Oracle Retail Merchandising Cloud Services/Xstore Suite 25.0

Implementation Guide

G34752-02
August 2025

Oracle Retail Merchandising Cloud Services/Xstore Suite 25.0 Implementation Guide,

G34752-02

Primary Author: Gerlinde Rust

Contents

3 Transaction Flow from Xstore to Sales Audit

4 Webservice Integration Configuration

5 Integration Considerations

6 RTLog Generator On-Premise

7 RTLog Generator Cloud

A Appendix: Xstore to Sales Audit Mapping Details

B Appendix: Flat File Mapping

List of Figures

List of Tables

Send Us Your Comments

Oracle Retail Merchandising Cloud Services/Xstore Suite 25.0 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 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, 25.0) or a later patch release (for example, 25.0.1). 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.
italic	Italic type indicates book titles, emphasis, or placeholder variables for which you supply particular values.

Convention	Meaning
`monospace`	Monospace type indicates commands within a paragraph, URLs, code in examples, text that appears on the screen, or text that you enter.

1

Overview

This document describes integration between the Xstore suite of solutions and the Merchandising Cloud Services 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 Cloud Services 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 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 Merchandising Integration Matrix

Xstore Office	Merchandising and Pricing System Data		
	Direct Web Services[1]	OCDS[2]	Flat File[3]
Cloud	Yes	Yes	Yes
On-Premise	Yes[4]	Yes	Yes

1 Introduced in v21 of MFCS /RPCS
2 OCDS is only available with MFCS v19x. It was replaced with Direct Web Services in v21+.
3 RPCS flat file export does not support export of promotions
4 OAuth credentials to be obtained from Oracle Retail Home

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 deployment topologies: Cloud vs On-Premise.

Table 1-2 Outbound Xstore Suite Merchandising Integration Matrix

Xstore Office	Merchandising Foundation Cloud Service Sales Audit				
	Sales Audit (REST)	SFTP	FTS	File Download (REST)	Publish to File System Folder
Cloud	Yes[1]	Yes[2]	Yes[3]	Yes	No
On-Premise	Yes[4]	No	Yes[5]	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+
4 Xstore Office On-Premise integration with Sales Audit REST API only available in RMFCS V21+. OAuth credentials to be obtained from Oracle Retail Home.
5 Xstore Office On-Premise integration with FTS only available in RMFCS V21+ . OAuth credentials to be obtained from Oracle Retail Home.

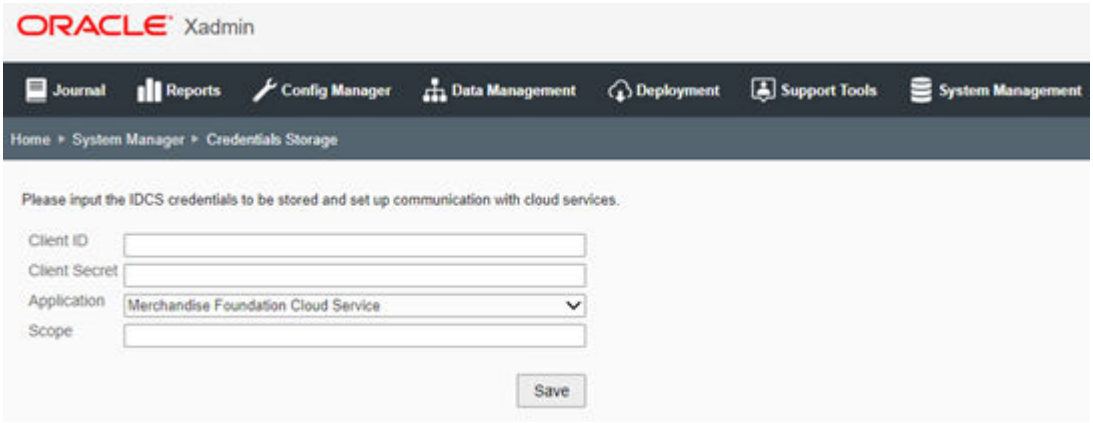
The details of the integration are covered in the remaining sections of this guide:

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

Xstore Office [On-Prem Only]: OAuth Credentials

The direct service MFCS integration, Sales Audit REST integration, and RTLog FTS integration require the use of OAuth client credentials in order for Xstore Office to communicate with MFCS/Sales Audit. Credentials should be obtained through the use of Oracle Retail Home. Once obtained, the Client ID, Client Secret, and Scope, should be submitted into Xstore Office using Xadmin's Credentials Manager.

Figure 1-1 Credentials Storage



2

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.

- RMFCS/RPCS Direct Services (MFCS version 21.x and above)
- OCDS (MFCS v19)
- Flat Files (deprecated)

RMFCS/RPCS Direct Services and OCDS integration styles utilize webservices to obtain merchandising and pricing data that is written into files using the dataloader's native .mnt file format.

Integrations with either provider can be established by using Xadmin’s Integration Manager UI for an Organization. See the *Oracle Retail Xstore Office User Guide* or the *Oracle Retail Xstore Office Cloud Service User Guide* for details about configuring these integrations using Xadmin’s Integration Manager.

Flat File integration style utilized files exported from Merchandising Foundation Cloud Service and Pricing Cloud Service.

Using RMFCS/RPCS Direct Services (MFCS version 21.x and above)

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.

> **Note**
>
> The Direct Web Services were introduced in RMFCS and RPCS in version 21 to avoid the overhead of replicating the data in another data hub (OCDS) and then having web services build on top of OCDS. OCDS was decommissioned in version 21 and RMFCS and RPCS started exposing the same webservices. Refer to the following documents for additional details.
>
> - *Oracle® Retail Merchandising Cloud Services Inbound and Outbound Integration Guide* on the Oracle Help Center
> - Reference paper *Publish APIs - Customer Enablement Overview* in the Oracle Retail Merchandising Documentation Library (Doc ID 1585843.1) on My Oracle Support

Using OCDS (MFCS v19)

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

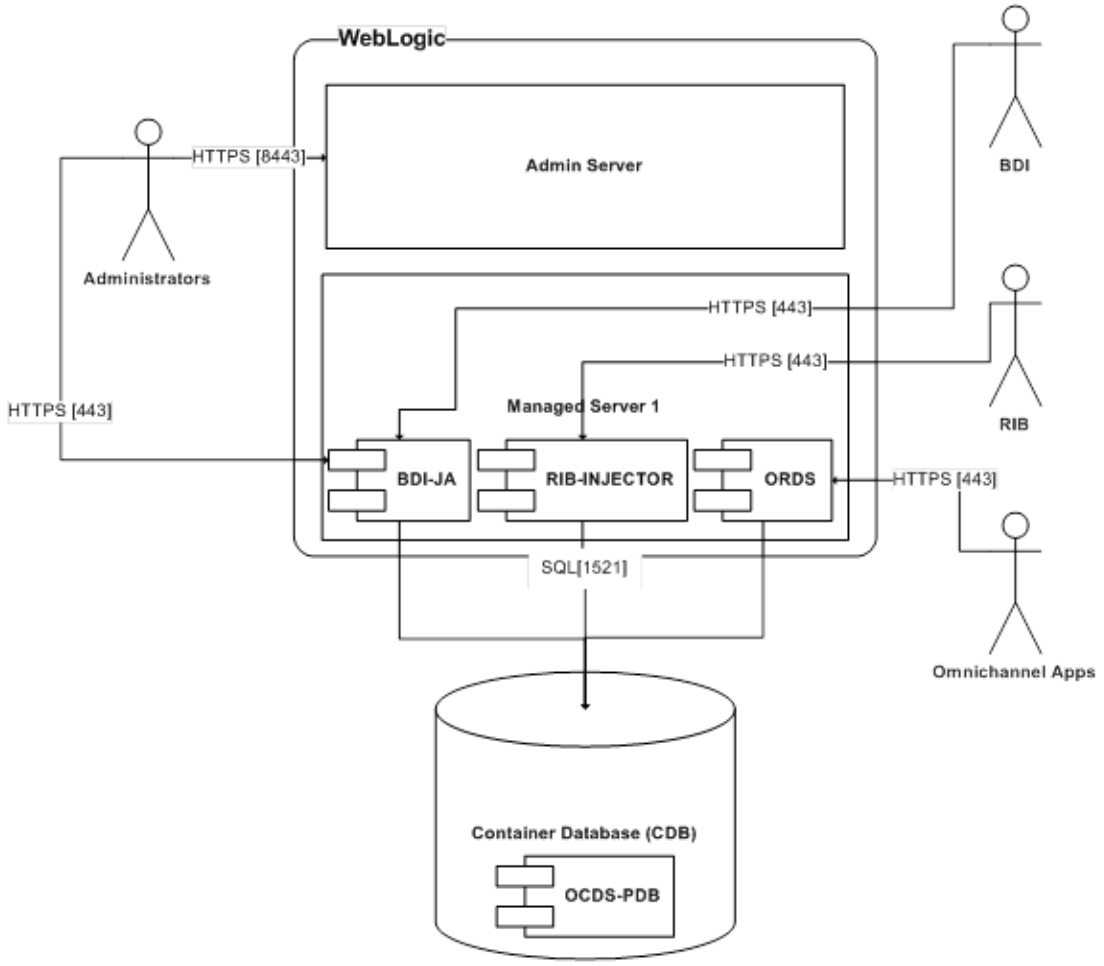
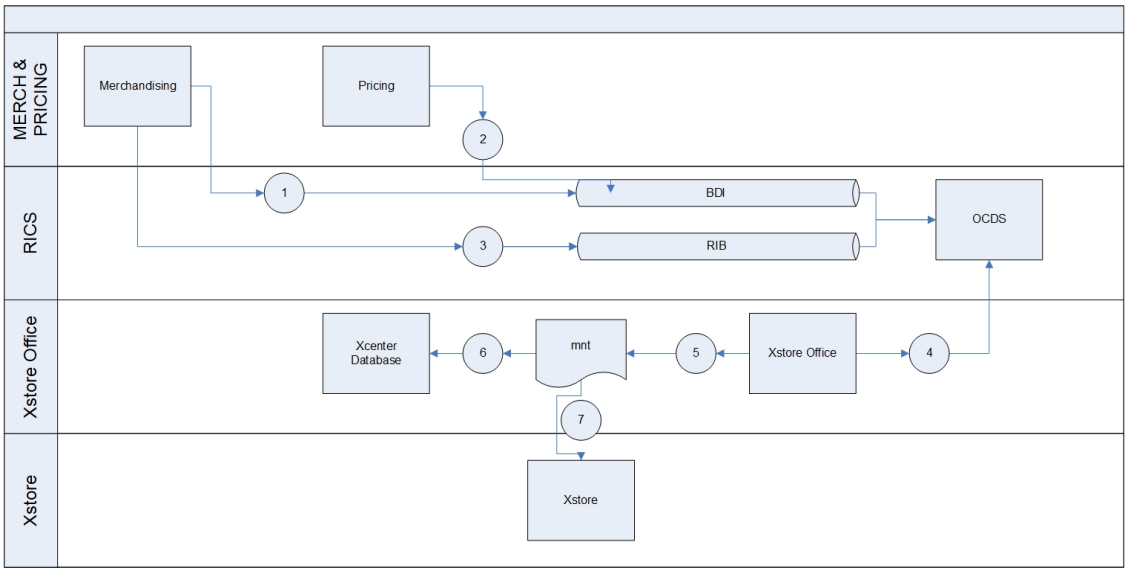


Figure 2-2 Conceptual Data Flow from Merchandising and Pricing to Xstore POS



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. For foundation data, this is generally a one-time push of data over BDI into OCDS. Incremental foundation data flows into OCDS from the RIB.

