Oracle® Retail Xstore Suite 18.0/Merchandising 16.0.2

Implementation Guide

Release 18.0

E99047-05

September 2020

Oracle Retail Xstore Suite 18.0/Merchandising 16.0.2 Implementation Guide, Release 18.0

E99047-05

Primary Author: Bernadette Goodman

Contributing Author: Siobhan McMahon, Owen Horne, Sandy Gu, Sergey Tinyaev

Value-Added Reseller (VAR) Language

Oracle Retail VAR Applications

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

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

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

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

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

You acknowledge and confirm that Oracle grants you use of only the object code of the VAR Applications. Oracle will not deliver source code to the VAR Applications to you. Notwithstanding any other term or condition of the agreement and this ordering document, you shall not cause or permit alteration of any VAR

Applications. For purposes of this section, "alteration" refers to all alterations, translations, upgrades, enhancements, customizations or modifications of all or any portion of the VAR Applications including all reconfigurations, reassembly or reverse assembly, re-engineering or reverse engineering and recompilations or reverse compilations of the VAR Applications or any derivatives of the VAR Applications. You acknowledge that it shall be a breach of the agreement to utilize the relationship, and/or confidential information of the VAR Applications for purposes of competitive discovery.

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

Contents

4 Configuration

5 Integration Considerations

List of Figures

List of Tables

Send Us Your Comments

Oracle Retail Xstore Suite 18.0/Merchandising 16.0.2 Implementation Guide

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

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

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

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

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

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

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

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

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

Preface

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

Audience

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

Documentation Accessibility

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

Access to Oracle Support

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

Related Documents

For more information, see the following Release documents:

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

Customer Support

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

https://support.oracle.com

When contacting Customer Support, please provide the following:

- Product version and program/module name
- Functional and technical description of the problem (include business impact)
- Detailed step-by-step instructions to re-create
- Exact error message received
- Screen shots of each step you take

Review Patch Documentation

When you install the application for the first time, you install either a base release (for example, 18.0) or a later patch release (for example, 18.0.1). 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 Technology Network 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 Technology Network at the following URL:

http://www.oracle.com/technetwork/documentation/oracle-retail-100266.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 Technology Network

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

`http://www.oracle.com/technetwork/documentation/oracle-retail-100266.html`

(Data Model documents are not available through Oracle Technology Network. You can obtain them 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.
`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

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

- Foundation and price data from Oracle Retail Merchandising System (RMS) and Oracle Retail Price Management (RPM) to Oracle Retail Xcenter and Xstore Office
- Point of Service transactions from Oracle Retail Xstore Point of Service to Oracle Retail Sales Audit (ReSA)

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.

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

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

2

Data Flow from Merchandising to Xstore

This chapter covers the data flow from RMS and RPM to Xcenter/Xstore.

RMS is the source of foundation data. RMS foundation data sent to Xcenter/Xstore is limited to the following:

- Merchandise hierarchy
- Organizational hierarchy
- Store (including addresses)
- Diff IDs
- Diff groups
- Items
- VAT tax rules and item associations
- Related items

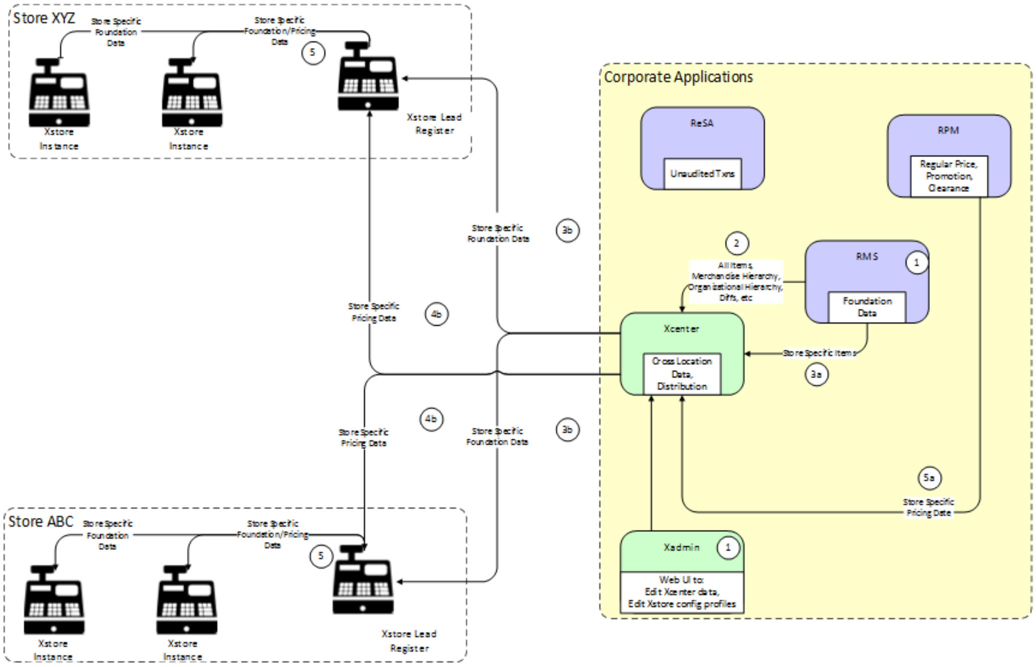
RPM is the source of pricing data. RPM pricing data sent to Xcenter/Xstore is limited to the following:

