonus point/award programs at the location. Data is optional in this field. | Character | 1 | NA | NA | NA | NA | Ignored. |
| 47 | NatlBrandCompareItem | Holds the nationally branded item to which the current item should be compared. Data is optional in this field. | Character | 25 | NA | NA | NA | NA | Ignored. |
| 48 | ReturnPolicy | Holds the return policy for the item at the location. Retailers may configure any number of return policies. Data is optional in this field. | Character | 6 | NA | NA | NA | NA | Ignored. |

Table B-5 (Cont.) RMS Item Location Mapping

| Pos | RMS Field Name | RMS Description | RMS Field Type | RMS Max Width | Xstore Table | Xstore Column | Xstore Data Type | Xstore Default Value | Xstore Comment |
|-----|--------------------------|---|----------------|---------------|------------------|------------------------|----------------------|----------------------|--|
| 49 | StopSaleInd | Indicates that sale of the item should be stopped immediately at the location (that is, in case of recall, and so on). Data is optional in this field. | Character | 1 | itm_item_options | ITEM_AVAILABILITY_CODE | VARCHAR2(30) CHAR | NA | If stopSaleInd=Y, item_availability_code is RECALL. Otherwise, see the itemLoc:status mapping. |
| 50 | ElectMktClub | Holds the code that represents the marketing clubs to which the item belongs at the location. Retailers may configure any number of marketing clubs. Data is optional in this field. | Character | 6 | NA | NA | NA | NA | Ignored. |
| 51 | Report Code | Code to determine which reports the location should run. Data is optional in this field. | Character | 6 | NA | NA | NA | NA | Ignored. |
| 52 | ReqShelfLifeOn Selection | Holds the required shelf life for an item on selection in days. This field is not required. Data is optional in this field. | Number | 4 | NA | NA | NA | NA | Ignored. |
| 53 | ReqShelfLifeOn Receipt | Holds the required shelf life for an item on receipt in days. Data is optional in this field. | Number | 4 | NA | NA | NA | NA | Ignored. |
| 54 | IBShelf Life | Holds the Investment Buy-specific shelf life for the item/location in days. Data is optional in this field. | Number | 4 | NA | NA | NA | NA | Ignored. |

Table B-5 (Cont.) RMS Item Location Mapping

| Pos | RMS Field Name | RMS Description | RMS Field Type | RMS Max Width | Xstore Table | Xstore Column | Xstore Data Type | Xstore Default Value | Xstore Comment |
|------------|-----------------------|---|-----------------------|----------------------|---------------------|----------------------|-------------------------|-----------------------------|-----------------------|
| 55 | StoreReorderableInd | Indicates whether the store may reorder the item. This field is required to be either Y - yes or N - no. The field defaults to N. Data is optional in this field. | Character | 1 | NA | NA | NA | NA | Ignored. |
| 56 | RackSize | Indicates the rack size that should be used for the item. This field is not required. Data is optional in this field. | Character | 6 | NA | NA | NA | NA | Ignored. |
| 57 | FullPalletItem | Indicates whether a store must reorder an item in full pallets only. This field is required to be either Y - yes or N - no. The field defaults to N. Data is optional in this field. | Character | 1 | NA | NA | NA | NA | Ignored. |
| 58 | InStoreMarketBasket | Holds the in-store market basket code for this item/location combination. Data is optional in this field. | Character | 6 | NA | NA | NA | NA | Ignored. |
| 59 | StorageLocation | Holds the current storage location or bin number for the item at the location. Data is optional in this field. | Character | 7 | NA | NA | NA | NA | Ignored. |
| 60 | AltStorageLocation | Holds the preferred alternate storage location or bin number for the item at the location. Data is optional in this field. | Character | 7 | NA | NA | NA | NA | Ignored. |

Table B-5 (Cont.) RMS Item Location Mapping

| Pos | RMS Field Name | RMS Description | RMS Field Type | RMS Max Width | Xstore Table | Xstore Column | Xstore Data Type | Xstore Default Value | Xstore Comment |
|-----|----------------|--|----------------|---------------|------------------|---------------------|------------------|----------------------|--|
| 61 | ReturnableInd | This field contains a value of Yes when the item can be returned to the location. Data is optional in this field. | Character | 1 | itm_item_options | NOT_RETURNABLE_FLAG | NUMBER (1,0) | NA | 1 if N, otherwise 0. |
| 62 | RefundableInd | This field contains a value of Yes when the item is refundable at the location. Data is optional in this field. | Character | 1 | NA | NA | NA | NA | Ignored. |
| 63 | BackOrderInd | This field contains a value of Yes when the item can be back-ordered to the location. Data is optional in this field. | Character | 1 | NA | NA | NA | NA | Ignored. |
| 64 | MerchandiseInd | Note that the MerchandiseInd does not vary by locations (items are not physical entities in one location and fees in another). The MerchandiseInd is denormalized to the item/location level because it is a common use case for some downstream systems to care only about physical or non-physical items. This field is denormalized to assist those systems in their filtering logic. All items, both merchandise and non-merchandise, are exported from RMS. This field will always have data for all records. | Character | 1 | NA | NA | NA | NA | If N, then NON_PHYSICAL item so record is skipped. |

Table B-5 (Cont.) RMS Item Location Mapping

| Pos | RMS Field Name | RMS Description | RMS Field Type | RMS Max Width | Xstore Table | Xstore Column | Xstore Data Type | Xstore Default Value | Xstore Comment |
|-----|----------------|--|----------------|---------------|----------------------------|-----------------|-------------------|---------------------------------|---------------------------------|
| 65 | ClearanceInd | A flag indicating if the item is in clearance price or regular price. "Y" for clearance price and "N" for regular price. | | | | | | | |
| | NA | NA | NA | NA | itm_item, itm_item_options | ORGANIZATION_ID | NUMBER(10,0) | \${dtv.location.organizationId} | Always uses default value. |
| | NA | NA | NA | NA | itm_item, itm_item_options | CREATE_DATE | TIMESTAMP(6) | now() | Not set when Type =ITEMLOC MOD. |
| | NA | NA | NA | NA | itm_item, itm_item_options | CREATE_USER_ID | VARCHAR2(30 CHAR) | DATALOADER | Not set when Type =ITEMLOC MOD. |
| | NA | NA | NA | NA | itm_item, itm_item_options | UPDATE_DATE | TIMESTAMP(6) | now() | Not set when Type =FULLITEMLOC. |
| | NA | NA | NA | NA | itm_item, itm_item_options | UPDATE_USER_ID | VARCHAR2(30 CHAR) | DATALOADER | Not set when Type =FULLITEMLOC. |

RMS Merchandise Hierarchy

Table B-6 describes the RMS Merchandise Hierarchy mapping.

Table B-6 RMS Merchandise Hierarchy Mapping

| Pos | RMS Field Name | RMS Description | RMS Field Type | RMS Max Width | Xstore Table | Xstore Column | Xstore Data Type | Xstore Default Value | Xstore Comment |
|-----|----------------|---------------------------------|----------------|---------------|---------------------|---------------|------------------|----------------------|---|
| 0 | Family | Value is always MERCHHIERARCHY. | Character | 20 | ITM_MERCH_HIERARCHY | NA | NA | NA | No need to store this information. This information is only consumed by the transformer to determine the feed type. |

Table B-6 (Cont.) RMS Merchandise Hierarchy Mapping

| Pos | RMS Field Name | RMS Description | RMS Field Type | RMS Max Width | Xstore Table | Xstore Column | Xstore Data Type | Xstore Default Value | Xstore Comment |
|-----|----------------|--|----------------|---------------|---------------------|---------------|------------------|----------------------|---|
| 1 | Type | <p>Value is always FULL when the program is run in full mode.</p> <p>If the program is run in delta mode, the type will vary, and be composed of the entity (DIVISION, GROUP, DEPT, CLASS, SUBCLASS) and the type of change (CRE, MOD, DEL). Examples include:</p> <p>DIVISIONCRE DIVISIONMOD DIVISIONDEL GROUPCR GROUPMOD GROUPDEL DEPTCRE DEPTMOD DEPTDEL CLASSCRE CLASSMOD CLASSDEL SUBCLASSCRE SUBCLASSMOD SUBCLASSDEL</p> <p>Note: Deleting a node of the merchandise hierarchy is unusual, and can only be done if no items are associated with the node.</p> | Character | 15 | ITM_MERCH_HEIRARCHY | NA | NA | NA | No need to store this information. This information is only consumed by the transformer to determine the action type. |

Table B-6 (Cont.) RMS Merchandise Hierarchy Mapping

| Pos | RMS Field Name | RMS Description | RMS Field Type | RMS Max Width | Xstore Table | Xstore Column | Xstore Data Type | Xstore Default Value | Xstore Comment |
|------------|-----------------------|--|-----------------------|----------------------|---------------------|----------------------|-------------------------|-----------------------------|---|
| 2 | HierarchyLevel | Value is always DIVISION, GROUP, DEPT, CLASS, SUBCLASS. This field cannot be null. This information identifies the level of the merchandise hierarchy that is described by this record. It is not possible to modify a HierarchyLevel on an existing record. | Character | 10 | ITM_MERCH_HEIRARCHY | level_code | VARCHAR2(30 CHAR) | NA | level |
| 3 | HierarchyNodeId | HierarchyNodeId is only unique within a HierarchyLevel (meaning it is possible, for example, that there is both a DIVISION 1 and a GROUP 1 in the full merchandise hierarchy). It is not possible to modify HierarchyNodeId on an existing record. | Number | 10 | ITM_MERCH_HEIRARCHY | heirarchy_id | VARCHAR2(60 CHAR) | NA | heirarchy id+first letter of hierarchy level. |
| 4 | HierarchyNodeName | Name of the organizational hierarchy entity. This field cannot be null. Description data is only sent in the primary integration language of the system. HierarchyNodename can be modified. | Character | 150 | ITM_MERCH_HEIRARCHY | description | VARCHAR2(254 CHAR) | NA | node name |

Table B-6 (Cont.) RMS Merchandise Hierarchy Mapping

| Pos | RMS Field Name | RMS Description | RMS Field Type | RMS Max Width | Xstore Table | Xstore Column | Xstore Data Type | Xstore Default Value | Xstore Comment |
|------------|-----------------------|---|-----------------------|----------------------|---------------------|----------------------|-------------------------|-----------------------------|--|
| 5 | ParentLevel | Level of the organizational hierarchy above the current node. Both ParentLevel and ParentId should be evaluated to correctly traverse the hierarchy. It is not possible to modify ParentLevel on an existing record. | Character | 10 | ITM_MERCH_HEIRARCHY | NA | NA | NA | This ID is used to validate the parent child relationship. It is not persisted into the database, but plays a key role in the parent child relationship validation of the merchandise hierarchy. |
| 6 | ParentId | ID of the level of the organizational hierarchy above the current node. Both ParentLevel and ParentId should be evaluated to correctly traverse the hierarchy. ParentId can be modified, meaning it is possible to change the division to which a group belongs. | Number | 10 | ITM_MERCH_HEIRARCHY | parent_id | VARCHAR2(60 CHAR) | NA | parent heirarchy id |