   > **Note**
   >
   > It is important to understand that the receipt of Merchandising Foundation data using BDI will trigger Xstore Office to request a full refresh of all data from OCDS because Xstore Suite will detect that the system of record, OCDS, has been refreshed. Ideally, OCDS receives its initial load before the integration is established in Xadmin. If Xadmin has an active OCDS integration and OCDS needs to be refreshed, then the OCDS integration should be either paused or deleted until all BDI jobs have been completed successfully to refresh OCDS. Once all BDI jobs have completed to refresh OCDS, then the integration can be recreated or unpaused, allowing Xstore Office to request initial load to replace foundation data in Xcenter and Xstore databases.

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.

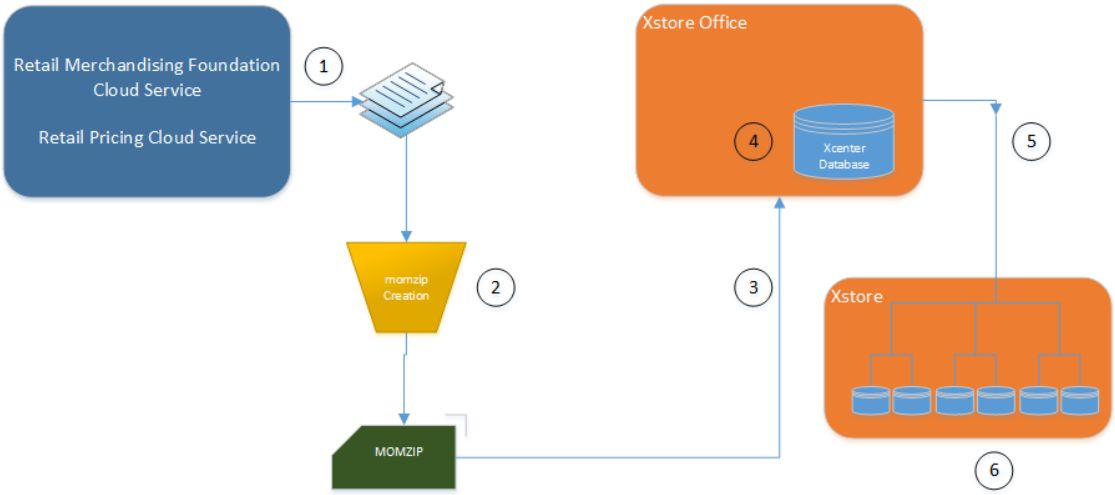
> **Note**
>
> OCDS leverages Basic Auth to authenticate web service requests made by Xstore Office. Retailers should develop procedures to proactively change the OCDS password on both sides of the integration before the OCDS Basic Auth password expires.

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

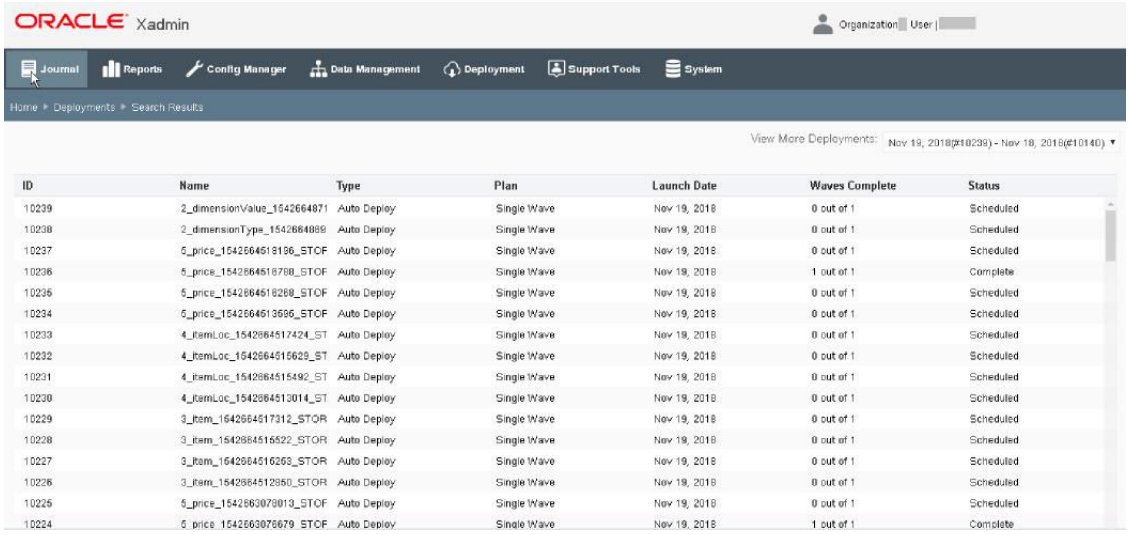
Each of the types of integration discussed in this guide involve the delivery of merchandising and pricing data, in file format, to Xstore Office's auto-deploy folder. Xadmin polls the auto-deploy folder at a configured interval and automatically processes new files. The polling-interval is configurable, 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



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

Table 2-1 (Cont.) MNT Files

Filename Prefix	Store Specific	Destination	Downlaod Time
5_clrpc	Yes	ALL	IMMEDIATE
5_price	Yes	ALL	IMMEDIATE
5_promo	Yes	XSTORE_ONLY	STORE_CLOSE
5_regp	Yes	ALL	IMMEDIATE

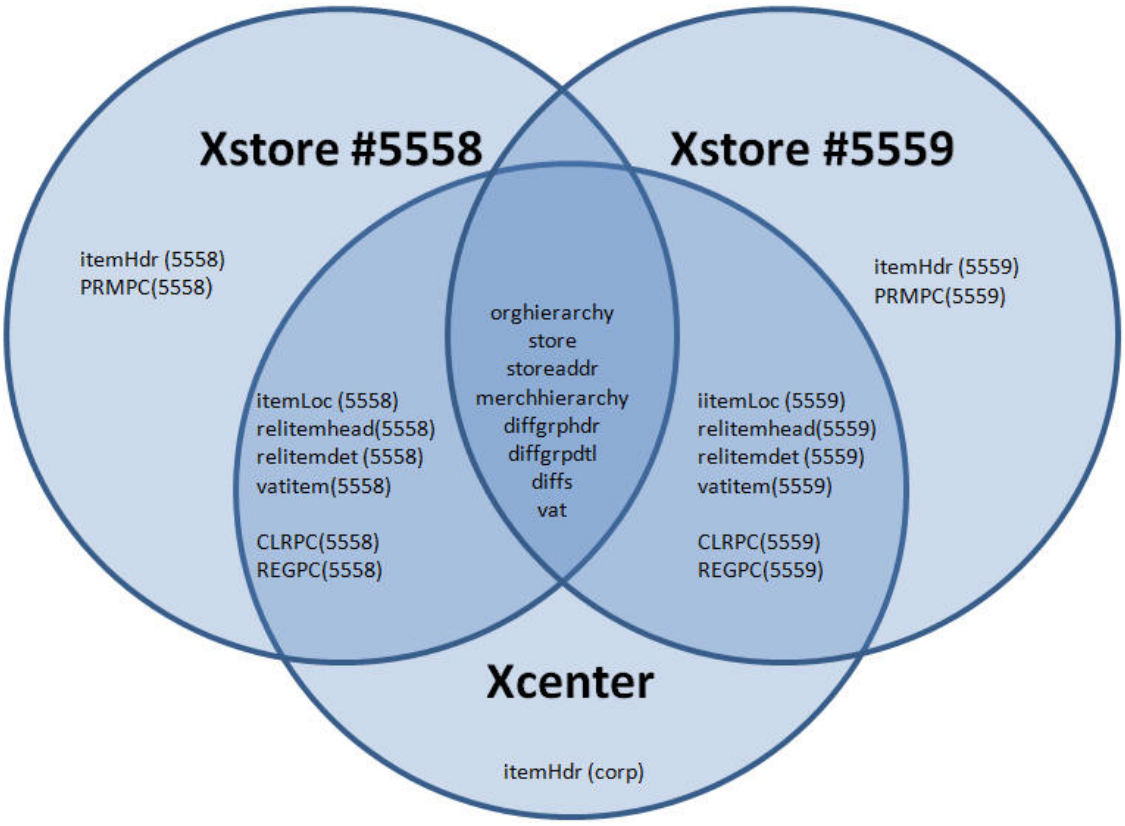
Using Flat Files

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

The figure below 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 Retails 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.

File Type Detection

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

Table 2-2 (Cont.) File Loading Dependency

File Type	Depends on
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.

Full Refresh

The integration styles discussed in this document support use-cases for **full refresh** and **incremental updates**.

Using RMFCS/RPCS Direct Services or OCDS

“January 1, 1970” is used by integration jobs to request all data from the source system. This exact date indicates that Xstore Office is requesting a full refresh, rather than requesting incremental changes. This date is called the “Since” date because when Xstore Office requests data it is requesting changes made to data in the source system “since” a specific point in time.

“January 1, 1970” is the default "Since” date used by both the Xadmin User Interface (UI) for configurating the integration, and the UI for defining an “on-demand job” in Xadmin’s Data Manager - Data Publisher feature. Xadmin enables users to define a different “Since” date. Retailers may choose to use a more recent “Since” date for use cases where the Xstore Suite databases already contains data that matches the source system up to a more recent date. However, if the purpose is to request a full refresh then the use of exactly January 1, 1970 is recommended.

Regardless of the “Since” date used for the integration’s initial schedule job, all subsequent scheduled jobs will request only incremental changes because the “Since” date is automatically determined by the last successful run of the schedule job.

Using Flat Files

RMFCS/RPCS support extract scripts to generate both FULL (full refresh) and delta (incremental) processing.

New Stores

Using RMFCS/RPCS Direct Services or OCDS

When a new store is added in the source system to a chain that is included in an active integration, the new store will automatically flow, as an incremental change, to the Organization Hierarchy and Store tables in Xstore Suite databases. However, mnt files are not automatically generated and deployed to the new store. The decision of when seed the new store with foundation data is left up to the retailer, and is accomplished using Xadmin’s Data Manager>Data Publisher feature to request the full set of data for the new store from the source system.

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

Xstore Office 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. Use Xadmin's Data Manager > Data Publisher feature to request the full set of data for the added store from the source system.

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 Xstore Suite databases. Data from the source system can be targeted to databases either at store (Xstore), or Xstore Office, or both.