- Price changes
- Promotions
- Clearance prices

Conceptual Data Flow

Figure 2–1 illustrates the data flow from the Merchandising applications to Xcenter/Xstore.

Figure 2–1 Conceptual Data Flow from Merchandising to Xstore Suite


Store XYZ
Store Specific Foundation Data
Store Specific Foundation/Pricing Data
5
Xstore Instance
Xstore Instance
Xstore Lead Register
Corporate Applications
ReSA
Unaudited Txns
RPM
Regular Price, Promotion, Clearance
Store Specific Foundation Data
3b
2
All Items, Merchandise Hierarchy, Organizational Hierarchy, Diffs, etc
RMS
1
Foundation Data
Store Specific Pricing Data
4b
Xcenter
Cross Location Data, Distribution
Store Specific Items
3a
Store Specific Pricing Data
4b
Store Specific Foundation Data
3b
5a
Store Specific Pricing Data
Store ABC
Store Specific Foundation Data
Store Specific Foundation/Pricing Data
5
Xadmin
1
Web UI to: Edit Xcenter data, Edit Xstore config profiles
Xstore Instance
Xstore Instance
Xstore Lead Register

The following steps describe the flow in Figure 2–1:

1. Manual process to set up some infrequently changing foundation data in both Xstore Office and RMS, for example, currency codes.
2. RMS produces foundation data consumed by Xcenter. Xcenter loads this foundation data to, among other things, facilitate cross location transactions. Xcenter loads foundation data to the appropriate lead registers at individual stores.
3. RMS produces store-specific foundation data consumed by Xcenter (3a). Xcenter does not load this data into its own repository, but instead distributes these files to the appropriate store lead registers (3b).
4. RPM produces store-specific pricing data consumed by Xcenter (4a). Xcenter does not load this data into its own repository, but instead distributes these files to the appropriate store lead registers (4b).
5. Lead register distributes store specific foundation data to all other Xstore instances in the store. This step occurs when the store is closed, or in a 24x7 configuration, when the retail period ends.

Technical Implementation

The technical implementation of the foundation/price data from Merchandising to Xcenter/Xstore consists of three main components:

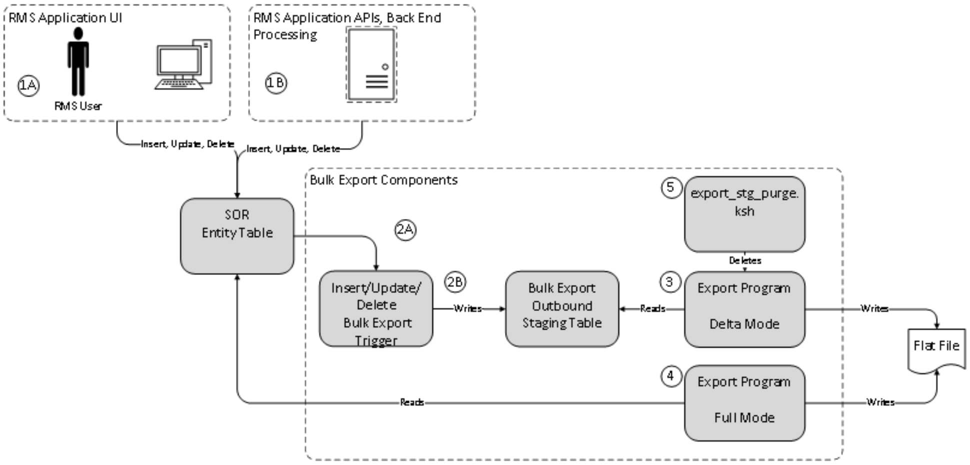
- RMS Foundation Data Extracts
- RPM Extracts
- Xstore DataLoader

RMS Foundation Data Extracts

RMS 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 (for Xcenter, to facilitate cross store processing) or store-specific files. For an example, see Figure 2–5.

There are some entity specific variations, but RMS uses a general pattern for foundation data bulk export as shown in Figure 2–2.