Table B-6 (Cont.) RMS Merchandise Hierarchy Mapping

| Pos | RMS Field Name | RMS Description | RMS Field Type | RMS Max Width | Xstore Table | Xstore Column | Xstore Data Type | Xstore Default Value | Xstore Comment |
|------------|---------------------------|--|-----------------------|----------------------|---------------------|----------------------|-------------------------|-----------------------------|-----------------------|
| 7 | GrandparentMerchDisplayId | <p>Only populated for SUBCLASS entities. For subclasses, this column holds the department ID used for display purposes in RMS (department is the grandparent of subclass).</p> <p>Note that in RMS, dept, class, and subclass display IDs are combined to form a composite unique key. Every department can have a class 1. Every class in Department 1000 can have a subclass 1. Looking only at the display IDs, all three values are required for uniqueness.</p> | Number | 4 | NA | NA | NA | NA | NA |
| 8 | parentMerchDisplayId | <p>Only populated for CLASS and SUBCLASS entities.</p> <p>For classes, this column holds the department ID used for display purposes in RMS (department is the parent of class).</p> <p>For subclasses, this column holds the class ID used for display purposes in RMS.</p> <p>Note that in RMS, dept, class, and subclass display IDs are combined to form a composite unique key. Every department can have a class 1. Every class in Department 1000 can have a subclass 1. Looking only at the display IDs, all three values are required for uniqueness.</p> | Number | 4 | NA | NA | NA | NA | NA |

Table B-6 (Cont.) RMS Merchandise Hierarchy Mapping

| Pos | RMS Field Name | RMS Description | RMS Field Type | RMS Max Width | Xstore Table | Xstore Column | Xstore Data Type | Xstore Default Value | Xstore Comment |
|------------|-----------------------|--|-----------------------|----------------------|---------------------|----------------------|-------------------------|-----------------------------|-----------------------|
| 9 | merchDisplayId | <p>Only populated for DEPARTMENT, CLASS, and SUBCLASS entities.</p> <p>For departments, this column holds the department display ID.</p> <p>For classes, this column holds the class display ID.</p> <p>For subclasses, this column holds the subclass display ID.</p> <p>Note that in RMS, dept, class, and subclass display IDs are combined to form a composite unique key. Every department can have a class 1. Every class in Department 1000 can have a subclass 1. Looking only at the display IDs, all three values are required for uniqueness.</p> | Number | 4 | NA | NA | NA | NA | NA |
| | NA | NA | NA | NA | ITM_MERCH_HEIRARCHY | CREATE_DATE | TIMESTAMP(6) | NA | NA |
| | NA | NA | NA | NA | ITM_MERCH_HEIRARCHY | CREATE_USER_ID | VARCHAR2(30 CHAR) | NA | NA |
| | NA | NA | NA | NA | ITM_MERCH_HEIRARCHY | UPDATE_DATE | TIMESTAMP(6) | NA | NA |
| | NA | NA | NA | NA | ITM_MERCH_HEIRARCHY | UPDATE_USER_ID | VARCHAR2(30 CHAR) | NA | NA |

RMS Organizational Hierarchy

Table B-7 describes the RMS Organizational Hierarchy mapping.

Table B-7 RMS Organizational Hierarchy Mapping

| P o s | RMS Field Name | RMS Description | RMS Field Type | RMS Max Width | Xstore Table | Xstore Column | Xstore Data Type | Xstore Default Value | Xstore Comment |
|----------------------|-------------------------------|---|-------------------------------|------------------------------|-------------------------|--------------------------|---------------------------------|-------------------------------------|---------------------------|
| 0 | Family | Value is always ORGHIERARCHY. | Character | 20 | NA | NA | NA | NA | Ignored. |
| 1 | Type | <p>Value is always FULL when the program is run in full mode.</p> <p>If the program is run in delta mode, the type will vary, and be composed of the entity (COMP, CHAIN, AREA, REGION, DISTRICT, STORE, WH) and the type of change (CRE, MOD, DEL). Examples include:</p> <p>COMP CRE - Creation of a Company node.</p> <p>COMP MOD - Modification of a Company node.</p> <p>CHAIN CRE - Creation of Chain node.</p> <p>CHAIN MOD - Modification of a Chain node.</p> <p>CHAIN DEL - Delete of Chain node.</p> <p>AREA CRE - Creation of an Area node.</p> <p>AREA MOD - Modification of Area node.</p> <p>AREA DEL - Delete of Area node.</p> <p>REGION CRE - Creation of Region node.</p> <p>REGION MOD - Modification of Region node.</p> <p>REGION DEL - Delete of Region node.</p> <p>DISTRICT CRE - Creation of District node.</p> | Character | 15 | NA | NA | NA | NA | Determines action. |

Table B-7 (Cont.) RMS Organizational Hierarchy Mapping

| P o s | RMS Field Name | RMS Description | RMS Field Type | RMS Max Width | Xstore Table | Xstore Column | Xstore Data Type | Xstore Default Value | Xstore Comment |
|----------------------|-------------------------------|--|-------------------------------|------------------------------|-------------------------|--------------------------|---------------------------------|-------------------------------------|---|
| | Type (Continued) | <p>DISTRICTMOD - Modification of District node.</p> <p>DISTRICTDEL - Delete of a District node.</p> <p>STORECRE - Creation of a Store node.</p> <p>STOREMOD - Modification of Store node.</p> <p>STOREDEL - Delete of Store node.</p> <p>WHCRE - Creation of WH node.</p> <p>WHMOD - Modification of WH node.</p> <p>WHDEL - Delete of WH node.</p> <p>Note: Deletion of Company node is not allowed.</p> | NA | NA | NA | NA | NA | NA | NA |
| 2 | HierarchyLevel | <p>Value is always COMPANY, CHAIN, AREA, REGION, DISTRICT, STORE, or WAREHOUSE. This field cannot be null.</p> <p>This information identifies the level of the organizational hierarchy that is described by this record.</p> <p>It is not possible to modify a HierarchyLevel on an existing record.</p> | Character | 10 | loc_org_hierarchy | org_code | VARCHAR2(30 CHAR) | NA | For the root COMPANY node, the column is hard-coded as "*". |

Table B-7 (Cont.) RMS Organizational Hierarchy Mapping

| P o s | RMS Field Name | RMS Description | RMS Field Type | RMS Max Width | Xstore Table | Xstore Column | Xstore Data Type | Xstore Default Value | Xstore Comment |
|----------------------|-------------------------------|---|-------------------------------|------------------------------|-------------------------|--------------------------|---------------------------------|-------------------------------------|---|
| 3 | HierarchyNode Id | <p>This information identifies the node of the organizational hierarchy that is described by this record. This field cannot be null.</p> <p>HierarchyNodeid is only unique within a HierarchyLevel (meaning it is possible, for example, that there is both a DISTRICT 1 and a REGION 1 in the full organizational hierarchy).</p> <p>It is not possible to modify HierarchyNodeid on an existing record.</p> | Number | 10 | loc_org_ hierarchy | org_code | VARCHA R2(60 CHAR) | NA | For the root COMPANY node, the column is hard-coded as "6". |
| 4 | HierarchyNode Name | <p>Name of the organizational hierarchy entity. This field cannot be null.</p> <p>Description data is only sent in the primary integration language of the system.</p> <p>HierarchyNodenam e can be modified.</p> | Character | 150 | loc_org_ hierarchy | descriptio n | VARCHA R2(254 CHAR) | NA | NA |

Table B-7 (Cont.) RMS Organizational Hierarchy Mapping

| P o s | RMS Field Name | RMS Description | RMS Field Type | RMS Max Width | Xstore Table | Xstore Column | Xstore Data Type | Xstore Default Value | Xstore Comment |
|----------------------|-------------------------------|--|-------------------------------|------------------------------|-------------------------|--------------------------|---------------------------------|-------------------------------------|--|
| 5 | ParentLevel | <p>Level of the organizational hierarchy above the current node. Both ParentLevel and ParentId should be evaluated to correctly traverse the hierarchy.</p> <p>ParentLevel is null for the COMPANY, but will exist for all other hierarchy levels. Warehouses always have the COMPANY node as their ParentLevel.</p> <p>It is not possible to modify ParentLevel on an existing record.</p> | Character | 10 | loc_org_hierarchy | parent_code | VARCHAR2(30 CHAR) | NA | <p>For the root COMPANY node, the column is hard-coded as null.</p> <p>For the second level CHAIN node, the column is hard-coded to "*".</p> |
| 6 | ParentId | <p>ID of the level of the organizational hierarchy above the current node. Both ParentLevel and ParentId should be evaluated to correctly traverse the hierarchy.</p> <p>ParentId is null for the COMPANY, but will exist for all other hierarchy levels. Warehouses always have the COMPANY node ID as their ParentId.</p> <p>ParentId can be modified, meaning it is possible to change the REGION ID that a DISTRICT ID belongs to (but a DISTRICT will always belong to a REGION as it is not possible to change the ParentLevel).</p> | Number | 10 | loc_org_hierarchy | parent_value | VARCHAR2(60 CHAR) | NA | <p>For the root COMPANY node, the column is hard-coded as null.</p> <p>For the second level CHAIN node, the column is hard-coded to "*".</p> |

Table B-7 (Cont.) RMS Organizational Hierarchy Mapping

| P o s | RMS Field Name | RMS Description | RMS Field Type | RMS Max Width | Xstore Table | Xstore Column | Xstore Data Type | Xstore Default Value | Xstore Comment |
|----------------------|-------------------------------|--|-------------------------------|------------------------------|-------------------------|--------------------------|---------------------------------|-------------------------------------|--|
| 7 | MgrName | <p>Manager of the current node of the hierarchy.</p> <p>MgrName is null for the COMPANY, and is optional for CHAIN, AREA, REGION, DISTRICT, and WAREHOUSE. It will always exist for STORE in Cre and Mod records.</p> <p>MrgName can be modified.</p> | Number | 10 | loc_org_hierarchy | level_mgr | VARCHAR2(254 CHAR) | NA | NA |
| 8 | CurrencyCode | <p>Currency of the current node of the hierarchy.</p> <p>CurrencyCode is null for the COMPANY and optional for CHAIN, AREA, REGION, DISTRICT, and WAREHOUSE. It will always exist for STORE in Cre and Mod records.</p> <p>CurrencyCode can be modified for a CHAIN, AREA, REGION, and DISTRICT. It cannot be modified for a STORE or Warehouse.</p> | Character | 3 | NA | NA | NA | NA | Ignored. |
| NA | NA | NA | NA | NA | loc_org_hierarchy | level_order | NUMBER (10,0) | NA | 0 - COMPANY 10 - CHAIN 20 - AREA 30 - REGION 40 - DISTRICT 1000 - STORE 2000 - Warehouse |
| NA | NA | NA | NA | NA | loc_org_hierarchy | sort_order | NUMBER (10,0) | NA | Always set to 0. |

Table B-7 (Cont.) RMS Organizational Hierarchy Mapping

| P o s | RMS Field Name | RMS Description | RMS Field Type | RMS Max Width | Xstore Table | Xstore Column | Xstore Data Type | Xstore Default Value | Xstore Comment |
|-------------|----------------------|--------------------|----------------------|---------------------|-----------------------|-------------------------|--------------------------|--|---|
| NA | NA | NA | NA | NA | loc_org_ hierarchy | inactive_ flag | NUMBER (1,0) | NA | Use default 0. |
| NA | NA | NA | NA | NA | loc_org_ hierarchy | ORGANI ZATION_ ID | NUMBER (10,0) | #{dtv.loc ation.or ganizati onId} | Always uses default value. |
| NA | NA | NA | NA | NA | loc_org_ hierarchy | CREATE_ DATE | TIMESTA MP(6) | now() | Not set when Type represents UPDATE. |
| NA | NA | NA | NA | NA | loc_org_ hierarchy | UPDATE_ DATE | TIMESTA MP(6) | now() | Not set when Type represents CREATE. |
| NA | NA | NA | NA | NA | loc_rtl_ loc | UPDATE_ USER_ ID | VARCHA R2(30 CHAR) | DATAL OADER | Always set. |

RMS Related Item Detail

Table B-8 describes the RMS Related Item Detail mapping.