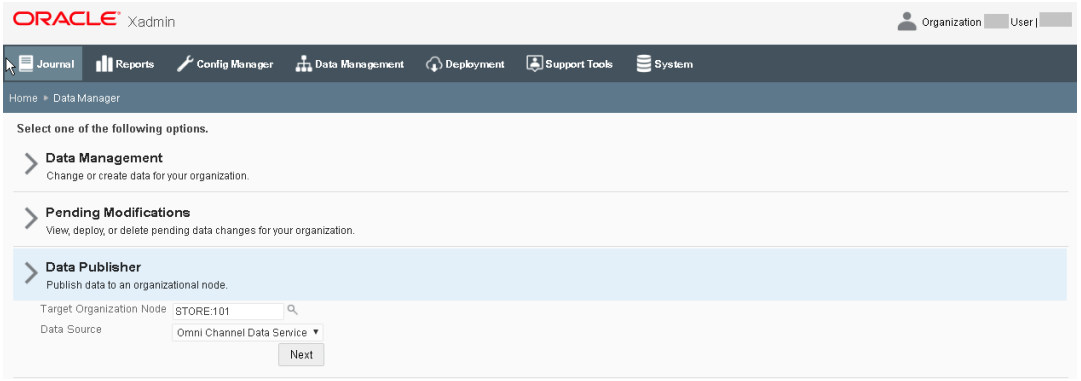
If “Purge Before Full Refresh” is enabled in the Integration’s configuration, use of the Data Publisher to replenish one or more types of Merchandising and Pricing data at a store will result in the purging of all existing sourced data, followed by the loading of a full set of the most recent data. When the integration is enabled in Xstore Office, the Data Manager 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:

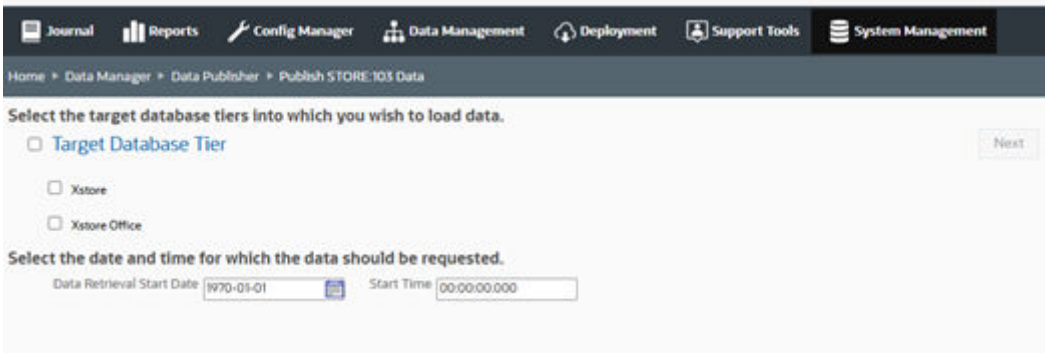
1. Choose the desired Organization Node for the target stores, select the integration, and click **Next.**

Figure 2-6 Xstore Office - Data Publisher Option



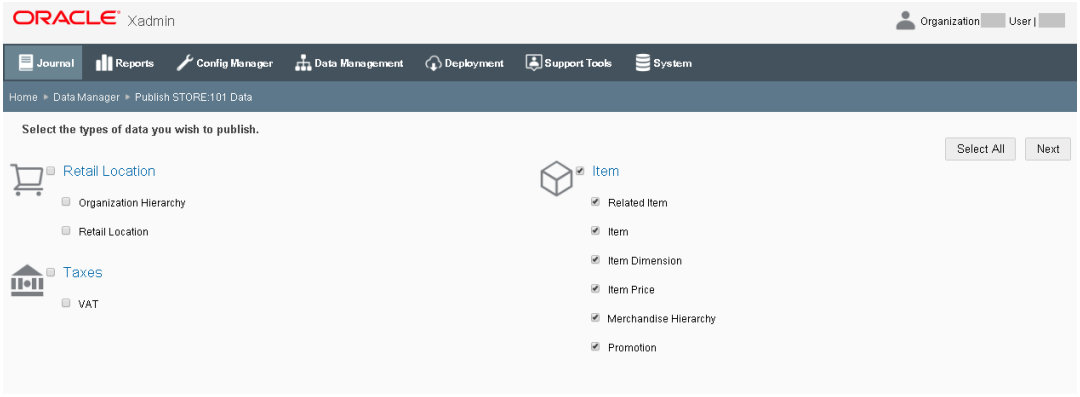
2. Choose the target database tier and the date from which data changes in the source system should be requested.

Figure 2-7 Xstore Office - Target Database Tiers



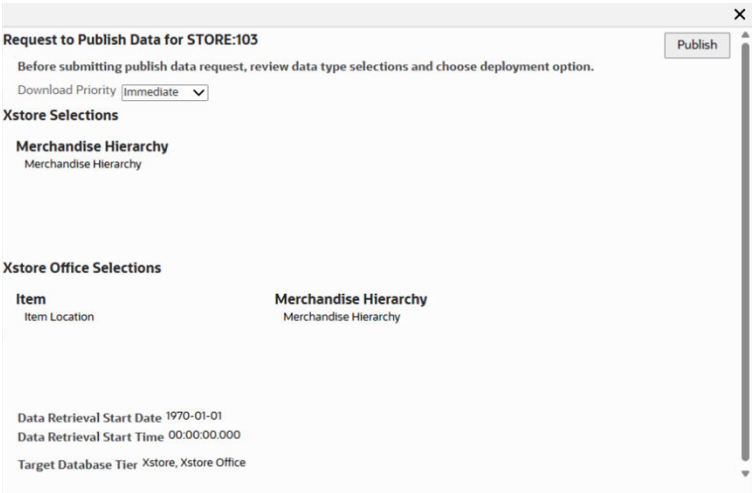
3. Choose the type of data you wish to publish for the database tier, then click **Next**.

Figure 2-8 Xstore Office - Data to Publish



4. Select a Download Priority, Immediate or Store Close, and click **Publish**.

Figure 2-9 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.

3

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

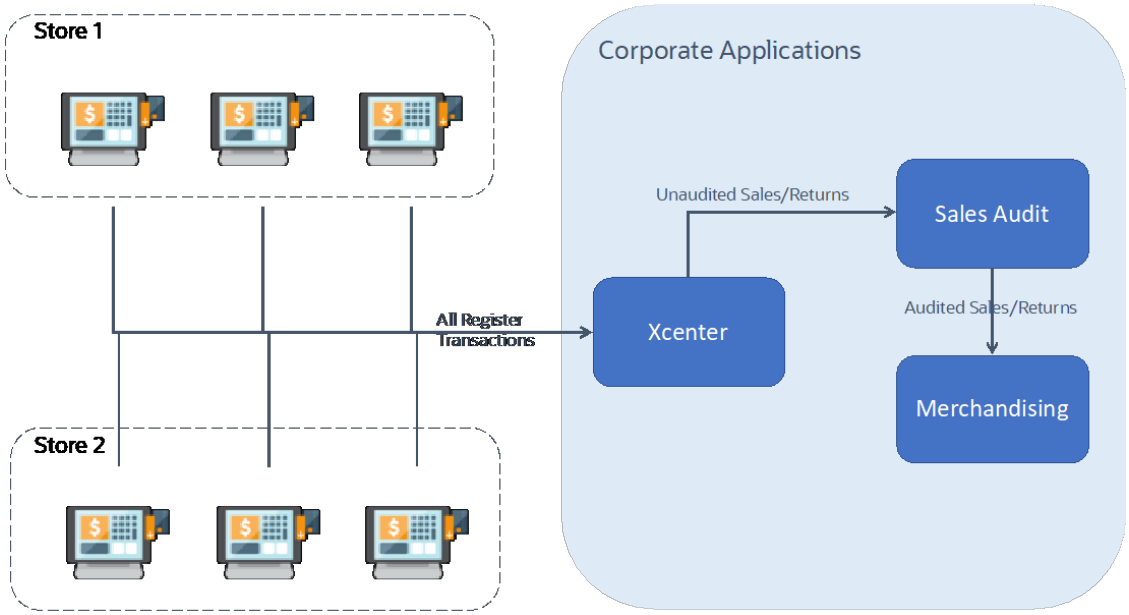
All transactions produced in Xstore are sent to Sales Audit. Sales Audit processing is primarily concerned with transactions that alter inventory or contain payment. Sales Audit loads other types of Xstore transactions (such as entering training mode, gift registry creation, and so on) into an OTHER 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. The direct integration to ReSA via the RESA Web Service is the go forward integration approach that should be used by customers. 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.

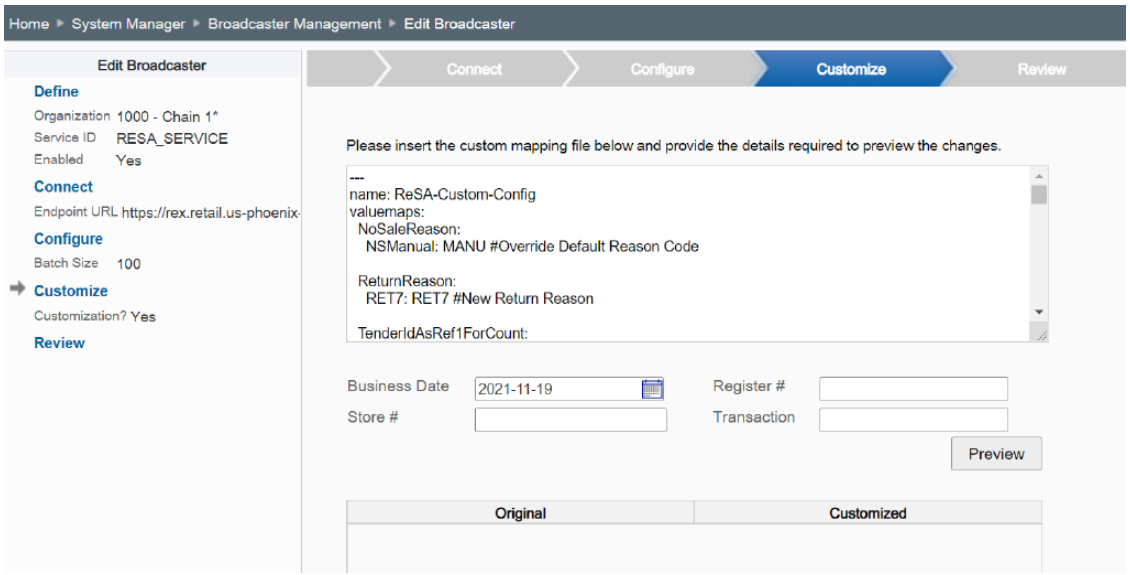
> **Note**
>
> The endpoint url is in the form https://<host>:<port>/<tenancyId>. Do not specify endpoint url with full service path like following: https://<host>:<port>/<tenancyId>/ResaRestServices/services/private/ReSA/salesService

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.

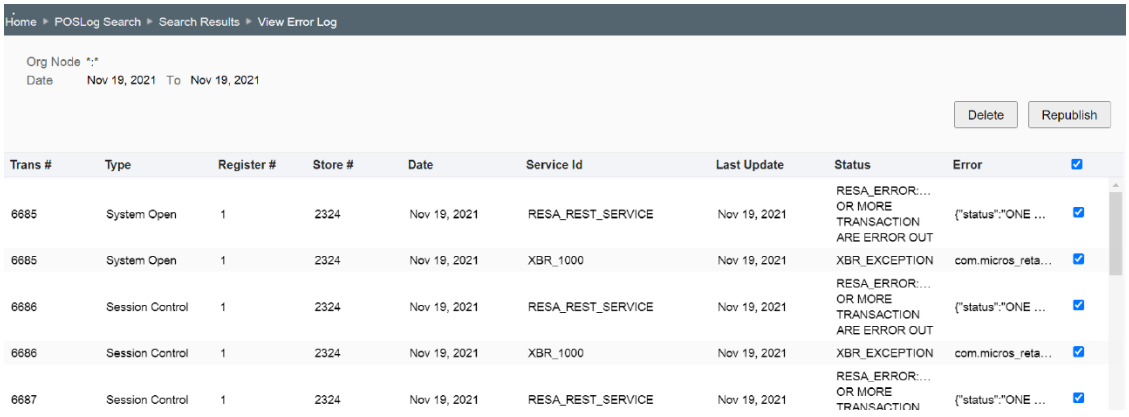
Scheduled Job to Query Sales Audit Rejected Transactions

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



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 RTLog Generator On-Premise.

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*

4

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

Configuration of the direct service integration involves self-service activities on both sides of the configuration. For Merchandising and POM configurations required for direct service integration from Merchandising and Pricing, see the sections Merchandising Configuration and POM Configuration below.

> **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

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*.

Xstore Office Configuration

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

Table 4-1 RMFCS Connection Configuration

Fields	Description
Endpoint URL	A URL describing protocol, host name, port and namespace of the Merchandising and Pricing Web Services. Example: https://XXX.XXXXXX.XX-XXXX-X.ocs.oraclecloud.com/rgbu-XXX-XXXX-XXXX-mfcs
Merchandising Chain ID(s)	A comma separated list of the chain ids in Merchandising containing stores in the Xstore Organization.
Oracle Retail Pricing Cloud Service	Choice of V1 or V2 RPCS APIs for regular price change, clearance, and promotion integrations.