Figure 2–2 RMS Foundation Data Bulk Export



The following steps describe the data bulk export flow shown in Figure 2–2:

1. A business user using the RMS application UI (1A) or an API/Batch Process (1B) performs an insert/update/delete on a System of Record (SOR) table.
2. Trigger on SOR entity table fires on insert/update/delete (2A). Trigger writes new/changed/deleted information to the outbound staging table (2B).
3. In delta mode, the program reads the bulk export staging table to get recently created, modified, and deleted records and writes them to a file. Records are marked as exported.
4. In full mode, the program reads all current records from the SOR table and writes them to a file. Note that recently deleted records are not part of the data set.
5. export_stg_purge.ksh drops aged partitions from the export outbound staging tables.

> **Note:** If bulk extract programs are not run for some time, it is possible that delta records will be purged without having been exported. It is important to run these jobs daily.

For more detailed information, see the following documents:

- Retail Reference Architecture available on My Oracle Support
- *Oracle Retail Merchandising System Operations Guide, Volume 1 - Batch Overviews and Designs*

RMS Batch Jobs

The following batch jobs are used for the integration:

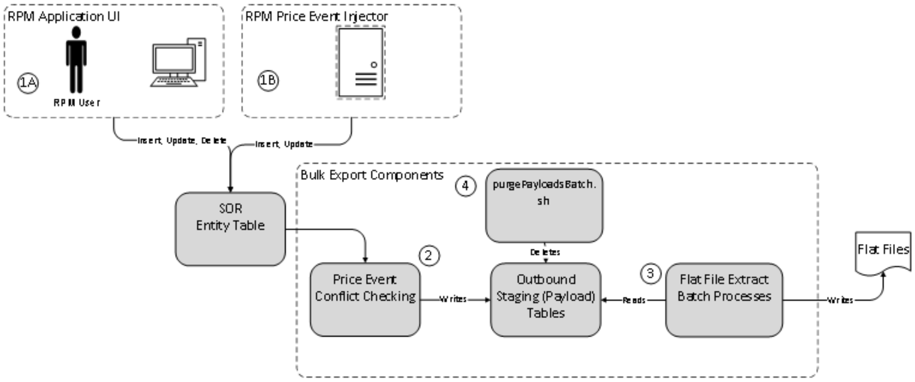
- export_merchhier.ksh
- export_orghier.ksh
- export_stores.ksh
- export_diffs.ksh
- export_diffgrp.ksh
- export_itemmaster.ksh
- export_itemloc.ksh
- export_itemvat.ksh
- export_relitem.ksh
- export_vat.ksh
- export_stg_purge.ksh

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

RPM Extracts

RPM writes data to .dat files using a set of base processes. These base processes send pricing data to both Xcenter/Xstore and Oracle Retail Store Inventory Management (SIM). This ensures consistent price information across the Xstore applications. For an example, see Figure 2–5.

Figure 2–3 RPM Extract Flow



The following steps describe the flow in Figure 2–3:

1. A Pricing Analyst creates and approves price events (regular price changes, clearances, and promotion details) in the RPM UI (1A). Price events are created and approved in a bulk fashion using the Price Event Injector batch process (1B).
2. With data created on the SOR tables in RPM, the conflict checking process stages data on the outbound (payload) tables (2).
3. Data on the outbound staging tables is read by the flat file extraction batch processes to create delimited flat files. The associated outbound staging data is flagged as having been extracted as part of these batch processes (3).

4. purgePayloadsBatch.sh purges data from the staged outbound tables that has already been extracted. This purge batch is run in conjunction with the extract batch processes (4).

For more detailed information, see the following documents:

- Retail Reference Architecture available on My Oracle Support
- *Oracle Retail Price Management Operations Guide*

RPM Batch Jobs

The following batch jobs are used for the integration:

- RegularPriceChangePublishBatch
- PromotionPriceChangePublishBatch
- ClearancePricePublishBatch

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

Data Import Flow

The following process describes the flow of the Merchandising data file import:

1. Xstore Office plays a central role in the Merchandising data importing. It periodically polls the configured auto-deploy directory on the file system. The 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*.
2. Data files (.dat) generated by the RMS/RPM extract programs are delivered to Xstore Office's auto-deploy directory in the form of a zip-format archive file using the file extension .momzip. It is the System Integrator's responsibility to create the archive, and deliver it using a preferred file transfer protocol.
3. Once the archive is detected by Xstore Office, it regroups its content into deployments based on their targeted locations. For data files targeted to corporate, it invokes DataLoader immediately to import them into the Xcenter data source. For data files targeted to a store, it creates a deployment of these files to the store, and uploads them to an Apache Server for the store to download when updates are applied. For a traditional store, this happens at store close, and for a 24x7 store, this happens when the retail period changes. For details on the set of Merchandising files targeted to corporate or stores, see "Merchandising File Consumption by Location".
4. Once a store is closed or when the retail period changes in a 24x7 configuration, Xenvironment of the lead register pulls down the files from the Apache server and runs DataLoader to import all the files deployed into the store primary database.