Table B-8 RMS Related Item Detail Mapping

| P o s | RMS Field Name | RMS Description | RMS Field Type | RMS Max Width | Xstore Table | Xstore Column | Xstore Data Type | Xstore Default Value | Xstore Comment |
|-------------|----------------------|---|----------------------|---------------------|-----------------|------------------|------------------------|----------------------------|---|
| 0 | Family | Value is always ITEMS. Data will always be present in this field. | Character | 20 | NA | NA | NA | NA | Ignored. |
| 1 | Type | Value is always FULLRELITEMDET when the program is run in full mode. If the program is run in delta mode, the type will vary based on the type of change (CRE, MOD, DEL). Valid values are: RELITEMDETCRE RELITEMDETMOD RELITEMDETDEL Data will always be present in this field. | Character | 15 | NA | NA | NA | NA | Determines Action. Relationshi pType is looked up from rms_ related_ item_head by Relationshi pId, Location, and Organizatio nId. |

Table B-8 (Cont.) RMS Related Item Detail Mapping

| P o s | RMS Field Name | RMS Description | RMS Field Type | RMS Max Width | Xstore Table | Xstore Column | Xstore Data Type | Xstore Default Value | Xstore Comment |
|----------------------|-------------------------------|---|-------------------------------|------------------------------|---|---|---------------------------------|-------------------------------------|---------------------------|
| 2 | Relation shipID | Unique identifier for each relationship header. Data will always be present in this field. | Number | 10 | itm_ substitute _items, itm_ attached_ items | EXTERN AL_ID | VARCHA R2(60 CHAR) | NA | NA |
| 3 | RelatedI tem | Item ID of the related item Data will always be present in this field. | Character | 25 | itm_ substitute _items, itm_ attached_ items | ATTACH ED_ ITEM_ID, SUBSTIT UTE_ ITEM_ID | VARCHA R2(60 CHAR) | NA | NA |
| 4 | Locatio n | For corporate level files, this field holds the string CORPORATE. In location specific files, this field holds the numeric ID of the store of WH. Data will always be present in this field. | Character | 10 | itm_ substitute _items, itm_ attached_ items | LEVEL_ VALUE | VARCHA R2(60 CHAR) | NA | NA |
| 5 | Priority | Priority is only defined if the relationship is of the type SUBS at the related item header level. In the case of multiple related substitute items, this column could be used (optionally) to define relative priority. If there are multiple substitute items, it is possible not to define a priority, in which case all possible substitutions have the same relative priority. Data is optional in this field. | Number | 4 | NA | NA | NA | NA | Ignored. |

Table B-8 (Cont.) RMS Related Item Detail Mapping

| P o s | RMS Field Name | RMS Description | RMS Field Type | RMS Max Width | Xstore Table | Xstore Column | Xstore Data Type | Xstore Default Value | Xstore Comment |
|-------------|----------------------|---|----------------------|---------------------|---|------------------|------------------------|---------------------------------|--------------------------------------|
| 6 | StartDate | Optional start date that the item should be used in the relationship in DD-MON-YYYY format. If defined, this date indicates the first date the item should be offered as an upsell, substitute, or cross sell. Data is optional in this field. | Character | 11 | itm_substitute_items, itm_attached_items | BEGIN_DATETIME | TIMESTAMP(6) | NA | NA |
| 7 | EndDate | Optional end date that the item should be used in the relationship; in DD-MON-YYYY format. If defined, this date indicates the last date the item should be offered as an upsell, substitute or cross sell. Data is optional in this field. | Character | 11 | itm_substitute_items, itm_attached_items | END_DATETIME | TIMESTAMP(6) | NA | NA |
| NA | NA | NA | NA | NA | itm_substitute_items, itm_attached_items | ORGANIZATION_ID | NUMBER(10,0) | \${dtv.location.organizationId} | Always uses default value. |
| NA | NA | NA | NA | NA | itm_substitute_items, itm_attached_items | CREATE_DATE | TIMESTAMP(6) | now() | Not set when Type =RELITEM DETMOD. |
| NA | NA | NA | NA | NA | itm_substitute_items, itm_attached_items | CREATE_USER_ID | VARCHAR2(30 CHAR) | DATALOADER | Not set when Type =RELITEM DETMOD. |
| NA | NA | NA | NA | NA | itm_substitute_items, itm_attached_items | UPDATE_DATE | TIMESTAMP(6) | now() | Not set when Type =FULLRELI ITEMDET. |

Table B-8 (Cont.) RMS Related Item Detail Mapping

| P o s | RMS Field Name | RMS Description | RMS Field Type | RMS Max Width | Xstore Table | Xstore Column | Xstore Data Type | Xstore Default Value | Xstore Comment |
|-------------|----------------------|--------------------|----------------------|---------------------|---|---------------------------------------|---------------------------|---|--|
| NA | NA | NA | NA | NA | itm_ substitute_ items, itm_ attached_ items | UPDATE _USER_ ID | VARCHA R2(30 CHAR) | DATAL OADER | Not set when Type =FULLREL ITEMDET. |
| NA | NA | NA | NA | NA | itm_ attached_ items | PROMPT _TO_ ADD_ FLAG | NUMBER (1,0) | TRUE | Configurab le in dataloader- beans.xml. |
| NA | NA | NA | NA | NA | itm_ attached_ items | PROMPT _TO_ ADD_ MSG_ KEY | VARCHA R2(254 CHAR) | _ commo nAttach edItems Prompt | Configurab le in dataloader- beans.xml. |

RMS Related Item Header

Table B-9 describes the RMS Related Item Header mapping.

Table B-9 RMS Related Item Header Mapping

| P o s | RMS Field Name | RMS Description | RMS Field Type | RMS Max Width | Xstore Table | Xstore Column | Xstore Data Type | Xstore Default Value | Xstore Comment |
|-------------|----------------------|--|----------------------|---------------------|-----------------|------------------|------------------------|----------------------------|--|
| 0 | Family | Value is always ITEMS. Data will always be present in this field. | Character | 20 | NA | NA | NA | NA | Ignored. |
| 1 | Type | Value is always FULLRELITEMHDR when the program is run in full mode. If the program is run in delta mode, the type will vary based on the type of change (CRE, MOD, DEL). Valid values are: RELITEMHEADCR E RELITEMHEADM OD RELITEMHEADDE L Data will always be present in this field. | Character | 15 | NA | NA | NA | NA | Determines action. MOD will move records between itm_ substute_ items and itm_ attached_ items when the modified record represents a relationship type change between CRSL/UPS L and SUBS. |

Table B-9 (Cont.) RMS Related Item Header Mapping

| P o s | RMS Field Name | RMS Description | RMS Field Type | RMS Max Width | Xstore Table | Xstore Column | Xstore Data Type | Xstore Default Value | Xstore Comment |
|-------------|----------------------|---|----------------------|---------------------|-----------------------------------|---------------------------|---------------------------|----------------------------|--|
| 2 | Relation shipID | Unique identifier for each relationship header. Data will always be present in this field. | Number | 20 | rms_ related_ item_ head | RELATIO NSHIP_ ID | NUMBER (20,0) | NA | Used as externaId in itm_ substute_ items and itm_ attached_ items records. |
| 3 | Item | Item for which the relationships are defined. Data will always be present in this field. | Character | 25 | rms_ related_ item_ head | ITEM | VARCHA R2(25 CHAR) | NA | When SUBS, this is the primaryIte m. When CRSL/UPS L, this is the soldItem. |
| 4 | Locatio n | For corporate level files, this field holds the string CORPORATE. In location specific files, this field holds the numeric ID of the store of WH. Data will always be present in this field. | Character | 10 | rms_ related_ item_ head | LOCATI ON | VARCHA R2(10 CHAR) | NA | Used as levelValue in itm_ substute_ items and itm_ attached_ items records. |
| 5 | Relation shipName | Name given to the relationship. Data will always be present in this field for the RELITEMHEADCR E and RELITEMHEADM OD records, but will not be present for RELITEMHEADDE L records. | Character | 255 | rms_ related_ item_ head | RELATIO NSHIP_ NAME | VARCHA R2(255 CHAR) | NA | Ignored. |

Table B-9 (Cont.) RMS Related Item Header Mapping

| P o s | RMS Field Name | RMS Description | RMS Field Type | RMS Max Width | Xstore Table | Xstore Column | Xstore Data Type | Xstore Default Value | Xstore Comment |
|----------------------|-------------------------------|--|-------------------------------|------------------------------|-----------------------------------|---------------------------|---------------------------------|-------------------------------------|--|
| 6 | Relation shipType | Describes the type of relationship. Valid values are: CRSL Cross Sell SUBS Substitution UPSL Up Sell Data will always be present in this field for the RELITEMHEADCR E and RELITEMHEADM OD records, but will not be present for RELITEMHEADDE L records. | Character | 6 | rms_ related_ item_ head | RELATIO NSHIP_ TYPE | VARCHA R2(6 CHAR) | NA | Determines the .type of Xstore related item. SUBS = Xstore Substitute Items, CRSL/UPS L = Xstore Attached Items. |

Table B-9 (Cont.) RMS Related Item Header Mapping

| P o s | RMS Field Name | RMS Description | RMS Field Type | RMS Max Width | Xstore Table | Xstore Column | Xstore Data Type | Xstore Default Value | Xstore Comment |
|-------------|----------------------|---|----------------------|---------------------|-----------------------------------|-----------------------|-------------------------|----------------------------|---------------------------------|
| 7 | MandatoryInd | <p>This field indicates whether the relationship should be mandatory.</p> <p>For example, an item such as a laptop may have a mandatory cross sell relationship. The related items could be power cords for the US, UK, Mainland Europe, India, and so on. When the laptop is sold, it should be mandatory that one of the related power cords also be selected. Note that Merchandising/Sales Audit do not validate that the mandatory related item is also sold.</p> <p>Generally, only cross sell relationships are mandatory. Substitution and upsell relationships can be defined as mandatory, but in those cases, the definition of mandatory is at the discretion of the client and generally means that substitution or upsell must, as business process, be offered to consumers.</p> <p>Data will always be present in this field for the RELITEMHEADCRE and RELITEMHEADMOD records, but will not be present for RELITEMHEADDEL records.</p> | Character | 1 | rms_ related_ item_ head | MANDA TORY_ IND | VARCHA R2(1 CHAR) | NA | No logic is based on this data. |

Table B–9 (Cont.) RMS Related Item Header Mapping

| P o s | RMS Field Name | RMS Description | RMS Field Type | RMS Max Width | Xstore Table | Xstore Column | Xstore Data Type | Xstore Default Value | Xstore Comment |
|----------------------|-------------------------------|----------------------------|-------------------------------|------------------------------|-----------------------------------|---------------------------------------|---------------------------------|--|---|
| NA | NA | NA | NA | NA | rms_ related_ item_ head | ORGANI ZATION_ ID | NUMBER (10,0) | #{dtv.loc ation.or organizati onId} | Always uses default value. |
| NA | NA | NA | NA | NA | rms_ related_ item_ head | CREATE_ DATE | TIMESTA MP(6) | now() | Not set when Type =ITEMHD RMOD. |
| NA | NA | NA | NA | NA | rms_ related_ item_ head | UPDATE _DATE | TIMESTA MP(6) | now() | Not set when Type =FULLREL ITEMHDR |
| NA | NA | NA | NA | NA | item_ item_ options | ATTACH ED_ ITEMS_ FLAG | NUMBER (1,0) | NA | Set to 1 when CRSL or UPSL, otherwise 0 for Item. |
| NA | NA | NA | NA | NA | item_ item_ options | SUBSTIT UTE_ AVAILAB LE_FLAG | NUMBER (1,0) | NA | Set to 1 when SUBS, otherwise 0 for Item. |

RMS Store

Table B–10 describes the RMS Store mapping.