Xstore Office: OCDS Integration Communication

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

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.

Table 4-2 OCDS Connection Configuration

Fields	Description
Endpoint URL	A URL describing protocol, host name, and port of the OCDS Web Services. Example: https://XXX-XXXX-ribcs-rics.oracleindustry.com
Service Path Prefix	The root service path for OCDS. Example: /ords/ocds/omnichannel
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: Additional OCDS/MFCS Configuration Options

Xadmin Integration Management feature includes the flowing configuration options which appear in the configuration wizard for both OCDS and MFCS configurations.

Table 4-3 Additional OCDS/MFCS Configuration Options

Setting	Description
Timeout Connect	Number of seconds to wait for connection.
Timeout Read	Number of seconds to wait for completion of response data.

Table 4-3 (Cont.) Additional OCDS/MFCS Configuration Options

Setting	Description
Enabled Data Types	The types of data to be requested by Xstore Office. Once data starts flowing this selection should not be changed. Typically all data types are enabled unless not using VAT or (very uncommon) price/promotion.
Retain Job History	The number of days to retain Job History data in the Xadmin database.
Scheduled Job Interval Minutes	This is the frequency with which Xadmin requests data from OCDS/MFCS. The value should be greater than or equal to 30 minutes.
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.
Orphan Data Protection Offset Seconds	This is the minimum amount of time data must age before it can be visible to Xstore Office. 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.
Records Per Request Limit	This specifies the number of records to request in webservice calls. If no limit is defined then MFCS/OCDS will determine the maximum number of records.
Retail Location: Till Accountability	This specifies if Retail Locations/Stores created from data are to use Till Accountability or not.
Retail Location: Default Locale	This specifies the Locale to use in Retail Locations created from 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
Item: Tax Group ID for non-taxable Items	This specifies the Tax Group ID assigned to non-taxable items created from MFCS/OCDS data.
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.
VAT Rounding Code	Specifies how the system will round when calculating the tax amount.
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.
VAT Rounding at Transaction Level	Specifies the ability to round at transaction level when calculating the tax amount. When the calculated tax will be rounded at the transaction level. When unchecked, 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 Cloud Services 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.

Oracle Retail Pricing Cloud Service Version

The Xstore Suite gives a customer the choice of using V1 or V2 RPCS APIs for regular price change, clearance, and promotion integrations. V1 is deprecated. V2 is backward compatible and serves as the default for new integrations. Existing integrations will continue to use V1 until the customer decides to switch through Xadmin integration UI.

V2 provides additional support for

- Promotion Distribution Rule, see the Promotions section for more details

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 Service's Integration Manager, use the Configure tab when completing the wizard UI to establish the integration.

Migration to MFCS

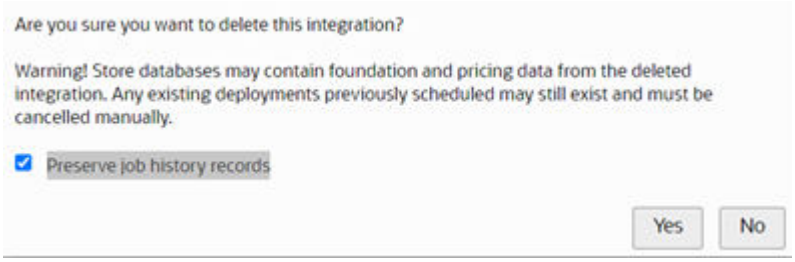
By default on the initial run of the MFCS integration’s scheduled job, Xstore Office will request all data from the source system that has changed since the Unix epoch (January 1st, 1970 at

00:00:00 UTC). Use of the Unix epoch effectively means: “request all data changes since the beginning of time”. Alternatively, a more recent point in time can be configured when creating a new MFCS integration in Xadmin’s Integration Manager.

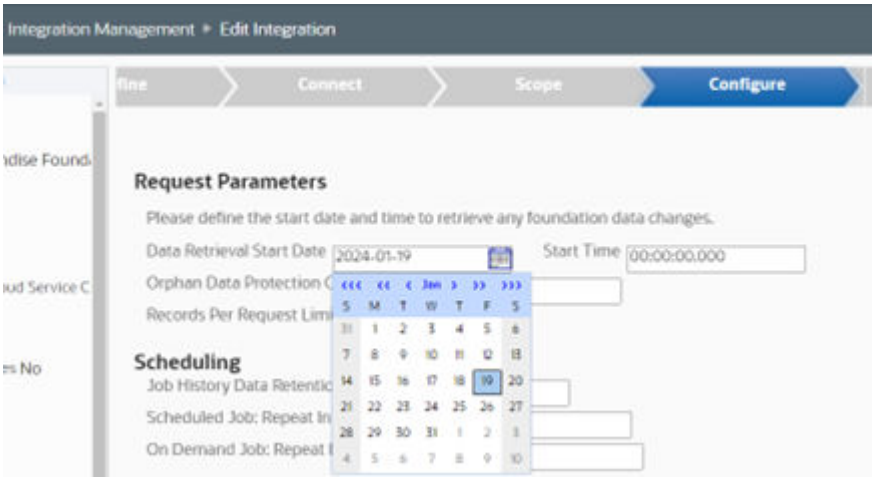
Migration: OCDS to MFCS

Two steps can be taken to avoid a requesting a full-refresh of MFCS data, and the replacement of all OCDS data, when a newly created MFCS integration’s first scheduled job is run.

1. Select “Preserve job history records” when deleting the OCDS integration.



2. Set the Data Retrieval Start Date and Start Time on the Configure tab when defining the MFCS integration to the point in time that ensures that Xstore Office will request MFCS data that primarily is not already duplicated by data pulled from OCDS. To assist in determined the optimal point in time, the UI fields is populated with the last time the OCDS integration successfully ran a schedule job (if step #1 was used).



Migration: Non OCDS Merchandsing System to MFCS

Retailers with an existing enterprise populated from an old merchandising/pricing system may prefer the integration with the new system begin from a much more recent point in time. Set the Data Retrieval Start Date and Start Time on the Configure tab when defining the MFCS integration to the point in time that ensures that Xstore Office will request MFCS data that primarily isn’t already duplicated by data pulled from the old system.

Purging on Full Refresh

The MFCS/OCDS integration can generate (purge) and deploy mnt files containing instructions to delete merchandising and pricing data in two use-cases:

1. When a scheduled job has no job history.
2. When an on-demand job is requested from Xadmin's Data Publisher feature.

Select the desired **Purge Integration Data** checkboxes in Integration Manager's Configure tab when defining the integration to enable the generation of purge files.

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.

5

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

> **Note**
>
> The volume of data used in the integration can provide operational challenges if the time required to import and deploy data to stores takes longer than desired. This is generally only a concern when number of items and number of stores are very large, and is mostly a concern when processing initial loads.

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: 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: 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 financial 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.

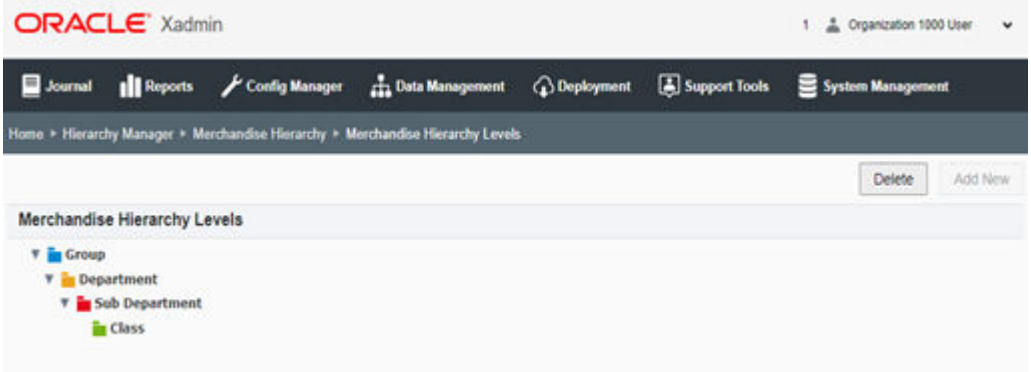
All stores in Merchandising that you expect to get a sales file from Xstore should be set up to have Integrated Sales flag set to checked (Yes) as part of the store setup. Additionally, the store level attribute Unique Transaction Number should be set to Register for all Xstore

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 Levels

Xstore supports up to four levels of merchandise hierarchy. This means that up to four of the lowest level Merchandise Hierarchy Levels used by the Merchandising System can be mapped to the Xstore Suite Merchandise Hierarchy Levels. The default Merchandise Hierarchy Levels used by Xstore (Department, Sub Department, Class, Subclass) do not match the bottom four levels of the merchandise hierarchy from Merchandising, which are Group, Department, Class, and Subclass. Retailers should use the Merchandise Hierarchy Level feature under Xadmin's Hierarchy Manager to configure the desired mapping. The most common mapping is either four levels (Group, Department, Class, and Subclass) or three levels (Department, Class, and Subclass). It is critical that this mapping is established before the integration is activated and any data is imported into the Xstore Suite from the Merchandising system. It is also critical that the Merchandise Hierarchy Levels are not changed once item data has been imported.

Figure 5-1 Merchandise Hierarchy Levels



Merchandise Hierarchy Identifiers

Xstore requires the use of unique indentifiers across all levels of the merchandise hierarchy. Merchandising does not have this requirement, so to create uniqueness the integration transforms keys imported from merchandising by appending a character representing the level as a suffix. An example would be Department 10 in Merchandising becomes Department 10D in Xstore.

Additionally in Merchandising, the ID displayed for Class and Subclass are not by themselves unique, these IDs are only unique in the context of their parents' ID. In Xstore, and some other Point Of Service systems, merchandise hierarchy IDs must be unique within the same level. Merchandising enables integration with Point Of Service systems having this requirement by also maintaining a unique key for Class and Subclass -- the unique key is held in the Merchandising tables for class and subclass. This unique value is not visible to users of Merchandising. The Xstore Suite imports only the unique key for use in Class and Subclass identifiers.

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.

Unit of Measure Items

Merchandising allows a customer to define a UOM for an item at retail location level. In addition, the number of fractional digits allowed for the UOM item quantity can be specified. An example would be to define KG (kilogram) as the UOM with 2 fractional digits for item xyz at location abc.

They are both mapped to Xstore’s item options. Once deployed Xstore will prompt the user to enter a quantity for the item during a sale, and the quantity can contain fractional digits up to the defined number of fractional digits allowed.

Other Item Attribute Notes

The set of data entity attributes managed by the Xstore Suite and MFCS overlap but are not identical. Some data and fields supported by Xstore can not be obtained from Merchandising, and some data and attributes are imported but require transformation to bridge differences in the two systems. The following is not an exhausted list but does call out some key differences:

- 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 (CFAS). 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.

- Merchandising Deposit item attributes are not supported in the integration.
- Merchandising Item Images are not supported in the integration.
- Item Availability – The ITEM_AVAILABILITY_CODE column in Xstore's itm_item_options table is populated from Merchandising's ItemLocation's STATUS attribute using the following rules:
  - When STATUS is "A" then itm_item_options.ITEM_AVAILABILITY_CODE is set to AVAILABLE.
  - When STATUS is "C" then itm_item_options.ITEM_AVAILABILITY_CODE is set to AVAILABLE and itm_item_options.STOCK_STATUS is set to DISCONTINUED.
  - When STATUS is "I" or "D" then itm_item_options.ITEM_AVAILABILITY_CODE is set to NA.

  However, if Merchandising's ItemLocation's STOPSALEIND attribute is "Y", then Xstore's itm_item_options.ITEM_AVAILABILITY_CODE is set to RECALL.