Xstore DataLoader

DataLoader is the Xstore component responsible for translating RPM and RMS flat files into database data that can be used by Xstore. It can consume .mnt files from third-party sources, or the foundation data batch .dat files produced by RMS and RPM.

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 RPM and RMS 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. For most file types, a unit of work corresponds to a single line of flat file data; however, RPM promotion files are an example of a file type where a unit of work can consist of multiple lines of flat file data.

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* available on My Oracle Support.

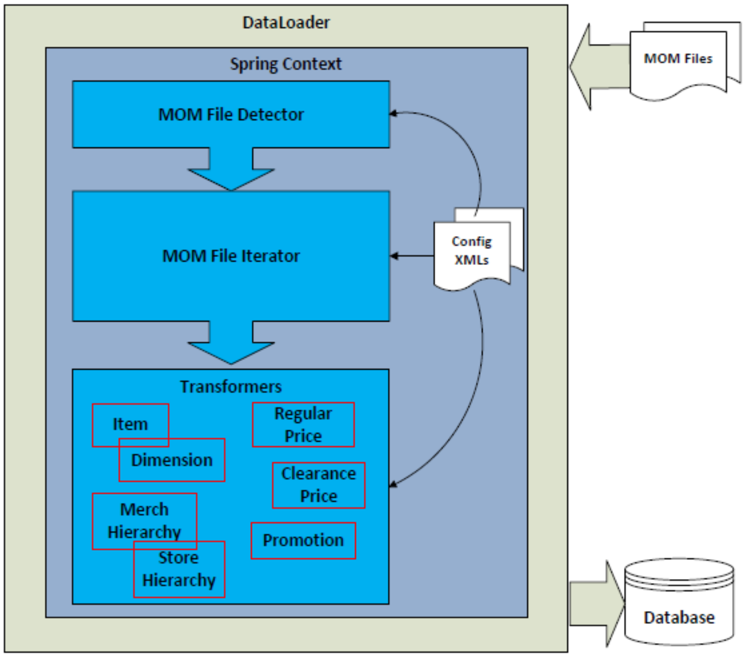
Both documents are available on My Oracle Support.

DataLoader Detailed Description

DataLoader is configured at the corporate and store level. It is responsible for detecting and sorting incoming data files, and iterating through them to convert each file record into persistable objects (DAO or SQLQuery) and writing them into the database. All its major components are spring loaded from dataloader-bean.xml, and are therefore highly customizable.

Figure 2–4 shows the flow of the DataLoader.

Figure 2–4 DataLoader Flow



The following sections describe each major execution phase.

File 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 (excluding FHEAD and FTAIL) does not match the line count, a warning is logged. A file name is matched against regular expressions configured to detect its line count. Only RMS extracts support line count in their file names.

File Sorting

There are some data dependencies when importing RMS 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 the following table:

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

File Iteration and Transformation

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

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

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

- Unit dividers are lines that end a unit. They are configured as unit definitions for each Merchandising file type.

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

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

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

Persisting into the Database

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

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

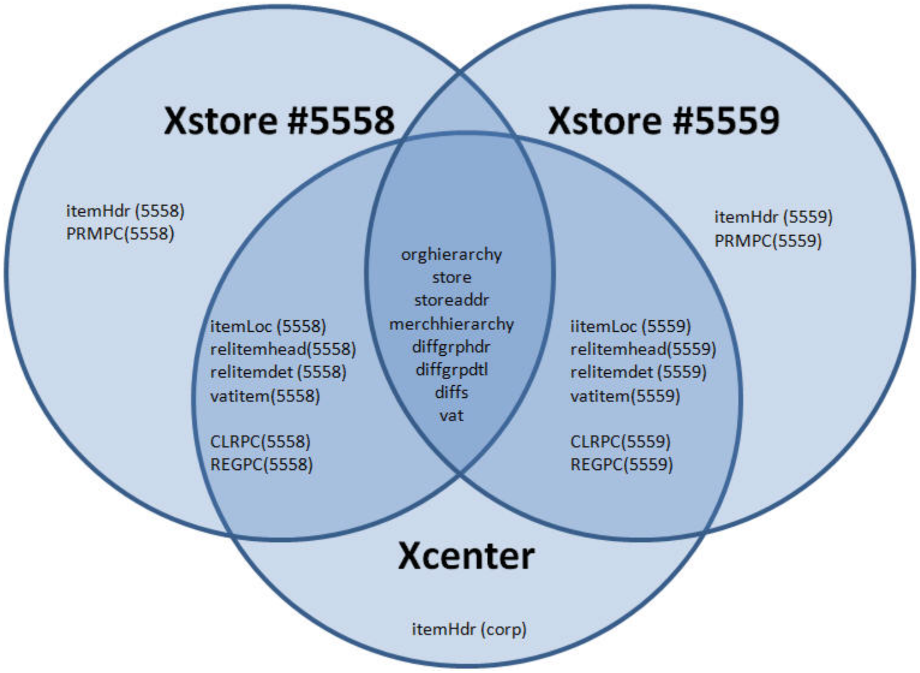
Merchandising File Consumption by Location

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

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

Figure 2–5 illustrates the type of RMS and RPM files loaded at each location, using a two store chain example.