Table B–10 RMS Store Mapping

| Po s | RMS Field Name | RMS Description | RMS Field Type | RMS Max Width | Xstore Table | Xstore Column | Xstore Data Type | Xstore Default Value | Xstore Comment |
|-----------------|-------------------------------|---|-------------------------------|------------------------------|-------------------------|--------------------------|---------------------------------|-------------------------------------|---------------------------|
| 0 | Family | Value is always STORES. Data will always be present in this field. | Character | 20 | NA | NA | NA | NA | Ignored. |

Table B-10 (Cont.) RMS Store Mapping

| Po s | RMS Field Name | RMS Description | RMS Field Type | RMS Max Width | Xstore Table | Xstore Column | Xstore Data Type | Xstore Default Value | Xstore Comment |
|---------|----------------------|---|----------------------|---------------------|--|----------------------------------|------------------------|----------------------------|--------------------|
| 1 | Type | Value is always FULL when the program is run in full mode. If the program is run in delta mode, the type will vary based on the type of change (CRE, MOD, DEL). Valid values are: STORECRE - Creation of an Area node STOREMOD - Modification of Area node STOREDEL - Delete of Area node Data will always be present in this field. | Character | 15 | NA | NA | NA | NA | Determines action. |
| 2 | StoreId | Contains the unique ID of the store. Data will always be present in this field. | Number | 10 | loc_rtl_ loc, tax_ rtl_loc_ tax_ mapping | STORE_ NBR, RTL_ LOC_ID | NUMBER (10,0) | NA | NA |
| 3 | StoreNa me | Contains the full name of the store. Data will always be present in this field for creation and modification records. | Character | 150 | loc_rtl_ loc | DESCRIP TION | VARCHA R(254) | NA | NA |
| 4 | StoreNa me10 | Contains a shortened, no more than 10 character name, for the store. Data will always be present in this field for creation and modification records. | Character | 10 | loc_rtl_ loc | STORE_ NAME | VARCHA R(254) | NA | NA |
| 5 | StoreNa meAbbr | Contains an abbreviation for the store name. Data will always be present in this field for creation and modification records. | Character | 3 | NA | NA | NA | NA | Ignored. |
| 6 | StoreNa meSec | Contains an secondary name for this store. Data in this field is optional. | Character | 150 | NA | NA | NA | NA | Ignored. |

Table B-10 (Cont.) RMS Store Mapping

| Po s | RMS Field Name | RMS Description | RMS Field Type | RMS Max Width | Xstore Table | Xstore Column | Xstore Data Type | Xstore Default Value | Xstore Comment |
|---------|----------------------|---|----------------------|---------------------|-----------------|-----------------------|------------------------|----------------------------|-------------------|
| 7 | StoreCl assId | Contains the code letter indicating the class of which the store is a member. Valid values are A, B, C, D, E, and X. Data will always be present in this field for creation and modification records. | Character | 1 | NA | NA | NA | NA | Ignored. |
| 8 | StoreCl assDesc | Contains the description of the store class. Data will always be present in this field for creation and modification records. | Character | 250 | NA | NA | NA | NA | Ignored. |
| 9 | Manage r | Contains the name of the store manager. Data will always be present in this field. for creation and modification records. | Character | 120 | loc_rtl_ loc | STORE_ MANAG ER | VARCHA R(254) | NA | NA |
| 10 | OpenD ate | Contains the date on which the store opened. Data will always be present in this field for creation and modification records. Format is YYYYMMDD. | Date | 8 | NA | NA | NA | NA | Ignored. |
| 11 | CloseD ate | Contains the date on which the store closed. Data in this field is optional. Format is YYYYMMDD. | Date | 8 | NA | NA | NA | NA | Ignored. |
| 12 | Aquire Date | Contains the date on which the store was acquired. Data in this field is optional. Format is YYYYMMDD. | Date | 8 | NA | NA | NA | NA | Ignored. |
| 13 | Remode IDate | Contains the date on which the store was last remodeled. Data in this field is optional. Format is YYYYMMDD. | Date | 8 | NA | NA | NA | NA | Ignored. |

Table B-10 (Cont.) RMS Store Mapping

| Po s | RMS Field Name | RMS Description | RMS Field Type | RMS Max Width | Xstore Table | Xstore Column | Xstore Data Type | Xstore Default Value | Xstore Comment |
|---------|------------------------|--|----------------------|---------------------|---------------------------------|-------------------|------------------------|----------------------------|-------------------------------|
| 14 | FaxNu mber | Contains the fax number for the store. Data in this field is optional. | Character | 20 | loc_rtl_ loc | TELEPH ONE_2 | VARCHA R(32) | NA | Optional fax number. |
| 15 | PhoneN umber | Contains the phone number for the store. Data in this field is optional. | Character | 20 | loc_rtl_ loc | TELEPH ONE_1 | VARCHA R(32) | NA | Optional phone number. |
| 16 | Email | Holds the email address for the location. Data in this field is optional. | Character | 100 | loc_rtl_ loc | EMAIL_ ADDR | VARCHA R(254) | NA | Optional email address. |
| 17 | TotalSq Feet | Contains the total square footage of the store. Data in this field is optional. | Number | 8 | NA | NA | NA | NA | Ignored. |
| 18 | SellingS qFeet | Contains the total square footage of the stores selling area. Data in this field is optional. | Number | 8 | NA | NA | NA | NA | Ignored. |
| 19 | LinearD istance | Holds the total merchandisable space of the location. Data in this field is optional. | Number | 8 | NA | NA | NA | NA | Ignored. |
| 20 | VatRegi on | Contains the number of the Value Added Tax region in which this store is contained. Data in this field is optional. | Number | 4 | tax_rtl_ loc_tax_ mapping | taxLocati onId | VARCHA R2(60) | NA | NA |
| 21 | VatIncll nd | Indicates whether Value Added Tax will be included in the retail prices for the store. Valid values are Y or N. Data in this field is optional. | Character | 1 | NA | NA | NA | NA | Ignored. |
| 22 | StockH oldingI n | This column indicates whether the store can hold inventory. Data will always be present in this field for creation and modification records. | Character | 1 | NA | NA | NA | NA | Ignored. |

Table B-10 (Cont.) RMS Store Mapping

| Po s | RMS Field Name | RMS Description | RMS Field Type | RMS Max Width | Xstore Table | Xstore Column | Xstore Data Type | Xstore Default Value | Xstore Comment |
|---------|-------------------------|--|----------------------|---------------------|-----------------|------------------|------------------------|----------------------------|-------------------|
| 23 | Channe IId | Contains the channel with which the store is associated. Data in this field is optional. | Number | 4 | NA | NA | NA | NA | Ignored. |
| 24 | Channe IName | Contains the name of the channel. Data in this field is optional. | Character | 120 | NA | NA | NA | NA | Ignored. |
| 25 | StoreFo rmat | Contains the number indicating the format of the store. Data in this field is optional. | Number | 4 | NA | NA | NA | NA | Ignored. |
| 26 | StoreFo rmatNa me | Contains the description of the format of the store (for example, mall, standalone, city express, and so on). Data in this field is optional. | Character | 60 | NA | NA | NA | NA | Ignored. |
| 27 | MallNa me | Contains the name of the mall in which the store is located. Data in this field is optional. | Character | 120 | NA | NA | NA | NA | Ignored. |
| 28 | District | Contains the number of the district in which the store is a member. Data will always be present in this field. for creation and modification records. Further organizational hierarchy information is available in the Organizational Hierarchy Extract. | Number | 10 | NA | NA | NA | NA | Ignored. |
| 29 | Transfer Zone | NA | Number | 4 | NA | NA | NA | NA | Ignored. |
| 30 | Default Wh | Contains the number of the warehouse that may be used as the default for creating cross-dock masks. | Number | 10 | NA | NA | NA | NA | Ignored. |

Table B-10 (Cont.) RMS Store Mapping

| Po s | RMS Field Name | RMS Description | RMS Field Type | RMS Max Width | Xstore Table | Xstore Column | Xstore Data Type | Xstore Default Value | Xstore Comment |
|---------|-------------------------|--|----------------------|---------------------|-----------------|------------------|------------------------|----------------------------|--------------------------|
| 31 | StopOr derDay s | Contains the number of days before a store closing when the store will stop accepting orders. This column is used when the store_close_date is defined. | Number | 3 | NA | NA | NA | NA | Ignored. |
| 32 | StartOr derDay s | Contains the number of days before the store_open_date that the store will begin accepting orders. Data will always be present in this field. for creation and modification records. | Number | 3 | NA | NA | NA | NA | Ignored. |
| 33 | Currenc yCode | This field contains the currency code under which the store operates. Data will always be present in this field. for creation and modification records. | Character | 3 | loc_rtl_ loc | CURREN CY_ID | VARCHA R(3) | NA | ISO Currency Code. |
| 34 | StoreLa ngSOC ode | This column identifies the language to be used for the given store. Data will always be present in this field for creation and modification records. | Character | 6 | NA | NA | NA | NA | Ignored. |

Table B-10 (Cont.) RMS Store Mapping

| PoS | RMS Field Name | RMS Description | RMS Field Type | RMS Max Width | Xstore Table | Xstore Column | Xstore Data Type | Xstore Default Value | Xstore Comment |
|-----|-----------------|---|----------------|---------------|--------------|---------------|------------------|----------------------|----------------|
| 35 | TranNo Generate | Contains the level at which unique POS transaction numbers are generated. If the store has one sequence number that is used for all registers, then the value in this column will be S (Store). Otherwise, the store has unique sequence numbers for each register and the value in this column will be R (Register). Data will always be present in this field for creation and modification records. | Character | 6 | NA | NA | NA | NA | Ignored. |
| 36 | IntPOSInd | Indicates whether or not the POS system at the store is integrated. Data will always be present in this field for creation and modification records. | Character | 1 | NA | NA | NA | NA | Ignored. |
| 37 | DunsNumber | This field holds the Dun and Bradstreet (D&B) number to identify the store. A D&B number is a unique identification number for each physical location of a business. A DUNS number may be issued to any business worldwide, and is required by many credit reporting, national government, and trade organizations. Data in this field is optional. | Character | 9 | NA | NA | NA | NA | Ignored. |
| 38 | DunsLoc | Legacy, generally not used. Data in this field is optional. | Character | 4 | NA | NA | NA | NA | Ignored. |

Table B-10 (Cont.) RMS Store Mapping

| Po s | RMS Field Name | RMS Description | RMS Field Type | RMS Max Width | Xstore Table | Xstore Column | Xstore Data Type | Xstore Default Value | Xstore Comment |
|---------|----------------------|--|----------------------|---------------------|-----------------|------------------|------------------------|----------------------------|-------------------|
| 39 | SisterStore | This field holds a store number which is used to relate the current store to the historical data of an existing store. Data in this field is optional. | Number | 10 | NA | NA | NA | NA | Ignored. |
| 40 | TsfEntityId | Data in this field is optional. | Number | 10 | NA | NA | NA | NA | Ignored. |
| 41 | OrgUnitId | Column contains the organizational unit ID value. Data in this field is optional. | Number | 15 | NA | NA | NA | NA | Ignored. |
| 42 | AutoRecv | This column indicates whether the client is allowing automatic receipt for the store. Valid values are Y (Yes), N (No), and D (System Default). Default value should be D. Data will always be present in this field for creation and modification records. | Character | 1 | NA | NA | NA | NA | Ignored. |
| 43 | RemerchInd | Identifies stores that are undergoing a significant re-merchandising effort. Defaults to N. Other values may be present if AIP is integrated with RMS. Data will always be present in this field for creation and modification records. | Character | 1 | NA | NA | NA | NA | Ignored. |
| 44 | StoreType | This indicates whether a particular store is a franchise (F) or company store (C). Data will always be present in this field for creation and modification records. | Character | 6 | NA | NA | NA | NA | Ignored. |