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.

Global Tax (GTS)

Merchandising integration includes tax data setup provided by the Global Tax Solution (GTS) in case the tax solution system option is set in MFCS. The data is provided exclusively via direct Web Services that can be verified in the *Oracle® Retail Merchandising Cloud Services Inbound and Outbound Integration Guide* on the Oracle Help Center.

> **Note**
>
> GTS tax data setup is published via REST services from MFCS irrespectively of the destination system. Xstore consumption of that data is to be made available. Field enablement is required until then.

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.

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

OCDS/MFCS: Unlike Xstore, Merchandising does not have a non-store-specific serialized item flag, so the integration infers this information. Merchandising does have a uinType item-level field that is store-specific. XOCS uses this field to recognized an item as “a serialized item” when receiving item header data for a store. The Xstore Suite’s serialized item flag attribute is stored in its itm_item table. This table is not a store-specific item attribute table, though the data maintained in the table could represent the assortment of a single store, or the assortment of all stores.

- A store’s database’s itm_item is populated by the results of requests from MFCS/OCDS for item data for a specific store. The response includes only items in that store’s item-assortment AND an item’s store-specific uinType (if one exists). This allows Xstore to properly set the serialized item flag at the store in the itm_item table
- – When Xcenter’s database’s itm_item table is populated by the results of requests from MFCS/OCDS, item records represent the assormtment in all stores and are stored at the *:* level. Since the context of this data is not specific to any one store the concept of a store-specific uinType doesn’t apply, and the serialized item flag will be false.

Flat File: The serialized flag can vary by location for an item in Merchandising, the last location to be dataloaded by the integration code sets the item level serialized flag in the Xcenter database. Since a store database is populated with the item assortement for only for a single store, the serialized flag will be appropriate set.

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 it 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 place 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 synch 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.

Initial Price

An “Initial Price” from the Merchandising system is persisted in Xstore Suite databases with an effective date of January 1, 1970 at midnight UTC.

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.

The following are features supported with the integration of V2 RPCS. They should be configured off in Pricing when integrating with V1 RPCS.

Pricing Application - Promotion Offer Distribution Rules (OFDR)

Pricing allows users to define how the rewards should be distributed to items on a customer's purchase by selecting a distribution rule as part of the promotional offer creation. The possible values are:

- G: distribute rewards to Get items.
- BG: distribute rewards to Buy and Get items.
- B: distribute rewards to Buy items.

Xstore only supports case 1 and 2. To support case 2 or a mix of case 1 and 2, Xstore config Deals---DisplayProportionDealAmt has to be set to true. If left as false, which is the default, only case 1 works in Xstore.

Case 3 is not supported in Xstore. Any promotion created with distribution rule B in Pricing is converted to BG during the integration. The following warning will be logged.

"Distribution rule code B in offer xxx:xxx is unsupported. Default to BG to distribute rewards to both buy and get items."

If the distribution rule is left as unspecified, it will be defaulted to BG during the integration.

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.

6

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 Merchandising/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 a Jetty 12 or Tomcat 11.

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

Installation

Run Xstore Office Jetty or Tomcat installer to install RTLog Generator. Copy the rtlog-generator.war file to the installation directory for the installer to detect the war file and copy it to the server's wabapps directory.

Refer to *Oracle Retail Xstore Suite Implementation and Security Guide* for information on Jetty and Tomcat Xstore Office installers.

The installer does not copy RTLog generator's configuration zip file. Follow the steps in the next Configuration section to manually configure RTLog generator.

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 Jetty or Tomcat 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: `-DRTLOG-GENERATOR_CONFIG=C:/<custom_directory>/`

     For Jetty, specify the JVM argument in the wrapper.conf file under <JettyServer>/bin directory.

Example:

```
wrapper.java.additional.16=-DRTLOG-GENERATOR_CONFIG=C:/rtlog-generator-config-2
```

For Tomcat, specify the JVM argument in catalina.sh or catalina.bat under <TomcatServer>/bin directory.

Example:

```
JAVA_OPTS="$JAVA_OPTS -DRTLOG-GENERATOR_CONFIG=C:/rtlog-generator-config-2"
```

- 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 6-1 Example of context-param Field Update

```
<?xml version="1.0" encoding="UTF-8"?>
<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">
    <display-name>RTLOG-GENERATOR</display-name>
    <context-param>
        <param-name>contextConfigLocation</param-name>
        <param-value>/WEB-INF/classes/applicationContext.xml</param-value>
    </context-param>
    <!-- Customizable external location for RTLog config files-->
    <context-param>
        <param-name>rtlog.generator.config.home</param-name>
        <param-value>C:/custom_directory/</param-value>
    </context-param>
    <!--Customizable RTLog generator app_name. If not unspecified, it remains "rtlog-generator". This
    <context-param>
        <param-name>rtlog.generator.application.name</param-name>
        <param-value />
    </context-param>
    <!--Customizable external log4j xml file. Specify just the file name without any extension. By de
    <context-param>
        <param-name>rtlog.generator.config.log4j</param-name>
        <param-value />
    </context-param>
```

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. In a cluster environment, the directory must be shared by all the instances in the cluster.

     – `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. In a cluster environment, the directory must be shared by all the instances in the cluster.

     – `clusterNodeNumber`: This property should only be enabled when running in a clustered environment. For more information, see the Running Multiple RTLog Generator instances in a Cluster section.

     – `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.

– `spring.profiles.active:` This property is by default commented out. Uncomment it in the following two scenarios.

  a. Enable FTS delivery of RTLog files.

     * Set it to “fts,archive” without quotes to send RTLog files to ReSA CFS object storage through FTS (File Transfer Service).

     * Set it to “fts,zip,archive” without quotes to send RTLog zip files to ReSA CFS object storage through FTS (File Transfer Service).

  b. Set it to “zip” without quotes to zip up RTLog files placed in the directory specified by property “resaFileDropDir”.

Following is an example of the properties. No spring.profiles.active property is specified in this case.

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

Following is another example to enable zipping of RTLog files.

```
processingDir = C:/RTLogGen/Output/Store/RTLOGS
resaFileDropDir = C:/RTLogGen/Output/ReSA
clusterNodeNumber = 1
deliveryTarget = resa-cs
-OR-
deliveryTarget = resa-onprem
spring.profiles.active = zip
```

Additional properties and configuration have to be specified if FTS delivery of RTLog files is enabled. Skip this section if it is not enabled.

– `cloud.fts.endpointUrl:` This property specifies the ReSA FTS endpoint URL to deliver the RTLog files to.

> **ⓘ Note**
>
> RTLog Generator supports two forms of RMS FTS endpoints.
>
> a. RMS Platform FTS wrapper API: This API is the new FTS endpoint supported in RMS. The customer is advised to use this endpoint for FTS integration. Its URL takes the following format: https://xxxx/xxxx/RmsPlatformServices/services/private/FTSWrapper
>
> b. RMS FTS API (deprecated): This API is deprecated but still supported in RMS. The customer is advised to switch to use the new platform FTS wrapper API. The deprecated API may no longer be available in RMS in the future. Its URL takes the following format: https://xxxx/xxxx/RmsReSTServices/services/private/fts

To invoke ReSA FTS endpoint URL secured in Oauth, IDCS client ID, secrete, and scope are required. They are captured in Xcenter DB. The following properties must be specified for RTLog generator to request them through Xcenter on premise REST APIs.

- xcenter.protocol: http or https
- xcenter.host: Xcenter on premise host name
- xcenter.port: Xcenter on premise port
- xcenter.db.user: encrypted basic auth username for Xcenter on premise REST apis
- xcenter.db.pwd: encrypted basic auth password for Xcenter on premise REST apis
- dtv.location.organizationId: Xcenter organization id to request for IDCS client id/secret/scope
- dtv.CustomerId: customer ID used in decryption of username and password
- dtv.CustomerId.salt: salt used in decryption of username and password

Following is an example rtlogconfig.properties with FTS delivery enabled.

```
processingDir = C:/RTLogGen/Output/Store/RTLOGS
resaFileDropDir = C:/RTLogGen/Output/ReSA
clusterNodeNumber = 1
deliveryTarget = resa-cs
spring.profiles.active = fts,zip,archive
-OR-
spring.profiles.active = fts,archive
cloud.fts.endpointUrl=https://xxxx/xxxx/RmsPlatformServices/services/
private/
FTSWrapper
xcenter.protocol=https
xcenter.host=<xcenter onprem host>
xcenter.port=<xcenter onprem port>
xcenter.db.user=<encrypted xcenter onprem basic auth username>
xcenter.db.pwd=<encrypted xcenter onprem basic auth passowrd>
dtv.location.organizationId=1000
dtv.CustomerId=xxx
dtv.CustomerId.salt=xxxx
```

Cipher Keys: for rtlog-generator to decrypt xcenter.db.user and xcenter.db.pwd, cipher keys must be copied from xcenter-config/res/keys to rtlog-generator-config/res/keys. This includes all key files of the following naming patterns.

– config.cip

– config.*.cip (for example config.2022-02-07.cip)

Truststore: certificate to communicate to Xcenter must be placed in .truststore file under rtlog-generator/res/ssl

- 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 6-2 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 ouf 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\classes\resa-cs for Sales Audit integrations (deliveryTarget = resa_cs) or WEB-INF\classes\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="PO1" RecordValue="PO1"/>
 <VALUE_MAPPING sourceValue="PO2" RecordValue="PO2"/>
 <VALUE_MAPPING sourceValue="PO3" RecordValue="PO3"/>
 <VALUE_MAPPING sourceValue="PO4" RecordValue="PO4"/>
 <VALUE_MAPPING sourceValue="PO5" RecordValue="PO5"/>
<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 Xstore Office installer is not used to lay down RTLog Generator, or re-deployment is necessary, follow the steps below to deploy or redeploy the application.

1. Stop the Jetty or Tomcat server.
2. Copy the rtlog-generator.war file to the webapps directory under Jetty or Tomcat server.
3. Start the Jetty or Tomcat server.

Running Multiple RTLog Generator Instances in a Cluster

In a cluster setup, each RTLog Generator instance must specify its own unique cluster node number in the property `clusterNodeNumber` in `rtlogconfig.properties` under its external configuration directory.

To meet this requirement for a vertical cluster, multiple RTLog Generator instances on a same physical machine must not share one default external configuration directory at C:\rtlog-generator-config or /usr/local/rtlog-generator-config. Each instance must specify its own external configuration directory. For more information on how to override the default external configuration directory, see the Configuration section.

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 Jetty or Tomcat. 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 Jetty or Tomcat server and importing the public certificate into Xstore Office's (client) truststore. Transport level security is configured by Xstore Office installer. No additional work is required.

Container Level Security

This section describes the Container Level Security.

Jetty

1. Go to <JettyServer>/etc directory. Open file jetty-xcenter.xml in a text editor.
2. Locate “Configure Xcener User Realm” section under the following heading.

```
<!-- =========================================================== -->
<!-- Configure Xcenter User Realm -->
<!-- =========================================================== -->
```

3. Add the highlighted section to introduce user realm `myrealm` for RTLog Generator. It should be inserted after the existing section (unhighlighted) for user realm `Xcenter Dtx`.

```
  <Call name="addBean">
    <Arg>
      <New class="org.eclipse.jetty.security.jaas.JAASLoginService">
```

```
        <Set name="name">Xcenter Dtx</Set>
        <Set name="LoginModuleName">xcenterLoginModule</Set>
        <Set name="roleClassNames">
          <Array type="java.lang.String">
            <Item>oracle.retail.xstore.passwd.jaas.SimpleRole</Item>
          </Array>
        </Set>
      </New>
    </Arg>
  </Call>
  <Call name="addBean">
    <Arg>
      <New class="org.eclipse.jetty.security.jaas.JAASLoginService">
        <Set name="name">myrealm</Set>
        <Set name="LoginModuleName">xcenterLoginModule</Set>
        <Set name="roleClassNames">
          <Array type="java.lang.String">
            <Item>oracle.retail.xstore.passwd.jaas.SimpleRole</Item>
          </Array>
        </Set>
      </New>
    </Arg>
  </Call>
```

4. Under the same directory, open file passwds in a text editor. The file contains username/password used for RTLog Generator basic auth. The password must be hashed. Follow the steps below to hash the password and setup a user.
   - Open a command prompt in <JettyServer>/lib directory.
   - Run `java -cp dtv-password.jar oracle.retail.xstore.passwd.impl.Ssha2Hasher`.
   - Type in the password you want RTLog Generator to accept and press enter
   - Collect the line of text from the console.
   - Add it in a new line to passwds containing the following format.