Figure 2–5 Example of Loaded RMS and RPM Files



3

Transaction Flow from Xstore to ReSA

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

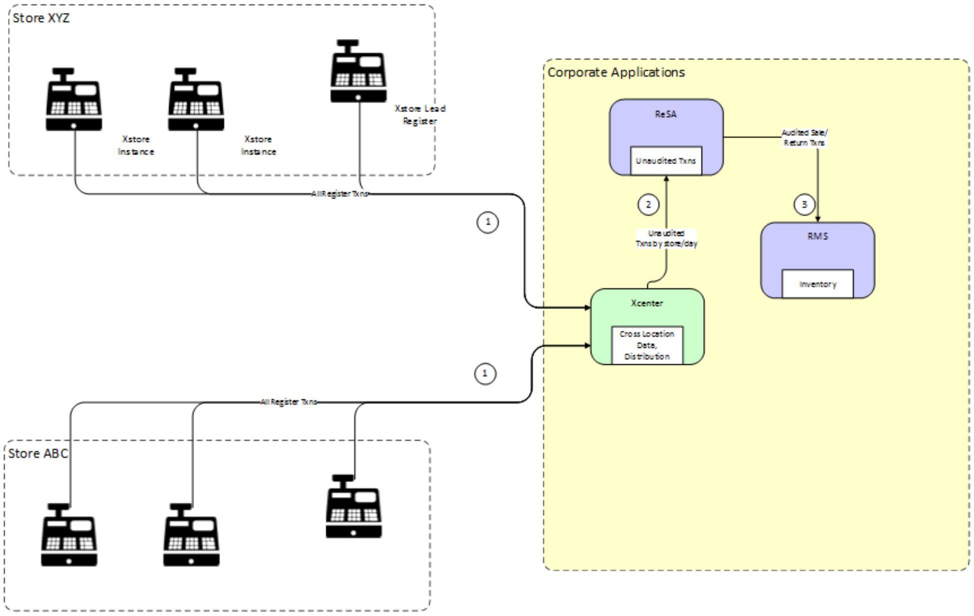
All transactions produced in Xstore are sent to ReSA. ReSA processing is primarily concerned with transactions that alter inventory or contain payment. ReSA loads other types of Xstore transactions (such as entering training mode, gift registry creation, and so on) into an OTHER transaction type for 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.

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

Conceptual Data Flow

Figure 3–1 illustrates the transaction flow from Xstore to ReSA.

Figure 3–1 Xstore to ReSA 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 ReSA in the form of RTLogs generated multiple times per day. For more information, see "ReSA saimptlog/i".
3. After successful totaling and auditing, ReSA sends all sale/return transactions to RMS, where the transactions impact perpetual inventory. For detailed information about uploadsales_all.ksh, see *Oracle Retail Merchandising System Operations Guide, Volume 1 - Batch Overviews and Designs*.

Technical Implementation

The technical implementation of the foundation/price data from Merchandising to Xcenter/Xstore consists of three main components:

- Xstore Broadcaster
- RTLog Generator
- ReSA saimptlog/i

Xstore Broadcaster

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.

There are a few systems which the base version of Xcenter can readily broadcast data to, simply by making configuration changes.

For more detailed information, see the following documents:

- Retail Reference Architecture available on My Oracle Support
- *Oracle Retail Xstore Technical Guide* available on My Oracle Support
- *Oracle Retail Xstore Suite Implementation Guide*

RTLog Generator

RTLog generator is a component that collects and aggregates broadcaster transactions and transforms them to the RTLog file format. The RTLog generator is packaged with Xstore, but is generally deployed in the same file system as ReSA.

For more information, see Chapter 6.

ReSA saimptlog/i

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

- saimptlogi.c validates files and directly inserts the transactions into the ReSA tables. This includes (as necessary) creating errors for the auditors to research and correct.
- saimptlog.c 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 ReSA 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 ReSA batch schedule which must be completed before POS transactions can be loaded. For more information about supporting batch jobs, see *Oracle Retail Merchandising System Operations Guide, Volume 1 - Batch Overviews and Designs*.