Table B-10 (Cont.) RMS Store Mapping

| Po s | RMS Field Name | RMS Description | RMS Field Type | RMS Max Width | Xstore Table | Xstore Column | Xstore Data Type | Xstore Default Value | Xstore Comment |
|---------|----------------------|---|----------------------|---------------------|---------------------------------------|------------------------------|------------------------|---------------------------------|--|
| 45 | WFCustomer | Numeric ID of the customer. Data in this field is optional. | Number | 10 | NA | NA | NA | NA | Ignored. |
| 46 | Timezone | Indicates the time zone of the store. For example, America/New_York. Data will always be present in this field for creation and modification records. | Character | 64 | NA | NA | NA | NA | Ignored. |
| 47 | CustomerOrderLocInd | This column determines whether the location is a customer order location. If the indicator is Y, the location can be used by OMS for sourcing/ fulfillment or both. Otherwise, it cannot be used. It is used only for the company stores. | Character | 1 | NA | NA | NA | NA | Ignored. |
| NA | NA | NA | NA | NA | loc_rtl_loc | STORE_NBR | VARCHAR(254) | NA | Always RMS StoreId. |
| NA | NA | NA | NA | NA | loc_rtl_loc, tax_rtl_loc, tax_mapping | ORGANIZATION_ID | NUMBER(10,0) | \$(dtv.location.organizationId) | Always uses the default value. |
| NA | NA | NA | NA | NA | loc_rtl_loc | LOCATION_TYPE | VARCHAR(60) | STORE | Value can be configured in dataloader-beans.xml. If not configured, then null. |
| NA | NA | NA | NA | NA | loc_rtl_loc | use_till_accountability_flag | NUMBER(1,0) | FALSE | Value can be configured in dataloader-beans.xml. |
| NA | NA | NA | NA | NA | loc_rtl_loc | CREATE_DATE | TIMESTAMP(6) | now() | Not set when Type = STOREMOD. |

Table B-10 (Cont.) RMS Store Mapping

| Pos | RMS Field Name | RMS Description | RMS Field Type | RMS Max Width | Xstore Table | Xstore Column | Xstore Data Type | Xstore Default Value | Xstore Comment |
|-----|----------------|-----------------|----------------|---------------|--------------|----------------|-------------------|----------------------|-------------------------------|
| NA | NA | NA | NA | NA | loc_rtl_loc | CREATE_USER_ID | VARCHAR2(30 CHAR) | DATALOADER | Not set when Type = STOREMOD. |
| NA | NA | NA | NA | NA | loc_rtl_loc | UPDATE_DATE | TIMESTAMP(6) | now() | Always set. |
| NA | NA | NA | NA | NA | loc_rtl_loc | UPDATE_USER_ID | VARCHAR2(30 CHAR) | DATALOADER | Always set. |

RMS Store Address

Table B-11 describes the Store Address mapping.

Table B-11 RMS Store Address Mapping

| Pos | RMS Field Name | RMS Description | RMS Field Type | RMS Max Width | Xstore Table | Xstore Column | Xstore Data Type | Xstore Default Value | Xstore Comment |
|-----|----------------|---|----------------|---------------|--------------|---------------|------------------|----------------------|--|
| 0 | Family | Value is always STORES. Data will always be present in this field. | Character | 20 | NA | NA | NA | NA | Ignored. |
| 1 | Type | Value is always FULL when the program is run in full mode. If the program is run in delta mode, the type will vary based on the type of change (CRE, MOD, DEL). Valid values are: STOREDTLCRE Creation of store address information. STOREDTLMOD Modification of store address information. STOREDTLDEL Deletion of store address information. Data will always be present in this field. | Character | 15 | NA | NA | NA | NA | Determines action. FULL, STOREDTLCRE, and STOREDTLMOD are all interpreted as updates to an existing loc_rtl_loc record. STOREDTLDEL results in the setting of all non-PK address mapped fields with an empty string. |
| 2 | StoreId | Contains the unique ID of the store. Data will always be present in this field. | Number | 10 | loc_rtl_loc | RTL_LOC_ID | NUMBER(10,0) | NA | NA |

Table B-11 (Cont.) RMS Store Address Mapping

| Po s | RMS Field Name | RMS Description | RMS Field Type | RMS Max Width | Xstore Table | Xstore Column | Xstore Data Type | Xstore Default Value | Xstore Comment |
|---------|----------------------|--|----------------------|---------------------|-----------------|------------------|------------------------|----------------------------|---|
| 3 | AddrType | Contains the code used to identify the address type. Data will always be present in this field. | Character | 2 | NA | NA | NA | NA | Must match the configurable AddrType code, otherwise record is ignored. |
| 4 | AddrTypeDesc | Description of the address type code. Common examples include: 01 - Business, 02 - Postal, 03 - Returns, 04 - Order, 05 - Invoice, 06 - Remittance Optional. | Character | 20 | NA | NA | NA | NA | Ignored. |
| 5 | Primary AddrInd | Indicates whether the address is the primary address for the address type. Valid values are Y and N. Data will always be present in this field. | Character | 1 | NA | NA | NA | NA | If not Y, the entire record is ignored. |
| 6 | Add1 | Contains the first line of the address. Data will always be present in this field for STOREDTLCRE and STOREDTLMOD. | Character | 240 | loc_rtl_ loc | ADDRESS1 | VARCHAR2(254 CHAR) | NA | NA |
| 7 | Add2 | Contains the second line of the address. Optional. | Character | 240 | loc_rtl_ loc | ADDRESS2 | VARCHAR2(254 CHAR) | NA | NA |
| 8 | Add3 | Contains the third line of the address. Optional. | Character | 240 | loc_rtl_ loc | ADDRESS3 | VARCHAR2(254 CHAR) | NA | NA |
| 9 | City | Contains the name of the city that is associated with the address. Data will always be present in this field for STOREDTLCRE and STOREDTLMOD. | Character | 120 | loc_rtl_ loc | CITY | VARCHAR2(30 CHAR) | NA | NA |
| 10 | County | Contains the county name for the location. Optional. | Character | 250 | NA | NA | NA | NA | NA |
| 11 | State | Contains the state abbreviation for the address. Optional. | Character | 3 | loc_rtl_ loc | STATE | VARCHAR2(30 CHAR) | NA | NA |

Table B-11 (Cont.) RMS Store Address Mapping

| Po s | RMS Field Name | RMS Description | RMS Field Type | RMS Max Width | Xstore Table | Xstore Column | Xstore Data Type | Xstore Default Value | Xstore Comment |
|---------|----------------------|---|----------------------|---------------------|-----------------|------------------|------------------------|---------------------------------|----------------------------|
| 12 | Country | Contains the country where the address exists, using ISO 3166-1 alpha-2. Data will always be present in this field for STOREDTLCRE and STOREDTLMOD. | Character | 3 | loc_rtl_loc | COUNTRY | VARCHAR2(254 CHAR) | NA | NA |
| 13 | PostCode | Contains the postal code name for the location. Optional. | Character | 30 | loc_rtl_loc | POSTAL_CODE | VARCHAR2(30 CHAR) | NA | NA |
| 14 | JurisdictionCode | Contains the tax jurisdiction code for the location. Optional. | Character | 10 | NA | NA | NA | NA | Ignored. |
| 15 | Contact Name | Contains the name of the contact person at this address. Optional. | Character | 120 | NA | NA | NA | NA | Ignored. |
| 16 | Contact Phone | Contains a phone number of the contact person at this address. Optional. | Character | 20 | NA | NA | NA | NA | Ignored. |
| 17 | Contact Fax | Contains a fax number of the contact person at this address. Optional. | Character | 20 | NA | NA | NA | NA | Ignored. |
| 18 | Contact Email | Contains an email for the contact person at this address. Optional. | Character | 100 | NA | NA | NA | NA | Ignored. |
| NA | NA | NA | NA | NA | loc_rtl_loc | ORGANIZATION_ID | NUMBER(10,0) | \${dtv.location.organizationId} | Always uses default value. |
| NA | NA | NA | NA | NA | loc_rtl_loc | UPDATE_DATE | TIMESTAMP(6) | now() | NA |
| NA | NA | NA | NA | NA | loc_rtl_loc | UPDATE_USER_ID | VARCHAR2(30 CHAR) | DATALOADER | NA |

RMS VAT

Table B-12 describes the RMS VAT mapping.

Table B-12 RMS VAT Mapping

| P o s | RMS Field Name | RMS Description | RMS Field Type | RMS Max Width | Xstore Table | Xstore Column | Xstore Data Type | Xstore Default Value | Xstore Comment |
|----------------------|-------------------------------|---|-------------------------------|------------------------------|---|------------------------------|---------------------------------|-------------------------------------|--|
| 0 | Family | Value is always VAT. Data will always be present in this field. | Character | 20 | NA | NA | NA | NA | Ignored. |
| 1 | Type | Value is always FULL when the program is run in full mode. If the program is run in delta mode, the type will vary based on the type of change (CRE, MOD, DEL). Valid values are: VATCRE - Creation of VAT information. VATMOD - Modification of VAT information. VATDEL - Deletion of VAT information. Data will always be present in this field. | Character | 15 | NA | NA | NA | NA | Determines action. |
| 2 | VatRegion | ID of the VAT region. Stores are assigned to a VAT region (if VAT is used in RMS). This field will always have data. | Number | 4 | tax_tax_loc, tax_tax_authority, tax_tax_group_rule, tax_tax_rate_rule | TAX_LOC_ID, TAX_AUTHORITY_ID | VARCHAR2(60 CHAR) | NA | Use RMS VAT region ID as the tax authority ID in Xstore. |
| 3 | VatRegionName | This field contains the alphanumeric identification for the VAT code. Valid values include, but are not limited to: S - Standard C - Composite Z - Zero E - Exempt This field will always have data. | Character | 120 | tax_tax_loc, tax_tax_authority | NAME, DESCRIPTION | VARCHAR(254) | NA | Use RMS VAT region name as the tax authority name in Xstore. |