     ```
     the desired username>:<the output collected in last
            step>:92464520400:0:0:RTLogUserGroup
     ```

     The following is an example with `rtloguser` as the username.

     ```
     rtloguser:
     {SHA512}ac5a43e1282ba165c357ec6b9e4a935bd20a3a6d4abe9c76e48bbd650d29378d
     47678e23d82c983b437cd4b39151841c8f08948700b4e2c2f69978143d55f513$a539daf
     2a1c165b7$100000:92464520400:0:0:RTLogUserGroup
     ```

Tomcat

1. Go to <TomcatServer>/conf directory. Open the tomcat-users.xml file in a text editor.
2. Add a new line to define `RTLogUserGroup` role in the tomcat-users.xml file.

   ```
   <role rolename="RTLogUserGroup"/>
   ```

3. The file contains username/password used for RTLog Generator basic auth. The password must be hashed. Follow the steps below to hash the password and setup a user.
   - Open a command prompt in the <TomcatServer>/bin directory.
   - Run `./digest.sh -a SHA-512 -e UTF-8 -i 1000 -s 64 <the password you want to use>`
   - Collect the portion of the resulting output following `<the password you want to use>:`
   - Add a new line to the tomcat-users.xml file containing the following format:

     ```
     <user username="<the desired username>" password="<the
             output collected in last step>" roles="RTLogUserGroup"/>
     ```

     The following is an example with rtloguser as the username.

     ```
     <role rolename="RTLogUserGroup"/>
     <user username="rtloguser" password="<password>"
     roles="RTLogUserGroup"/>
     ```

7

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 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"
```

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/RTLogFormatConf ig" >
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/RTLogFormatConf ig"
```

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.

  > **Note**
  >
  > The RTLog Generator supports two forms of RMS FTS endpoints.
  >
  > 1. RMS Platform FTS wrapper API: This API is the new FTS endpoint supported in RMS. The customer is advised to use this endpoint for FTS integration. Its URL takes the following format: https://xxxx/xxxx/RmsPlatformServices/services/private/FTSWrapper
  > 2. RMS FTS API (deprecated): This API is deprecated but still supported in RMS. The customer is advised to switch to use the new platform FTS wrapper API. The deprecated API may no longer be available in RMS in the future. Its URL takes the following format: https://xxxx/xxxx/RmsReSTServices/services/private/fts