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

- Retail Reference Architecture available on My Oracle Support
- *Oracle Retail Merchandising System Operations Guide, Volume 1 - Batch Overviews and Designs*

4 Configuration

This chapter provides information on the configuration changes that can be made for the integration with RMS. There are no configuration options available for integration with RPM.

RMS

For information on configuration for the integration with RMS for the following type of data, see the following:

- Diffs
- Item
- Merchandise Hierarchy
- Organizational Hierarchy
- Related Item
- Store
- VAT

Configuration is not available for all the data. If there is no available configuration, that is called out in the following sections.

The Xcenter and Xstore config.jar files contain a Spring bean configuration file used by the DataLoader. Configuration modifications are made in the following file:

```
dataloader/spring/dataloader_beans.xml
```

Diffs

Diffs (also known as Diff IDs) information provided by RMS is consumed by Xstore using the DataLoader. Diff is short for Differentiator and is similar in concept to Xstore's Dimension concept. Diffs data is the primary source of rms_diff_ids records representing an identifier record for a Diff.

RMS Diff information is communicated to Xstore in three separate extract files: Diff Group Header, Diff Group Detail, and Diffs. The rms_diff_group_head table is a staging table supporting the loading of ItemHeader records. The Diff staging tables are accessed by the DataLoader to generate Dimension data when an ItemHeader record is a Style or Style Item. Xstore and Xcenter do not directly access the rms_diff_group_head table.

Diff Group Header data is the primary source of rms_diff_group_head records representing a header record for a Diff Group. Diff Group Detail data is the primary source of rms_diff_group_detail records representing a detail record for a Diff Group.

RMS Diffs

The DataLoader uses dtv.dataloader.mom.DiffTransformer to process RMS Diffs records. This bean has one configuration option.

The prefix used for DIFF Groups generated by the DataLoader as a Default Diff Group for each Diff Type. The prefix must be four characters or less. If not provided, DEF_ is used as the prefix value.

```
<bean id="diffTransformer" class="dtv.dataloader.mom.DiffTransformer" >
  <property name="defaultDiffGroupIdPrefix" value="****" />
</bean>
```

RMS Diff Group Header Records

The DataLoader uses dtv.dataloader.mom.DiffGroupHeadTransformer to process RMS Diff Group Header records. There are no configuration options for the diffGroupHeadTransformer bean.

```
<bean id="diffGroupHeadTransformer"
class="dtv.dataloader.mom.DiffGroupHeadTransformer" />
```

RMS Diff Group Detail Records

The DataLoader uses dtv.dataloader.mom.DiffGroupDetailTransformer to process RMS Diff Group Detail records. There are no configuration options for the diffGroupDetailTransformer bean.

```
<bean id="diffGroupDetailTransformer"
class="dtv.dataloader.mom.DiffGroupDetailTransformer" />
```

Item

Configuration is available for item header and item location.

Item Header

Item Header information provided by RMS is consumed by Xstore using the DataLoader. Item Header data is the primary source of itm_item records representing physical items. Records representing non-physical items are ignored.

The DataLoader uses dtv.dataloader.mom.ItemHeadTransformer to process RMS ItemHeader records. This bean has one configuration option.

The prefix used for DIFF Groups generated by the DataLoader as a Default Diff Group for each Diff Type. This value must match the property configuration used by the diffTransformer bean. The prefix must be four characters or less. If not provided, DEF_ is used as the prefix value.

```
<bean id="itemHeadTransformer" class="dtv.dataloader.mom.ItemHeadTransformer" >
  <property name="defaultDiffGroupIdPrefix" value="****" />
</bean>
```

Item Location

Item Location information provided by RMS is consumed by Xstore using the DataLoader. Item Location data is the primary source of itm_item_option records representing store-specific attributes of physical items. Records representing non-physical items are ignored.

The DataLoader uses dtv.dataloader.mom.ItemLocTransformer to process RMS ItemLoc records. This bean has two configuration options:

- When an RMS ItemLoc record's TaxableInd='N', the TaxGroupId of the Xstore item ItemOptions record is populated with the value configured for nonTaxableTaxGroupId.
- When translateItemDescriptionEnabled is true, a com_translations record is populated using the RMS ItemLoc records' Local Item Description.

> **Note:** For anyone configuring Xcenter, this property should be configured with "true" when loading ItemLoc data into the Xstore database; the property should be configured with "false" when loading ItemLoc data into the Xcenter database.

```
<bean id="itemLocTransformer" class="dtv.dataloader.mom.ItemLocTransformer" >
    <property name="nonTaxableTaxGroupId" value="0" />
    <property name="translateItemDescriptionEnabled" value="true"/>
</bean>
```

Merchandise Hierarchy

RMS Merchandise Hierarchy information provided by RMS is consumed by Xstore using the DataLoader. RMS supports the following merchandise hierarchy levels:

COMPANY, DIVISION, GROUP, DEPARTMENT, CLASS, SUBCLASS

Xstore has four merchandise hierarchy levels, so mapping the top two RMS merchandise hierarchy levels (COMPANY, DIVISION) is not supported. Only the bottom four RMS levels (GROUP, DEPARTMENT, CLASS, SUBCLASS) can be mapped.