Table B-12 (Cont.) RMS VAT Mapping

| P o s | RMS Field Name | RMS Description | RMS Field Type | RMS Max Width | Xstore Table | Xstore Column | Xstore Data Type | Xstore Default Value | Xstore Comment |
|-------------|----------------------|--|----------------------|---------------------|--|---------------------|------------------------|----------------------------|---|
| 4 | VatCode | This field contains the description of the VAT Code. This field will always have data in the VATCRE and VATMOD records. | Character | 6 | tax_tax_group, tax_tax_group_rule, tax_tax_rate_rule | TAX_GROUP_ID | VARCHAR(60) | NA | NA |
| 5 | VatCodeDesc | This field contains the description of the VAT Code. This field will always have data in the VATCRE and VATMOD records. | Character | 120 | tax_tax_group, tax_tax_group_rule | NAME, DESCRIPTION | VARCHAR(254) | NA | For the root COMPANY node, the column is hard-coded as null. For the second level CHAIN node, the column is hard-coded to "*". |
| 6 | ActiveDate | This field is the date the VAT code is active for the VAT region, in human readable format DD-MON-YYYY. This field will always have data. | Character | 12 | tax_tax_rate_rule | PERCENTAGE | NUMBER(8,6) | NA | NA |
| 7 | VatRate | VAT rate for the VAT code/VAT region. This field will always have data in the VATCRE and VATMOD records. | Number | 20 | tax_tax_rate_rule | EFFECTIVE_DATE_TIME | TIMESTAMP(6) | NA | NA |
| NA | NA | NA | NA | NA | tax_tax_group_rule | TAX_TYPCODE | VARCHAR(30) | VAT | Always uses default value. |
| NA | NA | NA | NA | NA | tax_tax_group_rule | TAX_RULE_SEQ_NBR | NUMBER(10,0) | 1 | Always uses default value. The assumption is that there can only be one tax rule per VAT region/VAT code combination. |

Table B-12 (Cont.) RMS VAT Mapping

| P o s | RMS Field Name | RMS Description | RMS Field Type | RMS Max Width | Xstore Table | Xstore Column | Xstore Data Type | Xstore Default Value | Xstore Comment |
|----------------------|-------------------------------|----------------------------|-------------------------------|------------------------------|----------------------------|---|---------------------------------|-------------------------------------|--|
| NA | NA | NA | NA | NA | tax_tax_ group_ rule | TAXED_ AT_ TRANS_ LEVEL_ FLAG | NUMBER (1,0) | TRUE | A configurati on spring loaded by the transformer . The out-of-box default is TRUE. |

Table B-12 (Cont.) RMS VAT Mapping

| P o s | RMS Field Name | RMS Description | RMS Field Type | RMS Max Width | Xstore Table | Xstore Column | Xstore Data Type | Xstore Default Value | Xstore Comment |
|----------------------|-------------------------------|----------------------------|-------------------------------|------------------------------|-------------------------|--------------------------|---------------------------------|-------------------------------------|---|
| | NA | NA | NA | NA | tax_tax_ rate_rule | EXPR_ DATETI ME | TIMESTA MP(6) | NA | <p>This has to be derived by the integration layer.</p> <p>During a CRE/FULL , if there is a subsequent rate rule as sorted by effective date, expire this rate rule by setting its expiration date to be 1 millisecond less than the effective date of the subsequent rate rule; otherwise do not expire this rate rule by setting its expiration date to NULL.</p> <p>During a CRE/FULL , if there is a prior rate rule as sorted by effective date, expire the prior rate rule by setting its expiration date to be 1 millisecond less than this effective date.</p> |

Table B-12 (Cont.) RMS VAT Mapping

| P o s | RMS Field Name | RMS Description | RMS Field Type | RMS Max Width | Xstore Table | Xstore Column | Xstore Data Type | Xstore Default Value | Xstore Comment |
|-------------|----------------------|--------------------|----------------------|---------------------|-----------------------|--|------------------------|----------------------------|--|
| NA | NA | NA | NA | NA | NA | EXPR_ DATETI ME (Continue d) | NA | NA | During a DEL, if there is a prior rate rule as sorted by effective date, extend the prior rate rule by setting its expiration date to be the expiration date of this rate rule. |
| NA | NA | NA | NA | NA | tax_tax_ rate_rule | TAX_ RULE_ SEQ_ NBR | NA | 1 | Always uses the default value. The assumption is that there can only be one tax rule per VAT region/VA T code combinatio n. |
| NA | NA | NA | NA | NA | tax_tax_ rate_rule | TAX_ RATE_ RULE_ SEQ | NUMBER (10,0) | NA | This has to be derived by the integration layer. For a newly created VAT rate, its sequence number is one after the maximum rate rule sequence number in the table. |

Table B-12 (Cont.) RMS VAT Mapping

| P o s | RMS Field Name | RMS Description | RMS Field Type | RMS Max Width | Xstore Table | Xstore Column | Xstore Data Type | Xstore Default Value | Xstore Comment |
|-------------|----------------------|--------------------|----------------------|---------------------|--|--|------------------------|----------------------------|---|
| NA | NA | NA | NA | NA | tax_tax_ authority | ROUNDI NG_ CODE | VARCHA R(30) | HALF_ UP | A configurati on spring loaded by the transformer . The out-of-box default is HALF_UP. |
| NA | NA | NA | NA | NA | tax_tax_ authority | ROUNDI NG_ DIGITS_ QUANTI TY | NUMBER (10,0) | 2 | A configurati on spring loaded by the transformer . The out-of-box default is 2. |
| NA | NA | NA | NA | NA | tax_tax_ loc, tax_ tax_ authority, tax_tax_ group, tax_tax_ group_ rule, tax_ tax_rate_ rule | EXTERN AL_ SYSTEM | VARCHA R(30) | RMS | Always uses the default value. |
| NA | NA | NA | NA | NA | tax_tax_ loc, tax_ tax_ authority, tax_tax_ group, tax_tax_ group_ rule, tax_ tax_rate_ rule | ORG_ CODE | VARCHA R(30) | * | Always uses the default value. |
| NA | NA | NA | NA | NA | tax_tax_ loc, tax_ tax_ authority, tax_tax_ group, tax_tax_ group_ rule, tax_ tax_rate_ rule | ORG_ VALUE | VARCHA R(60) | * | Always uses the default value. |

Table B–12 (Cont.) RMS VAT Mapping

| P o s | RMS Field Name | RMS Description | RMS Field Type | RMS Max Width | Xstore Table | Xstore Column | Xstore Data Type | Xstore Default Value | Xstore Comment |
|-------------|----------------------|--------------------|----------------------|---------------------|--|-------------------------|------------------------|--|---|
| NA | NA | NA | NA | NA | tax_tax_ loc, tax_ tax_ authority, tax_tax_ group, tax_tax_ group_ rule, tax_ tax_rate_ rule | ORGANI ZATION_ ID | NUMBER (10,0) | #{dtv.loc ation.or ganizati onId} | Always uses the default value. |
| NA | NA | NA | NA | NA | tax_tax_ loc, tax_ tax_ authority, tax_tax_ group, tax_tax_ group_ rule, tax_ tax_rate_ rule | CREATE_ DATE | TIMESTA MP(6) | now() | Not set when Type represents UPDATE. |
| NA | NA | NA | NA | NA | tax_tax_ loc, tax_ tax_ authority, tax_tax_ group, tax_tax_ group_ rule, tax_ tax_rate_ rule | UPDATE_ DATE | TIMESTA MP(6) | now() | Not set when Type represents CREATE. |

RMS VAT Item

Table B–13 describes the RMS VAT Item mapping.

Note: Do not import a .dat file containing multiple VATITEM records with the same item ID. More than one record for the same item ID could result in an undesired outcome.

Table B–13 RMS VAT Item Mapping

| P o s | RMS Field Name | RMS Description | RMS Field Type | RMS Max Width | Xstore Table | Xstore Column | Xstore Data Type | Xstore Default Value | Xstore Comment |
|-------------|----------------------|---|----------------------|---------------------|-----------------|------------------|------------------------|----------------------------|-------------------|
| 0 | Family | Value is always VATITEM. Data will always be present in this field. | Character | 20 | NA | NA | NA | NA | Ignored. |

Table B-13 (Cont.) RMS VAT Item Mapping

| P o s | RMS Field Name | RMS Description | RMS Field Type | RMS Max Width | Xstore Table | Xstore Column | Xstore Data Type | Xstore Default Value | Xstore Comment |
|-------------|----------------------|--|----------------------|---------------------|------------------|------------------|------------------------|----------------------------|---------------------|
| 1 | Type | Value is always FULL when the program is run in full mode. If the program is run in delta mode, the type will vary based on the type of change (CRE, MOD, DEL). Valid values are: VATITEMCRE - Creation of VAT Item information. VATITEMMOD - Modification of VAT Item information. VATITEMDEL - Deletion of VAT Item information. Data will always be present in this field. | Character | 15 | NA | NA | NA | NA | Determines actions. |
| 2 | Item | ID of the item. Data will always be present in this field. | Character | 25 | itm_item_options | item_id | VARCHAR(60 CHAR) | NA | NA |
| 3 | VatRegion | ID of the VAT region. Stores are assigned to a VAT region if VAT is used in RMS. Additional information about VAT Regions is stored in RMS, but is not integrated out of the system. VAT region information may need to be synchronized to downstream systems as a manual process. This field will always have data. | Number | 4 | NA | NA | NA | NA | Ignored. |
| 4 | ActiveDate | Date that the VAT rate becomes active, in DD-MON-YYYY format. This field will always have data in the VATITEMCRE and VATITEMMOD records. | Character | 11 | NA | NA | NA | NA | Ignored. |

Table B-13 (Cont.) RMS VAT Item Mapping

| P o s | RMS Field Name | RMS Description | RMS Field Type | RMS Max Width | Xstore Table | Xstore Column | Xstore Data Type | Xstore Default Value | Xstore Comment |
|----------------------|-------------------------------|---|-------------------------------|------------------------------|-------------------------|--------------------------|---------------------------------|-------------------------------------|--|
| 5 | VatType | The field describes what the VAT rate applies to: C - Cost values R - Retail values B - Both Cost and Retail values This field will always have data. | Character | 1 | NA | NA | NA | NA | Record ignored when VatType=C. |
| 6 | VatCode | This field contains the alphanumeric identification for the VAT code. Valid values include, but are not limited to: S - Standard C - Composite Z - Zero E - Exempt If additional VAT Codes are defined in RMS, the VAT code information may need to be synchronized to downstream systems as a manual process. This field will always have data. | Character | 6 | itm_item_options | tax_group_id | VARCHAR(60) | NA | Column tax_group_id is set to null if Type is VATITEMDEL. Column tax_group_id is unchanged when VatType is C. Otherwise, column tax_group_id is set to the value of the VatCode. |
| 7 | VatRate | VAT rate for the item/VAT region. This field will always have data in the VATITEMCRE and VATITEMMOD records. | Number | 20 | NA | NA | NA | NA | Ignored. |
| 8 | Reverse VatInd | Indicates if the item is subject to reverse charge VAT at the VAT region. VAT reversal can apply to invoices on some items in some VAT regions. Valid values are Y and N. This field will always have data in the VATITEMCRE and VATITEMMOD records. | Character | 1 | NA | NA | NA | NA | Ignored. |

Table B-13 (Cont.) RMS VAT Item Mapping

| P o s | RMS Field Name | RMS Description | RMS Field Type | RMS Max Width | Xstore Table | Xstore Column | Xstore Data Type | Xstore Default Value | Xstore Comment |
|-------------|----------------------|--------------------|----------------------|---------------------|------------------|------------------|------------------------|----------------------------|-------------------|
| NA | NA | NA | NA | NA | itm_item_options | UPDATE_USER_ID | VARCHAR2(30 CHAR) | DATALOADER | NA |
| NA | NA | NA | NA | NA | itm_item_options | UPDATE_DATE | TIMESTAMP(6) | now() | NA |

RPM Clearance Price

The following tables describe the RPM Clearance Price mapping.

[Table B-14](#) describes the Clearance Price FDETL record mapping. FDETL records are interpreted as Create or Update events.