- 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 Protocoal	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://<rlog-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:idm:__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/RTLogMappingConfig"
```

A

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 ode types used by each of the applications that are used in this mapping:

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.

Table A-1 (Cont.) Transaction Type Mapping

Xstore Transaction Type	Sales Audit Transaction Type TRAT	Sales Audit Sub-Transaction Type TRAS	Description
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
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.

Table A-1 (Cont.) Transaction Type Mapping

Xstore Transaction Type	Sales Audit Transaction Type TRAT	Sales Audit Sub-Transaction Type TRAS	Description
	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
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.
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.
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

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**
	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	ManufacturerCoupon	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

Table A-3 (Cont.) Total Tender ID Mapping

Xstore		Sales Audit RTLog
TenderType	**TenderID**	**Total ID**
	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

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

Xstore Reason Code	Sales Audit Reason Code	Description
PAID_OUT	PO1	Stocks
PAID_OUT	PO2	Delivery
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

Table A-7 (Cont.) Discount Reason Codes

Xstore Reason Code	Sales Audit Reason Code	Description
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

Table A-9 (Cont.) Item Status/ and Sales Type Mapping

Xstore Item	Xstore Action	Sales Audit Item Status	Sales Audit Sales Type
	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 TOTTAX. 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) - incudes 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 retailTrnDetailrExportabilityMapper.

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 NonMerchandiceItem 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 ItemDicount 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 is 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.

B

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

Table B-1 describes the RMS Diff Group Detail mapping.

Table B-1 RMS Diff Group Detail 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 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

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	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	DisplaySeq	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

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 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
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

Table B-2 (Cont.) RMS Diff Group 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
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

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 DIFFS. Data will always be present in this field.	Character	20	NA	NA	NA	NA	Ignored.

Table B-3 (Cont.) RMS Diffs 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	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.
5	DiffType Desc	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.

Table B-3 (Cont.) RMS Diffs 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	Industry Code	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	Industry Subgroup	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
NA	NA	NA	NA	NA	rms_diff_ids	UPDATE_DATE	TIMESTAMP(6)	now()	NA

RMS Item Header

Table B-4 describes the RMS Item Header mapping.

Table B-4 RMS Item Header Mapping

Pos	RMS Field Name	RMS Description	RMS Field Type	RMSMax 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: ITEMHDRCRE 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_dimension_value, itm_item_dimension_type	ORG_CODE, ORG_VALUE	VARCHAR(30)	"*"	DimensionValue and DimensionType are only used for Style items.
3	Item	ID of the item. This field will always have data.	Character	25	itm_item, itm_item_dimension_value, itm_item_dimension_type	ITEM_ID, DIMENSION_SYSTEM	VARCHAR(60)	NA	DIMENSION_SYSTEM is populated with ItemID for Style items.

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

Pos	RMS Field Name	RMS Description	RMS Field Type	RMSMax 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.
6	PackInd	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. 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

Pos	RMS Field Name	RMS Description	RMS Field Type	RMSMax Width	Xstore Table	Xstore Column	Xstore Data Type	Xstore Default Value	Xstore Comment
7	SimplePackInd	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. 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

Pos	RMS Field Name	RMS Description	RMS Field Type	RMSMax Width	Xstore Table	Xstore Column	Xstore Data Type	Xstore Default Value	Xstore Comment
8	ItemLevel	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. The concept is best explained with the following typical (although not exhaustive) examples. 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). 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. Valid values are 1, 2, and 3. This field will always have data.	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	RMSMax Width	Xstore Table	Xstore Column	Xstore Data Type	Xstore Default Value	Xstore Comment
9	TranLevel	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. The concept is best explained with the following typical (although not exhaustive) examples. 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). Fashion styles generally have a TranLevel = 2 (the style itself is not sold/ inventoried). Fashion skus generally have an TranLevel = 2 (the fashion sku is sold/ inventoried). UPCs for fashion skus generally have a TranLevel = 2 (the fashion sku is sold/ inventoried).	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	RMSMax 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	Inventory Ind	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

Pos	RMS Field Name	RMS Description	RMS Field Type	RMSMax Width	Xstore Table	Xstore Column	Xstore Data Type	Xstore Default Value	Xstore Comment
11	Diff1Level	This field describes whether the Diff1 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 Diff1Type.
12	Diff1Type	This field contains the type of the Diff1 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	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	RMSMax 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

Pos	RMS Field Name	RMS Description	RMS Field Type	RMSMax Width	Xstore Table	Xstore Column	Xstore Data Type	Xstore Default Value	Xstore Comment
15	Diff2Type	This field contains the type of the Diff2 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	itm_item_dimension_type	DIMENSION	VARCHAR2(30 CHAR)	NA	Only used when Diff2Level is provided.
16	Diff2	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). Data is optional in this field for many items.	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	RMSMax Width	Xstore Table	Xstore Column	Xstore Data Type	Xstore Default Value	Xstore Comment
17	Diff3Level	This field describes whether the Diff3 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 the RMS DIFF Group Details to create the Dimension System information. When ID, DataLoader fetches the Default DIFF Group for the Diff3Type.
18	Diff3Type	This field contains the type of the Diff3 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	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	RMSMax 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

Pos	RMS Field Name	RMS Description	RMS Field Type	RMSMax 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	RMSMax Width	Xstore Table	Xstore Column	Xstore Data Type	Xstore Default Value	Xstore Comment
24	Class	Number identifying the class in the merchandise hierarchy to which the item belongs. If the item has a parent, the item's class will be the same as that of its parent. This field will always have data for the ITEMHDRCRE and ITEMHDRMOD records.	Number	4	NA	NA	NA	NA	Ignored.
25	UniqueClass	Number uniquely identifying the class node to which the item belongs. 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. This field will always have data for the ITEMHDRCRE and ITEMHDRMOD records.	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

Pos	RMS Field Name	RMS Description	RMS Field Type	RMSMax Width	Xstore Table	Xstore Column	Xstore Data Type	Xstore Default Value	Xstore Comment
26	Subclass	Number identifying the subclass in the merchandise hierarchy to which the item belongs. Subclass ID is not unique in the level of the merchandise hierarchy. The combination of Dept/ Class/Subclass is unique. If the item has a parent, the item's class will be the same as that of its parent. This field will always have data for the ITEMHDRCRE and ITEMHDRMOD records.	Number	4	NA	NA	NA	NA	Ignored.
27	UniqueSubclass	Number uniquely identifying the subclass node to which the item belongs. 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. This field will always have data for the ITEMHDRCRE and ITEMHDRMOD records	Character	10	itm_item	MERCH_LEVEL_4 (or configured level)	VARCHAR2(60 CHAR)	NA	S is always appended to the value.

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

Pos	RMS Field Name	RMS Description	RMS Field Type	RMSMax 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	SecondaryItemDesc	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	RMSMax Width	Xstore Table	Xstore Column	Xstore Data Type	Xstore Default Value	Xstore Comment
33	MerchandiseInd	Indicates if the item is a merchandise item (Y, N). 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). 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 the record is skipped.
34	PrimaryRefItemInd	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.

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

Pos	RMS Field Name	RMS Description	RMS Field Type	RMSMax Width	Xstore Table	Xstore Column	Xstore Data Type	Xstore Default Value	Xstore Comment
35	CostZoneGroupId	Cost zone group associated with the item for ELC calculations. Data is optional in this field.	Number	4	NA	NA	NA	NA	Ignored.
36	Standard UOM	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.
37	UOMConvFactor	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.

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

Pos	RMS Field Name	RMS Description	RMS Field Type	RMSMax Width	Xstore Table	Xstore Column	Xstore Data Type	Xstore Default Value	Xstore Comment
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.
40	StoreOrd Multi	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	Forecastlnd	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.

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

Pos	RMS Field Name	RMS Description	RMS Field Type	RMSMax Width	Xstore Table	Xstore Column	Xstore Data Type	Xstore Default Value	Xstore Comment
42	OriginalRetail	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	OriginalRetailCurrencyCode	The currency code of the OriginalRetail. Data is optional in this field.	Character	3	NA	NA	NA	NA	Ignored.
44	MfgRecRetail	Data is optional in this field.	Number	20	itm_item	LIST_PRICE	NUMBER(17,6)	NA	NA
45	MfgRecRetailCurrencyCode	The currency code of the MfgRecRetail. Data is optional in this field.	Character	3	NA	NA	NA	NA	Ignored.
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.

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

Pos	RMS Field Name	RMS Description	RMS Field Type	RMSMax Width	Xstore Table	Xstore Column	Xstore Data Type	Xstore Default Value	Xstore Comment
48	ItemAggregateInd	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.
49	Diff1AggregateInd	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). 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

Pos	RMS Field Name	RMS Description	RMS Field Type	RMSMax Width	Xstore Table	Xstore Column	Xstore Data Type	Xstore Default Value	Xstore Comment
50	Diff2AggregateInd	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). 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.
51	Diff3AggregateInd	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). 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

Pos	RMS Field Name	RMS Description	RMS Field Type	RMSMax Width	Xstore Table	Xstore Column	Xstore Data Type	Xstore Default Value	Xstore Comment
52	Diff4AggregateInd	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). 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

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

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

Pos	RMS Field Name	RMS Description	RMS Field Type	RMSMax Width	Xstore Table	Xstore Column	Xstore Data Type	Xstore Default Value	Xstore Comment
54	FormatID	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: 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 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). Data is optional in this field, and can only exist for the ItemNumberType = VPLU.	Character	1	NA	NA	NA	NA	Ignored.

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

Pos	RMS Field Name	RMS Description	RMS Field Type	RMSMax 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	RecHandlingTemp	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	RecHandlingSens	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	RMSMax 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

Pos	RMS Field Name	RMS Description	RMS Field Type	RMSMax Width	Xstore Table	Xstore Column	Xstore Data Type	Xstore Default Value	Xstore Comment
61	DefaultWastePct	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	ConstantDimInd	This field will always have data.	Character	1	NA	NA	NA	NA	Ignored.
63	ContainsInnerInd	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	SellableInd	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

Pos	RMS Field Name	RMS Description	RMS Field Type	RMSMax Width	Xstore Table	Xstore Column	Xstore Data Type	Xstore Default Value	Xstore Comment
65	OrderableInd	Indicates if the item may be ordered as a unit. Valid values are: Y - Yes, this item/pack may be ordered from a single supplier. N - No, this item/pack may not be ordered from a single supplier. 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: Ground beef item is not orderable because the retailer orders a side of beef and transforms it into ground beef. In-house bakery items are not orderable because the retailer produces them in house. 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

Pos	RMS Field Name	RMS Description	RMS Field Type	RMSMax Width	Xstore Table	Xstore Column	Xstore Data Type	Xstore Default Value	Xstore Comment
66	PackType	Indicates if a pack item is a vendor pack or a buyer pack. 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. 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. 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. Data is optional in this field.	Character	1	NA	NA	NA	NA	Ignored.

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

Pos	RMS Field Name	RMS Description	RMS Field Type	RMSMax 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

Pos	RMS Field Name	RMS Description	RMS Field Type	RMSMax Width	Xstore Table	Xstore Column	Xstore Data Type	Xstore Default Value	Xstore Comment
70	ShipAloneInd	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	ItemXformInd	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	CatchWeightInd	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

Pos	RMS Field Name	RMS Description	RMS Field Type	RMSMax Width	Xstore Table	Xstore Column	Xstore Data Type	Xstore Default Value	Xstore Comment
73	CatchWeightType	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	CatchWeightOrderType	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	CatchWeightSaleType	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	CatchWeightUOM	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

Pos	RMS Field Name	RMS Description	RMS Field Type	RMSMax Width	Xstore Table	Xstore Column	Xstore Data Type	Xstore Default Value	Xstore Comment
77	DepositItemType	This is the deposit item component type. A NULL value in this field indicates that this item is not part of a deposit item relationship. Valid values include: E - Contents A - Container Z - Crate T - Returned Item (Empty bottle) P - Complex pack (with deposit items) The Returned Item is flagged only to enable these items to be mapped to a separate GL account if required. Data is optional in this field.	Character	6	NA	NA	NA	NA	Ignored.
78	ContainerItem	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. Data is optional in this field.	Character	25	NA	NA	NA	NA	Ignored.

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

Pos	RMS Field Name	RMS Description	RMS Field Type	RMSMax Width	Xstore Table	Xstore Column	Xstore Data Type	Xstore Default Value	Xstore Comment
79	DepositInPricePerOUM	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	SOHInquiryAtPackInd	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	NotionalPackInd	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

Pos	RMS Field Name	RMS Description	RMS Field Type	RMSMax 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	VARCHAR2(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_TYP CODE	VARCHAR2(30 CHAR)	STANDARD	NA
NA	NA	NA	NA	NA	itm_item	DESCRIPTION	VARCHAR2(254 CHAR)	NA	Contains synthesized Translation key following the pattern: "+item:[ITEM ID]:description". Key maps to the com_tranlsation record populated from RMS ItemLoc.
NA	NA	NA	NA	NA	itm_item	NAME	VARCHAR2(254 CHAR)	NA	Contains synthesized Translation key following pattern: "+item:[ITEM ID]:description". Key maps to com_tranlsation record populated from RMS ItemLoc.

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

Pos	RMS Field Name	RMS Description	RMS Field Type	RMSMax 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.

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
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	SellingUnitRetail	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	SellingUOM	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 ITEMLOCMOD.	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	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). The intent is that this string is appropriate to print this description on signage/receipts at this location. This field will always have data.	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.

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
13	LocalShorDesc	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. The intent is that this string is appropriate to print this description on signage/receipts at this location. Data is optional in this field.	Character	120	NA	NA	NA	NA	Ignored.
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.

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
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	StoreOrderMultiple	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.
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.

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
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.
21	UomOfPrice	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.

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
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	Inbound Handlin gDays	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.

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
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.
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.

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
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_item.serialized_item_flag is set to 0. Otherwise, itm_item.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	CaptureTimeInProc	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.
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.

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
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	Costing Location	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	Costing LocType	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	Launch Date	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, othewise 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	RewardEligibleInd	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	NatlBrandCompItem	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	ElectMtkClub	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	ReportCode	Code to determine which reports the location should run. Data is optional in this field.	Character	6	NA	NA	NA	NA	Ignored.
52	ReqShelfLifeOnSelection	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	ReqShelfLifeOnReceipt	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	IBShelfLife	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.

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
64	Merch5andiseInd	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.
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 =ITEMLOCMOD.
	NA	NA	NA	NA	itm_item, itm_item_options	CREATE_USER_ID	VARCHAR2(30 CHAR)	DATALOADER	Not set when Type =ITEMLOCMOD.

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