Following are the supported configurations to be used in Xstore's SystemConfig.xml:

1. GROUP through SUBCLASS mapping

```
<MerchHierarchy dtype="Default">
  <NumberOfLevels dtype="Integer">4</NumberOfLevels>
  <Level1Code dtype="String">GROUP</Level1Code>
  <Level2Code dtype="String">DEPARTMENT</Level2Code>
  <Level3Code dtype="String">CLASS</Level3Code>
  <Level4Code dtype="String">SUBCLASS</Level4Code>
</MerchHierarchy>
```

2. DEPARTMENT through SUBCLASS mapping

```
<MerchHierarchy dtype="Default">
  <NumberOfLevels dtype="Integer">3</NumberOfLevels>
  <Level1Code dtype="String">DEPARTMENT</Level1Code>
  <Level2Code dtype="String">CLASS</Level2Code>
  <Level3Code dtype="String">SUBCLASS</Level3Code>
  <Level4Code dtype="String"/>
</MerchHierarchy>
```

3. CLASS through SUBCLASS mapping

```
<MerchHierarchy dtype="Default">
  <NumberOfLevels dtype="Integer">2</NumberOfLevels>
  <Level1Code dtype="String">CLASS</Level1Code>
  <Level2Code dtype="String">SUBCLASS</Level2Code>
```

```
    <Level3Code dtype="String"/>
    <Level4Code dtype="String"/>
  </MerchHierarchy>
```

4. SUBCLASS only mapping

```
<MerchHierarchy dtype="Default">
  <NumberOfLevels dtype="Integer">1</NumberOfLevels>
  <Level1Code dtype="String">SUBCLASS</Level1Code>
  <Level2Code dtype="String"/>
  <Level3Code dtype="String"/>
  <Level4Code dtype="String"/>
</MerchHierarchy>
```

The DataLoader uses dtv.dataloader.mom.MerchHierarchyTransformer to process RMS Merchandise Hierarchy records. There are no configuration options for the merchHierarachyTransformer bean.

```
<bean id="merchHierarchyTransformer"
class="dtv.dataloader.mom.MerchHierarchyTransformer"/>
```

Organizational Hierarchy

Organizational Hierarchy information provided by RMS is consumed by Xstore and Xcenter using the DataLoader.

The Dataloader uses dtv.dataloader.mom.OrgHierarchyTransformer to process RMS store records. The orgHierarchyTransformer bean has no configuration options.

```
<bean id="orgHierarchyTransformer"
class="dtv.dataloader.mom.OrgHierarchyTransformer" />
```

Related Item

RMS related items are communicated to Xstore in two separate extract files: Related Item Header and Related Item Detail. The rms_related_item_head table is a staging table supporting the processing of RelatedItemDetail records by the DataLoader. Xstore and Xcenter do not directly access the rms_related_item_head table.

Related Item Header

Related Item Header information provided by RMS is consumed by Xstore using the DataLoader. Related Item Header data is the primary source of rms_related_item_head records representing a header related Item record. RMS supports three types of related item relationships, Cross-Sell, Up-Sell, and Substitution. RMS Cross-Sell/Up-Sell records are interpreted as Xstore Attached Items. RMS Substitution Items are interpreted as Xstore substitution items.

The DataLoader uses dtv.dataloader.mom.RelatedItemHeadTransformer to process RMS Related Item Header records. There is one configuration option for the relitemHeadTransformer bean. For the case where itm_attached_items records are created, which occurs when a related item header update event results in the change of relationship type from SUBS to CRSL/UPSL, the value assigned the promptToAddMessageKey column will be the configured value.

```
<bean id="relitemHeadTransformer"
class="dtv.dataloader.mom.RelatedItemHeadTransformer" >
   <property name="promptToAddMessageKey" value="_commonAttachedItemsPrompt" />
</bean>
```

Related Item Detail

Related Item Detail information provided by RMS is consumed by Xstore using the DataLoader. Related Item Detail data is the primary source of itm_attached_items and itm_substitute_items records representing a detail related Item record. RMS supports three types of related item relationships, Cross-Sell, Up-Sell, and Substitution. RMS Cross-Sell/Up-Sell records are interpreted as Xstore Attached Items. RMS Substitution Items are interpreted as Xstore substitution items.