Table B-14 RPM Clearance Price FDETL Record Mapping

| P o s | RPM Field Name | RPM Description | RPM Field Type | Xstore Table | Xstore Column | Xstore Data Type | Xstore Default Value | Xstore Comment |
|-------------|-------------------------|--|----------------------|-----------------|------------------|---------------------|----------------------------|--|
| 0 | Record Descriptor | File Detail Marker (1 per clearance create/modify) | Char(5) | NA | NA | NA | NA | Used to determine unit type. |
| 1 | Line Id | Unique line identification | Number(10) | NA | NA | NA | NA | Ignored. |
| 2 | Event Type | CRE = Create, MOD = Modify | Char(3) | NA | NA | NA | NA | Used to determine the action type. |
| 3 | Id | Clearance identifier | Number(15) | itm_item_prices | EXTERNAL_ID | VARCHAR2(60 CHAR) | NA | NA |
| 4 | Item | Item identifier | Char(25) | itm_item_prices | ITEM_ID | VARCHAR2(60 CHAR) | NA | NA |
| 5 | Effective Date | Clearance Effective Date (YYYYMMDDHH24MISS) | Date | itm_item_prices | EFFECTIVE_DATE | TIMESTAMP(6) | NA | NA |
| 6 | Selling Retail | Selling retail with price change applied | Number(20,4) | itm_item_prices | PRICE | NUMBER(17,6) | NA | NA |
| 7 | Selling Retail UOM | Selling retail unit of measure | Char(4) | NA | NA | NA | NA | Ignored |
| 8 | Selling Retail Currency | Selling retail currency | Char(3) | NA | NA | NA | NA | Ignored (assumed to be base Currency). |

Table B-14 (Cont.) RPM Clearance Price FDETL Record Mapping

| P o s | RPM Field Name | RPM Description | RPM Field Type | Xstore Table | Xstore Column | Xstore Data Type | Xstore Default Value | Xstore Comment |
|-------------|--------------------------|-----------------------------------|----------------------|---------------------|---------------------|-----------------------|---|--|
| 9 | Reset Clearance ID | Clearance reset identification | Number(15) | NA | NA | NA | NA | Used to detect clearance reset. Clearance reset is when this value exactly matches the ID field's value. |
| N A | NA | NA | NA | itm_item_ prices | ORGANIZ ATION_ID | NUMBER(1 0,0) | \${dtv.loc ation.or ganizati onId} | Always uses default value. |
| N A | NA | NA | NA | itm_item_ prices | PROPERTY _CODE | VARCHAR 2(60 CHAR) | NA | CLEARA NCE_ PRICE when Clearance event, or REGULA R_PRICE when Clearance Reset event. |
| N A | NA | NA | NA | itm_item_ prices | CREATE_ DATE | TIMESTAM P(6) | now() | Not set when EventType = Mod. |
| N A | NA | NA | NA | itm_item_ prices | CREATE_ USER_ID | VARCHAR 2(30 CHAR) | DATAL OADER | Not set when EventType = Mod. |
| N A | NA | NA | NA | itm_item_ prices | UPDATE_ DATE | TIMESTAM P(6) | now() | Always set. |
| N A | NA | NA | NA | itm_item_ prices | UPDATE_ USER_ID | VARCHAR 2(30 CHAR) | DATAL OADER | Always set. |
| N A | NA | NA | NA | itm_item_ prices | EXTERNA L_SYSTEM | VARCHAR 2(60 CHAR) | RPM-C LRPC | NA |

Table B-15 describes the Clearance Price FDELE record mapping. FDELE records are interpreted as Delete events.

Table B-15 RPM Clearance Price FDELE Record Mapping

| P o s | RPM Field Name | RPM Description | RPM Field Type | Xstore Table | Xstore Column | Xstore Data Type | Xstore Default Value | Xstore Comment |
|-------------|----------------------|---|----------------------|---------------------|---------------------|-----------------------|---|-------------------------------------|
| 0 | Record Descriptor | File Detail Delete Marker (1per clearance delete) | Char(5) | NA | NA | NA | NA | Used to determine unit type. |
| 1 | Line Id | Unique line identification | Number(10) | NA | NA | NA | NA | Ignored. |
| 2 | Id | Clearance identifier | Number(15) | itm_item_ prices | EXTERNA L_ID | VARCHAR 2(60 CHAR) | NA | NA |
| 3 | Item | Item identifier | Char(25) | itm_item_ prices | ITEM_ID | VARCHAR 2(60 CHAR) | NA | NA |
| | NA | NA | NA | itm_item_ prices | ORGANIZ ATION_ID | NUMBER(1 0,0) | \${dtv.loc ation.or ganizati onId} | Always uses default value. |
| | NA | NA | NA | itm_item_ prices | EXTERNA L_SYSTEM | VARCHAR 2(60 CHAR) | RPM-C LRPC | NA |

RPM Regular Price

The following tables describe the RPM Regular Price mapping.

[Table B-16](#) describes the Regular Price FDETL record mapping. FDETL records are interpreted as Create or Update events.

Table B-16 RPM Regular Price FDETL Record Mapping

| Po s | RPM Field Name | RPM Description | RPM Field Type | Xstore Table | Xstore Column | Xstore Data Type | Xstore Default Value | Xstore Comment |
|---------|----------------------|---|----------------------|---------------------|--------------------|-----------------------|----------------------------|--|
| 0 | Record Descriptor | File Detail Marker (1 per price change create or modify) | Char(5) | NA | NA | NA | NA | Type of event. |
| 1 | Line Id | Unique line identification | Number(10) | NA | NA | NA | NA | Ignored. |
| 2 | Event Type | CRE = Create, MOD = Modify | Char(3) | NA | NA | NA | NA | Used by the transform er to determine the type of event. |
| 3 | Id | Price change identifier | Number(15) | itm_item_ prices | EXTERNA L_ID | VARCHAR 2(60 CHAR) | NA | NA |
| 4 | Item | Item identifier | Char(25) | itm_item_ prices | ITEM_ID | VARCHAR 2(60 CHAR) | NA | NA |
| 5 | Effective Date | Effective Date of the price change (YYYYMMDDHH2 4MISS) | NA | itm_item_ prices | EFFECTIV E_DATE | TIMESTAM P(6) | NA | NA |

Table B-16 (Cont.) RPM Regular Price FDETL Record Mapping

| Po s | RPM Field Name | RPM Description | RPM Field Type | Xstore Table | Xstore Column | Xstore Data Type | Xstore Default Value | Xstore Comment |
|------|-------------------------|---|----------------|-----------------|----------------------|-------------------|--------------------------------|--|
| 6 | Selling Unit Change Ind | Indicates whether the selling unit retail changed with this price event (0 = no change, 1 = changed). | Number(1) | NA | NA | NA | NA | This field need not be persisted in the Xstore database. This field is used in transformer logic. |
| 7 | Selling Retail | Selling retail with price change applied | Number(20,4) | itm_item_prices | PRICE | NUMBER(17,6) | NA | NA |
| 8 | Selling Retail UOM | Selling retail unit of measure | Char(4) | itm_item | UNIT_OF_MEASURE_CODE | VARCHAR2(30 BYTE) | NA | Ignored. |
| 9 | Selling Retail Currency | Selling retail currency | Char(3) | NA | NA | NA | NA | Ignored (assumed to be base Currency). |
| 10 | Multi-Unit Change Ind | Did multi-unit retail change with this price event (0 = no change, 1 = changed) | Number(1) | NA | NA | NA | NA | NA |
| 11 | Multi-Units | NA | Number(12,4) | NA | NA | NA | NA | NA |
| 12 | Multi-Unit Retail | Number of multi-units | Number(20,4) | NA | NA | NA | NA | NA |
| 13 | Multi-Unit UOM | Multi-Unit Retail unit of measure | Char(4) | NA | NA | NA | NA | NA |
| 14 | Multi-Unit Currency | Multi-Unit Retail Currency | Char(3) | NA | NA | NA | NA | NA |
| NA | NA | NA | NA | itm_item_prices | ORGANIZATION_ID | NUMBER(10,0) | #{dtv.location.organizationId} | Always uses the default value. |
| NA | NA | NA | NA | itm_item_prices | PROPERTY_CODE | VARCHAR2(60 CHAR) | NA | REGULAR_PRICE will be the value. |
| NA | NA | NA | NA | itm_item_prices | CREATE_DATE | TIMESTAMP(6) | now() | Current time stamp. Not set when the event type =MOD. |

Table B-16 (Cont.) RPM Regular Price FDETL Record Mapping

| PoS | RPM Field Name | RPM Description | RPM Field Type | Xstore Table | Xstore Column | Xstore Data Type | Xstore Default Value | Xstore Comment |
|-----|----------------|-----------------|----------------|-----------------|-----------------|-------------------|----------------------|--|
| NA | NA | NA | NA | itm_item_prices | CREATE_USER_ID | VARCHAR2(30 CHAR) | DATALOADER | This is the user ID stored in the database. Not set when event type = Mod. |
| NA | NA | NA | NA | itm_item_prices | UPDATE_DATE | TIMESTAMP(6) | now() | Current time stamp. Field value only set when event type=MOD. |
| NA | NA | NA | NA | itm_item_prices | UPDATE_USER_ID | VARCHAR2(30 CHAR) | DATALOADER | DATALOADER will be the user ID stored in the database. Set only when the event type=MOD. |
| NA | NA | NA | NA | itm_item_prices | EXTERNAL_SYSTEM | VARCHAR2(60 CHAR) | RPM-REGPC | NA |

Table B-17 describes the Regular Price FDELE record mapping. FDELE records are interpreted as Delete events.

Table B-17 RPM Regular Price FDELE Record Mapping

| PoS | RPM Field Name | RPM Description | RPM Field Type | Xstore Table | Xstore Column | Xstore Data Type | Xstore Default Value | Xstore Comment |
|-----|-------------------|---|----------------|-----------------|---------------|-------------------|----------------------|---|
| 0 | Record Descriptor | File Detail Delete Marker (1 per price change delete) | Char(5) | NA | NA | NA | NA | Used by the transformer to determine the unit type. |
| 1 | Line Id | Unique line identification | Number(10) | NA | NA | NA | NA | Ignored. |
| 2 | Id | Price change identifier | Number(15) | itm_item_prices | EXTERNAL_ID | VARCHAR2(60 CHAR) | NA | NA |
| 3 | Item | Item identifier | Char(25) | itm_item_prices | ITEM_ID | VARCHAR2(60 CHAR) | NA | NA |

Table B-17 (Cont.) RPM Regular Price FDELE Record Mapping

| P o s | RPM Field Name | RPM Description | RPM Field Type | Xstore Table | Xstore Column | Xstore Data Type | Xstore Default Value | Xstore Comment |
|-------------|----------------------|--------------------|----------------------|---------------------|---------------------|-----------------------|---|-------------------|
| NA | NA | NA | NA | itm_item_ prices | ORGANIZ ATION_ID | NUMBER(1 0,0) | \${dtv.loc ation.or ganizati onId} | NA |
| NA | NA | NA | NA | itm_item_ prices | EXTERNA L_SYSTEM | VARCHAR 2(60 CHAR) | RPM-C LRPC | NA |

RPM Promotions

Table B-18 describes the RPM Promotions mapping.