	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_HEIRARCHY	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	Value is always FULL when the program is run in full mode. 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: DIVISIONCRE DIVISIONMOD DIVISIONDEL GROUPCRE GROUPMOD GROUPDEL DEPTCRE DEPTMOD DEPTDEL CLASSCRE CLASSMOD CLASSDEL SUBCLASSCRE SUBCLASSMOD SUBCLASSDEL **Note:** Deleting a node of the merchandise hierarchy is unusual, and can only be done if no items are associated with the node.	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	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). 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.	Number	4	NA	NA	NA	NA	NA
8	parentMerchDisplayId	Only populated for CLASS and SUBCLASS entities. For classes, this column holds the department ID used for display purposes in RMS (department is the parent of class). For subclasses, this column holds the class ID used for display purposes in RMS. 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.	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	Only populated for DEPARTMENT, CLASS, and SUBCLASS entities. For departments, this column holds the department display ID. For classes, this column holds the class display ID. For subclasses, this column holds the subclass display ID. 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.	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

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 ORGHIERARCHY.	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, and be composed of the entity (COMP, CHAIN, AREA, REGION, DISTRICT, STORE, WH) and the type of change (CRE, MOD, DEL). Examples include: COMPCRE - Creation of a Company node. COMPMOD - Modification of a Company node. CHAINCRE - Creation of Chain node. CHAINMOD - Modification of a Chain node. CHAINDEL - Delete of Chain node. AREACRE - Creation of an Area node. AREAMOD - Modification of Area node. AREADEL - Delete of Area node. REGIONCRE - Creation of Region node. REGIONMOD - Modification of Region node. REGIONDEL - Delete of Region node. DISTRICTCRE - Creation of District node.	Character	15	NA	NA	NA	NA	Determines action.

Table B-7 (Cont.) RMS Organizational 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
	Type (Continued)	DISTRICTMOD - Modification of District node. DISTRICTDEL - Delete of a District node. STORECRE - Creation of a Store node. STOREMOD - Modification of Store node. STOREDEL - Delete of Store node. WHCRE - Creation of WH node. WHMOD - Modification of WH node. WHDEL - Delete of WH node. **Note:** Deletion of Company node is not allowed.	NA	NA	NA	NA	NA	NA	NA
2	HierarchyLevel	Value is always COMPANY, CHAIN, AREA, REGION, DISTRICT, STORE, or WAREHOUSE. This field cannot be null. This information identifies the level of the organizational hierarchy that is described by this record. It is not possible to modify a HierarchyLevel on an existing record.	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

Pos	RMS Field Name	RMS Description	RMS Field Type	RMS Max Width	Xstore Table	Xstore Column	Xstore Data Type	Xstore Default Value	Xstore Comment
3	HierarchyNodeId	This information identifies the node of the organizational hierarchy that is described by this record. This field cannot be null. 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). It is not possible to modify HierarchyNodeId on an existing record.	Number	10	loc_org_hierarchy	org_code	VARCHAR2(60 CHAR)	NA	For the root COMPANY node, the column is hard-coded as "*".
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	loc_org_hierarchy	description	VARCHAR2(254 CHAR)	NA	NA
5	ParentLevel	Level of the organizational hierarchy above the current node. Both ParentLevel and ParentId should be evaluated to correctly traverse the hierarchy. ParentLevel is null for the COMPANY, but will exist for all other hierarchy levels. Warehouses always have the COMPANY node as their ParentLevel. It is not possible to modify ParentLevel on an existing record.	Character	10	loc_org_hierarchy	parent_code	VARCHAR2(30 CHAR)	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 "*".

Table B-7 (Cont.) RMS Organizational 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
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 is null for the COMPANY, but will exist for all other hierarchy levels. Warehouses always have the COMPANY node ID as their ParentId. 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).	Number	10	loc_org_hierarchy	parent_value	VARCHAR2(60 CHAR)	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 "*".
7	MgrName	Manager of the current node of the hierarchy. 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. MrgName can be modified.	Number	10	loc_org_hierarchy	level_mgr	VARCHAR2(254 CHAR)	NA	NA

Table B-7 (Cont.) RMS Organizational 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
8	Currency Code	Currency of the current node of the hierarchy. 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. CurrencyCode can be modified for a CHAIN, AREA, REGION, and DISTRICT. It cannot be modified for a STORE or Warehouse.	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 - AREA30 - 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.
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	ORGANIZATION_ID	NUMBER(10,0)	${dtv.location.organizationId}	Always uses default value.
NA	NA	NA	NA	NA	loc_org_hierarchy	CREATE_DATE	TIMESTAMP(6)	now()	Not set when Type represents UPDATE.
NA	NA	NA	NA	NA	loc_org_hierarchy	UPDATE_DATE	TIMESTAMP(6)	now()	Not set when Type represents CREATE.
NA	NA	NA	NA	NA	loc_rtl_loc	UPDATE_USER_ID	VARCHAR2(30 CHAR)	DATALOADER	Always set.

RMS Related Item Detail

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

Table B-8 RMS Related Item Detail 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 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. Relationship Type is looked up from rms_related_item_head by RelationshipId, Location, and Organization Id.
2	RelationshipID	Unique identifier for each relationship header. Data will always be present in this field.	Number	10	itm_substitute_items, itm_attached_items	EXTERNAL_ID	VARCHAR2(60 CHAR)	NA	NA
3	RelatedItem	Item ID of the related item Data will always be present in this field.	Character	25	itm_substitute_items, itm_attached_items	ATTACHED_ITEM_ID, SUBSTITUTE_ITEM_ID	VARCHAR2(60 CHAR)	NA	NA
4	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. Data will always be present in this field.	Character	10	itm_substitute_items, itm_attached_items	LEVEL_VALUE	VARCHAR2(60 CHAR)	NA	NA

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

Table B-8 (Cont.) RMS Related Item Detail 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_substitute_items, itm_attached_items	CREATE_DATE	TIMESTAMP(6)	now()	Not set when Type =RELITEMDETMOD.
NA	NA	NA	NA	NA	itm_substitute_items, itm_attached_items	CREATE_USER_ID	VARCHAR2(30 CHAR)	DATALOADER	Not set when Type =RELITEMDETMOD.
NA	NA	NA	NA	NA	itm_substitute_items, itm_attached_items	UPDATE_DATE	TIMESTAMP(6)	now()	Not set when Type =FULLRELITEMDET.
NA	NA	NA	NA	NA	itm_substitute_items, itm_attached_items	UPDATE_USER_ID	VARCHAR2(30 CHAR)	DATALOADER	Not set when Type =FULLRELITEMDET.
NA	NA	NA	NA	NA	itm_attached_items	PROMPT_TO_ADD_FLAG	NUMBER(1,0)	TRUE	Configurable in dataloader-beans.xml.
NA	NA	NA	NA	NA	itm_attached_items	PROMPT_TO_ADD_MSG_KEY	VARCHAR2(254 CHAR)	_commonAttachedItemsPrompt	Configurable 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

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 ITEMS. Data will always be present in this field.	Character	20	NA	NA	NA	NA	Ignored.

Table B-9 (Cont.) RMS Related 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
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: RELITEMHEADCRE RELITEMHEADMOD RELITEMHEADDEL 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 item_attached_items when the modified record represents a relationshiptype change between CRSL/UPSL and SUBS.
2	RelationshipID	Unique identifier for each relationship header. Data will always be present in this field.	Number	20	rms_related_item_head	RELATIONSHIP_ID	NUMBER(20,0)	NA	Used as externalId in itm_substute_items and item_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	VARCHAR2(25 CHAR)	NA	When SUBS, this is the primaryItem. When CRSL/UPSL, this is the soldItem.
4	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. Data will always be present in this field.	Character	10	rms_related_item_head	LOCATION	VARCHAR2(10 CHAR)	NA	Used as levelValue in itm_substute_items and item_attached_items records.

Table B-9 (Cont.) RMS Related 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
5	RelationshipName	Name given to the relationship. Data will always be present in this field for the RELITEMHEADCRE and RELITEMHEADMOD records, but will not be present for RELITEMHEADDEL records.	Character	255	rms_related_item_head	RELATIONSHIP_NAME	VARCHAR2(255 CHAR)	NA	Ignored.
6	RelationshipType	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 RELITEMHEADCRE and RELITEMHEADMOD records, but will not be present for RELITEMHEADDEL records.	Character	6	rms_related_item_head	RELATIONSHIP_TYPE	VARCHAR2(6 CHAR)	NA	Determines the .type of Xstore related item. SUBS = Xstore Substitute Items, CRSL/UPSL = Xstore Attached Items.

Table B-9 (Cont.) RMS Related 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
7	MandatoryInd	This field indicates whether the relationship should be mandatory. 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. 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. Data will always be present in this field for the RELITEMHEADCRE and RELITEMHEADMOD records, but will not be present for RELITEMHEADDEL records.	Character	1	rms_related_item_head	MANDATORY_IND	VARCHAR2(1 CHAR)	NA	No logic is based on this data.

Table B-9 (Cont.) RMS Related 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	rms_related_item_head	ORGANIZATION_ID	NUMBER(10,0)	${dtv.location.organizationId}	Always uses default value.
NA	NA	NA	NA	NA	rms_related_item_head	CREATE_DATE	TIMESTAMP(6)	now()	Not set when Type =ITEMHDRMOD.
NA	NA	NA	NA	NA	rms_related_item_head	UPDATE_DATE	TIMESTAMP(6)	now()	Not set when Type =FULLRELITEMHDR
NA	NA	NA	NA	NA	item_item_options	ATTACHED_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	SUBSTITUTE_AVAILABLE_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

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.

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
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	StoreName	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	DESCRIPTION	VARCHAR(254)	NA	NA
4	StoreName10	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	VARCHAR(254)	NA	NA
5	StoreNameAbbr	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	StoreNameSec	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

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	StoreClassId	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	StoreClassDesc	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	Manager	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_MANAGER	VARCHAR(254)	NA	NA
10	OpenDate	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	CloseDate	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	AquireDate	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	Remodel Date	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

Pos	RMS Field Name	RMS Description	RMS Field Type	RMS Max Width	Xstore Table	Xstore Column	Xstore Data Type	Xstore Default Value	Xstore Comment
14	FaxNumber	Contains the fax number for the store. Data in this field is optional.	Character	20	loc_rtl_loc	TELEPHONE_2	VARCHAR(32)	NA	Optional fax number.
15	PhoneNumber	Contains the phone number for the store. Data in this field is optional.	Character	20	loc_rtl_loc	TELEPHONE_1	VARCHAR(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	VARCHAR(254)	NA	Optional email address.
17	TotalSqFeet	Contains the total square footage of the store. Data in this field is optional.	Number	8	NA	NA	NA	NA	Ignored.
18	SellingSqFeet	Contains the total square footage of the stores selling area. Data in this field is optional.	Number	8	NA	NA	NA	NA	Ignored.
19	LinearDistance	Holds the total merchandisable space of the location. Data in this field is optional.	Number	8	NA	NA	NA	NA	Ignored.
20	VatRegion	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	taxLocationId	VARCHAR2(60)	NA	NA
21	VatInclInd	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	StockHoldingIn	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.
23	ChannelId	Contains the channel with which the store is associated. Data in this field is optional.	Number	4	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
24	Channel Name	Contains the name of the channel. Data in this field is optional.	Character	120	NA	NA	NA	NA	Ignored.
25	StoreFormat	Contains the number indicating the format of the store. Data in this field is optional.	Number	4	NA	NA	NA	NA	Ignored.
26	StoreFormatName	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	MallName	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	DefaultWh	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.
31	StopOrderDays	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.

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
32	StartOrderDays	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	Currency Code	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	CURRENCY_ID	VARCHAR(3)	NA	ISO Currency Code.
34	StoreLangISOCode	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.
35	TranNoGenerate	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.

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
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.
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.

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
42	AutoRcv	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.
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.

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
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.
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
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.

Table B-11 (Cont.) 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
5	PrimaryAddrInd	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
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.

Table B-11 (Cont.) 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
15	Contact Name	Contains the name of the contact person at this address. Optional.	Character	120	NA	NA	NA	NA	Ignored.
16	ContactPhone	Contains a phone number of the contact person at this address. Optional.	Character	20	NA	NA	NA	NA	Ignored.
17	ContactFax	Contains a fax number of the contact person at this address. Optional.	Character	20	NA	NA	NA	NA	Ignored.
18	ContactEmail	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

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 VAT. Data will always be present in this field.	Character	20	NA	NA	NA	NA	Ignored.

Table B-12 (Cont.) RMS VAT 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	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.
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

Table B-12 (Cont.) RMS VAT 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	VatCode Desc	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_DATETIME	TIMESTAMP(6)	NA	NA
NA	NA	NA	NA	NA	tax_tax_group_rule	TAX_TYP CODE	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.
NA	NA	NA	NA	NA	tax_tax_group_rule	TAXED_AT_TRANS_LEVEL_FLAG	NUMBER(1,0)	TRUE	A configuration spring loaded by the transformer. The out-of-box default is TRUE.

Table B-12 (Cont.) RMS VAT 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	tax_tax_rate_rule	EXPR_DATETIME	TIMESTAMP(6)	NA	This has to be derived by the integration layer. 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. 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.

Table B-12 (Cont.) RMS VAT 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	NA	EXPR_DATETIME (Continued)	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/VAT code combination.
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.
NA	NA	NA	NA	NA	tax_tax_authority	ROUNDING_CODE	VARCHAR(30)	HALF_UP	A configuration spring loaded by the transformer. The out-of-box default is HALF_UP.

Table B-12 (Cont.) RMS VAT 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	tax_tax_authority	ROUNDING_DIGITS_QUANTITY	NUMBER(10,0)	2	A configuration 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	EXTERNAL_SYSTEM	VARCHAR(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	VARCHAR(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	VARCHAR(60)	*	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	ORGANIZATION_ID	NUMBER(10,0)	${dtv.location.organizationId}	Always uses the default value.

Table B-12 (Cont.) RMS VAT 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	tax_tax_loc, tax_tax_authority, tax_tax_group, tax_tax_group_rule, tax_tax_rate_rule	CREATE_DATE	TIMESTAMP(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	TIMESTAMP(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

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 VATITEM. Data will always be present in this field.	Character	20	NA	NA	NA	NA	Ignored.

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

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	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 the 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.

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

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 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

Table B-14 (Cont.) RPM Clearance 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
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).
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.
NA	NA	NA	NA	itm_item_prices	ORGANIZATION_ID	NUMBER(10,0)	${dtv.location.organizationId}	Always uses default value.
NA	NA	NA	NA	itm_item_prices	PROPERTY_CODE	VARCHAR2(60 CHAR)	NA	CLEARANCE_PRICE when Clearance event, or REGULAR_PRICE when Clearance Reset event.
NA	NA	NA	NA	itm_item_prices	CREATE_DATE	TIMESTAMP(6)	now()	Not set when EventType = Mod.
NA	NA	NA	NA	itm_item_prices	CREATE_USER_ID	VARCHAR2(30 CHAR)	DATALOADER	Not set when EventType = Mod.
NA	NA	NA	NA	itm_item_prices	UPDATE_DATE	TIMESTAMP(6)	now()	Always set.

Table B-14 (Cont.) RPM Clearance 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	UPDATE_USER_ID	VARCHAR2(30 CHAR)	DATALOADER	Always set.
NA	NA	NA	NA	itm_item_prices	EXTERNAL_SYSTEM	VARCHAR2(60 CHAR)	RPM-CLRPC	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

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 (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	EXTERNAL_ID	VARCHAR2(60 CHAR)	NA	NA
3	Item	Item identifier	Char(25)	itm_item_prices	ITEM_ID	VARCHAR2(60 CHAR)	NA	NA
	NA	NA	NA	itm_item_prices	ORGANIZATION_ID	NUMBER(10,0)	${dtv.location.organizationId}	Always uses default value.
	NA	NA	NA	itm_item_prices	EXTERNAL_SYSTEM	VARCHAR2(60 CHAR)	RPM-CLRPC	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

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 Marker (1 per price change create or modify)	Char(5)	NA	NA	NA	NA	Type of event.

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
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 transformer to determine the type of event.
3	Id	Price change 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	Effective Date of the price change (YYYYMMDDHH24MISS)	NA	itm_item_prices	EFFECTIVE_DATE	TIMESTAMP(6)	NA	NA
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

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	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.
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 (1per 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
NA	NA	NA	NA	itm_item_prices	ORGANIZATION_ID	NUMBER(10,0)	${dtv.location.organizationId}	NA
NA	NA	NA	NA	itm_item_prices	EXTERNAL_SYSTEM	VARCHAR2(60 CHAR)	RPM-CLRPC	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 (YYYYMMDDHHMISS)	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)	TIMPBE	Promotion (transaction head). Defines an action associated with a promotion.	NA	NA	NA
	Line ID	Number(10)		Unique line 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
	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.
	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 (YYYYMMDDHH24MISS)	prc_deal	effective_date	NA
	End Date	Date		End Date of Promotion Component Detail (YYYYMMDDHH24MISS)	prc_deal	end_date	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 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
	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.

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
	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).
	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.

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 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.
	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.

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
	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 - YYYYMMDDHH24MISS	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.
	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.

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 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 - YYYYMMDDHH24MISS	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
	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