Table B-18 RPM Promotions Record Mappings

| RPM Record Name | RPM Field Name | RPM Field Type | RPM Default Value | RPM Description | Xstore Table | Xstore Field | Xstore Comment |
|-----------------------|----------------------|-------------------|-------------------------|--|-----------------|------------------------|-------------------|
| FHEAD | Record Descriptor | Char(5) | FHEAD | File head marker | NA | NA | NA |
| | Line ID | Number(10) | 1 | Unique line identifier | NA | NA | NA |
| | File Type | Char(5) | PROMO | Promotions | NA | NA | NA |
| | Export Timestamp | Number(10) | | System clock timestamp (YYYYMMDDHHMI SS) | NA | NA | NA |
| | Format Version | Char(5) | 1.0 | File Format Version | NA | NA | NA |
| | Location | Number(10) | | Location identifier | NA | NA | NA |
| | Location Type | Char(1) | | S = Store, W = Warehouse | NA | NA | NA |
| TIMBPE | Record Descriptor | Char(5) | TIMBPE | Promotion (transaction head). Defines an action associated with a promotion. | NA | NA | NA |
| | Line ID | Number(10) | | Unique line identifier | NA | NA | NA |
| | Event Type | Char(3) | | CRE = Create, MOD = Modify | NA | NA | NA |
| TPDTL | Record Descriptor | Char(5) | TPDTL | Promotion Detail Component. Defines a new component. | NA | NA | Ignored. |
| | Line ID | Number(10) | | Unique line identifier | NA | NA | Ignored. |
| | Promo ID | Number(10) | | Promotion identifier | prc_deal | deal_id (partially) | NA |
| | Promo Comp ID | Number(10) | | Promotion Component Id | prc_deal | deal_id (partially) | NA |
| | Promo Name | Char(160) | | Promotion Header Name | NA | NA | Ignored. |

Table B-18 (Cont.) RPM Promotions Record Mappings

| RPM Record Name | RPM Field Name | RPM Field Type | RPM Default Value | RPM Description | Xstore Table | Xstore Field | Xstore Comment |
|-----------------|----------------------|----------------|-------------------|--|----------------|---------------------|--|
| | Promo Desc | Char(640) | | Promotion Header Description | NA | NA | Ignored. |
| | Promo Comp Desc | Char(160) | | Promotion Component Name | prc_deal | description | NA |
| | Promo Type | Number(2) | | Valid values: 0 = Multi-Buy Promotion 1 = Simple Promotion 2 = Threshold Promotion 3 = Finance Promotion (formerly tied to a value of 6) 4 = Transaction Promotion | NA | NA | Determines structure of resulting deal |
| | Promo Comp Detail ID | Number(10) | | Promotion Component Detail identifier | prc_deal | deal_id (partially) | NA |
| | Date Start | Date | | Start Date of Promotion Component Detail (YYYYMMDDHH24 MISS) | prc_deal | effective_date | NA |
| | End Date | Date | | End Date of Promotion Component Detail (YYYYMMDDHH24 MISS) | prc_deal | end_date | NA |
| | Apply to Code | Number(1) | | Holds the Apply to Code for the promotion detail. Determines if the promotion is applied to regular retail only (no clearances in effect), clearance retail only (only when a clearance is in effect), or both regular and clearance retail. Valid values: 0 = Regular Only 1 = Clearance Only 2 = Regular and Clearance | prc_field_test | NA | ApplyTo is mapped to an additional field test (PRICE BETWEEN X Y). |
| | Discount Limit | Number(3) | | The number of times that the promotion can be applied to a transaction. | prc_deal | iterationCap | NA |

Table B-18 (Cont.) RPM Promotions Record Mappings

| RPM Record Name | RPM Field Name | RPM Field Type | RPM Default Value | RPM Description | Xstore Table | Xstore Field | Xstore Comment |
|------------------------|------------------------------|-----------------------|--------------------------|---|---------------------|---------------------|--|
| | Apply Order | Number(1) | | Application Order of the Promotion | NA | NA | Ignored. |
| | Threshold ID | Number(6) | | Threshold identifier | NA | NA | Ignored. |
| | Customer Type ID | Number(10) | | Customer Type identifier | NA | NA | Ignored. |
| | Threshold Qualification Type | Number(1) | | The qualification type for the threshold. Will only be populated for threshold promotions. Valid values are 0 for item level and 1 for threshold level. | NA | NA | Ignored. |
| TPGRP | Record Descriptor | Char(5) | TPGRP | Promotion Detail Group | NA | NA | Ignored. |
| | Line ID | Number(10) | | Unique line identifier | NA | NA | Ignored. |
| | Group ID | Number(10) | | Group Number | prc_deal_item | ordinal | Mapped structurally, not by value. |
| TGLIST | Record Descriptor | Char(5) | TGLIST | Promotion Group List. Defines an item list. | NA | NA | Ignored. |
| | Line ID | Number(10) | | Unique line identifier | NA | NA | Ignored. |
| | List ID | Number(10) | | List identifier | NA | NA | Ignored. |
| | Reward Application | Number(1) | | How this reward is applied to the promotion detail. | NA | NA | Not mapped directly, but rather used to determine whether an action from the underlying TPDSC record should be used. |
| | Description | Char(120) | | Description | NA | NA | Ignored. |
| | Price Range Min | Number(20, 4) | | Contains price range promotion minimum valid retail value. | prc_deal_field_test | NA | Mapped to an additional field test (PRICE BETWEEN X Y). |

Table B-18 (Cont.) RPM Promotions Record Mappings

| RPM Record Name | RPM Field Name | RPM Field Type | RPM Default Value | RPM Description | Xstore Table | Xstore Field | Xstore Comment |
|-----------------|-------------------|----------------|-------------------|--|---------------------|--------------|---|
| | Price Range Max | Number(20, 4) | | Contains price range promotion maximum valid retail value. | prc_deal_field_test | NA | Mapped to an additional field test (PRICE BETWEEN X Y). |
| TLITM | Record Descriptor | Char(5) | TLITM | Promotion Group List. RPM defines lists by providing SKUs of eligible items. | NA | NA | Ignored. |
| | Line ID | Number(10) | | Unique line identifier | NA | NA | Ignored. |
| | Item ID | Char(25) | | Transaction Item Identifier | prc_deal_field_test | value1 | Mapped as SKU EQUAL field test. |
| TPDSC | Record Descriptor | Char(5) | TPDSC | Discount Detail for List. Defines a discount to apply to a preceding list of items. | NA | NA | Ignored. |
| | Line ID | Number(10) | | Unique line identifier | NA | NA | Ignored. |
| | Change Type | Number(2) | | Change Type Valid values: -1 = No Change 0 = Percent Off 1 = Amount Off 2 = Fixed Price | prc_deal_item | action | Type of action: AMOUNT_OFF > CURRENCY_OFF PERCENT_OFF > PERCENT_OFF FIXED_PRICE > NEW_PRICE |
| | Change Amount | Number(20, 4) | | Change Amount | prc_deal_item | action_arg | Discount amount when Change Type is AMOUNT_OFF or FIXED_PRICE. Negated for AMOUNT_OFF. |
| | Change Currency | Char(3) | | Change Currency | NA | NA | Ignored. |

Table B-18 (Cont.) RPM Promotions Record Mappings

| RPM Record Name | RPM Field Name | RPM Field Type | RPM Default Value | RPM Description | Xstore Table | Xstore Field | Xstore Comment |
|-----------------|-------------------------|----------------|-------------------|--|---------------|-------------------------------------|---|
| | Change Percent | Number(20,4) | | Change Percent | prc_deal_item | action_arg | Discount amount when Change Type is PERCENT_OFF. Negated. |
| | Change Selling UOM | Char(4) | | Change Selling UOM | NA | NA | Ignored. |
| | Qual Type | Number(2) | | Qualification Type | NA | NA | Affects where the Qual Value goes. |
| | Qual Value | Number(2) | | Qualification Value | prc_deal_item | min_qty / max_qty or min_item_total | Target field is picked based on the Qual Type value. PRC_DEAL_ITEM.max_qty either gets the same value or receives some big number depending on the deal type. |
| | Change Duration | Number(20,4) | | Change Duration | NA | NA | Ignored. |
| TPISR | Record Descriptor | Char(5) | TPISR | This record type is ignored for now. | NA | NA | Ignored. |
| | Line ID | Number(10) | | Unique line identifier | NA | NA | Ignored. |
| | Item ID | Char(25) | | Transaction Item Identifier | NA | NA | Ignored. |
| | Selling Retail | Number(20,4) | | Selling retail of the item | NA | NA | Ignored. |
| | Selling UOM | Char(4) | | Selling UOM of the item | NA | NA | Ignored. |
| | Effective Date | Date | | Effective Date of the selling retail - YYYYMMDDHH24 MISS | NA | NA | Ignored. |
| | Selling Retail Currency | Char(3) | | Selling retail currency | NA | NA | Ignored. |
| TPCDT | Record Descriptor | Char(5) | TPCDT | Credit Detail | NA | NA | Ignored. |

Table B-18 (Cont.) RPM Promotions Record Mappings

| RPM Record Name | RPM Field Name | RPM Field Type | RPM Default Value | RPM Description | Xstore Table | Xstore Field | Xstore Comment |
|------------------------|-----------------------|-----------------------|--------------------------|---|---------------------|---------------------|-----------------------|
| | Credit Detail ID | Number(10) | | Credit Detail Identifier | NA | NA | Ignored. |
| | Line ID | Number(10) | | Unique line identifier | NA | NA | Ignored. |
| | Credit Type | Char(40) | | Credit Type | NA | NA | Ignored. |
| | binNumberFrom | Number(10) | | Bin Number From | NA | NA | Ignored. |
| | binNumberTo | Number(10) | | Bin Number To | NA | NA | Ignored. |
| | Commission Rate | Number(10) | | Commission Rate | NA | NA | Ignored. |
| | Comments | Char(160) | | Comments | NA | NA | Ignored. |
| TPCIL | Record Descriptor | Char(5) | TPCIL | Cancel Item Loc | NA | NA | Ignored. |
| | Line ID | Number(10) | | Unique line identifier | NA | NA | Ignored. |
| | Promo ID | Number(10) | | Identifier of the promotion | NA | NA | Ignored. |
| | Promo Comp ID | Number(10) | | Promotion Component Identifier | NA | NA | Ignored. |
| | Promo Comp Detail ID | Number(10) | | Promotion Component Detail identifier | NA | NA | Ignored. |
| | Item ID | Char(25) | | Transaction Item Identifier for item | NA | NA | Ignored. |
| | Cancellation Date | Date | | Cancellation effective date - YYYYMMDDHH24 MISS | NA | NA | Ignored. |
| TTAIL | Record Descriptor | Char(5) | TTAIL | Transaction Tail | NA | NA | NA |
| | Line ID | Number(10) | | Unique line identifier | NA | NA | NA |
| FPDEL | Record Descriptor | Char(5) | FPDEL | Delete Promotion | NA | NA | NA |
| | Line ID | Number(10) | | Unique line identifier | NA | NA | NA |
| | Promo ID | Number(10) | | The ID of the promotion | NA | NA | NA |
| | Promo Comp ID | Number(10) | | Promotion Component Identifier | NA | NA | NA |
| | Promo Comp Detail ID | Number(10) | | Promotion Component Detail identifier | NA | NA | NA |
| | Group ID | Number(10) | | Group Number | NA | NA | NA |
| | List ID | Number(10) | | List identifier | NA | NA | NA |

Table B-18 (Cont.) RPM Promotions Record Mappings

| RPM Record Name | RPM Field Name | RPM Field Type | RPM Default Value | RPM Description | Xstore Table | Xstore Field | Xstore Comment |
|------------------------|-----------------------|-----------------------|--------------------------|---|---------------------|---------------------|-----------------------|
| | Item ID | Char(25) | | Transaction Item Identifier for item | NA | NA | NA |
| FTAIL | Record Descriptor | Char(5) | FTAIL | File tail marker | NA | NA | NA |
| | Line ID | Number(10) | | Unique line identifier | NA | NA | NA |
| | Number of lines | Number(10) | | Number of lines in the file not including FHEAD and FTAIL | NA | NA | NA |