The DataLoader uses dtv.dataloader.mom.RelatedItemDetailTransformer to process RMS Related Item Detail records. There is one configuration options for the relitemDetailTransformer bean. The value configured for the promptToAddMessageKey property is used to populate the prompt_to_add_msg_key column of the itm_attached_items table for Cross-Sell and Up-Sell records.

```
<bean id="relitemDetailTransformer"
class="dtv.dataloader.mom.RelatedItemDetailTransformer" >
  <property name="promptToAddMessageKey" value="_commonAttachedItemsPrompt" />
</bean>
```

Store

Store information provided by RMS is consumed by Xstore using the DataLoader.

The Dataloader uses dtv.dataloader.mom.StoreTransformer to process RMS store records. The following configuration options are available:

- Configure the use_till_accountability_flag value assigned to all records created in loc_rtl_loc.
- Configure the location_type value assigned to all record created in loc_rtl_loc. This property is optional. If not set, location_type will be null.

```
<bean id="storeTransformer" class="dtv.dataloader.mom.StoreTransformer">
  <property name="useTillAccountability" value="false" />
  <property name="locationType" value="STORE" />
</bean>
```

Store Address

Store Address information provided by RMS is consumed by Xstore using the DataLoader. RMS Store Address records are interpreted as an instructions to update the Address information of existing Store records. Each location in RMS can have many types of addresses. There is a configuration to specify which RMS AddrType is to be recognized as a store address. In addition, RMS supports multiple addresses for the same addrType. However, only the primary address is used to populate the address fields of an Xstore store record.

The DataLoader uses dtv.dataloader.mom.StoreAddressTransformer to process RMS store records. One configuration option is available.

Configure the addrType value. Only records having this Address Type value, and where the record's PrimaryAddrInd is true, are used as the source for a Store's Address information.

```
<bean id="storeAddressTransformer"
class="dtv.dataloader.mom.StoreAddressTransformer">
   <property name="addrType" value="01" />
</bean>
```

VAT

VAT information provided by RMS is consumed by Xstore and Xcenter using the DataLoader. RMS VAT information is used to populate the Xstore tax tables for when using VAT. RMS provides a VAT Item file containing the information used to populate an item's TaxGroupId.

The Dataloader uses dtv.dataloader.mom.VATTransformer to process RMS store records. The VATTransformer bean has three configuration options:

- A Boolean flag indicating if tax is calculated at the transaction level.
- A rounding code, such as HALF_UP, HALF_DOWN, and so on, set into TaxAuthorityDAO.
- The number of rounding digits set into TaxAuthorityDAO.

```
<bean id="VATTransformer" class="dtv.dataloader.mom.VATTransformer">
      <constructor-arg type="boolean" value="true" />
      <constructor-arg type="java.lang.String" value="HALF_UP" />
      <constructor-arg type="int" value="2" />
    </bean>
```

VAT Item

VAT Item information provided by RMS is consumed by Xstore and Xcenter using the DataLoader. RMS VAT item information is used to populate an item's TaxGroupId with a VAT Code.

The Dataloader uses dtv.dataloader.mom.VATItemTransformer to process RMS store records. The VATItemTransformer bean does not have any configuration options.

```
<bean id="VATItemTransformer" class="dtv.dataloader.mom.VATItemTransformer" />
```

5

Integration Considerations

This chapter provides the considerations that should be taken into account 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.

Getting Started

If you are already live with RMS and RPM and are implementing Xstore, you should use the kill/fill export option in the RMS and RPM extracts to populate Xstore, 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.

Foundation Data

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

- Seed Data
- Transaction Details
- Currency Exchange Rates

Seed Data

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

- Currency codes
- Country codes
- Units of measure

Transaction Details

The mapping of transaction details from Xstore POSlog to ReSA RTLog depends on the mappings of valid values. These mappings are detailed in Appendix A. It is critical that the mappings are complete. If additional valid values are configured for Xstore in `RTLogMappingConfig.xml`, they must also be configured for ReSA 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 RMS 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: POSLog to RTLog Mapping Details for details on configuring and mapping these entities.

Currency Exchange Rates

Exchanges rates for currencies are not one of the things integrated between RMS and Xstore, as RMS is not considered the system of record for this information at a retailer - generally that comes from the financials solution. However, if you require currency exchange rates in Xstore, then it expected that the same source of data used for exchange rates in RMS 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 ReSA.

Stores

By default, Xstore is configured to allow four digit store IDs, but it can be configured to hold up to five digit store numbers. Although RMS can hold up to a ten 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 RMS. 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.

Merchandise Hierarchy

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

Following is the Xstore Merchandise Hierarchy configuration:

```
<MerchHierarchy dtype="Default">
<NumberOfLevels dtype="Integer">4</NumberOfLevels>
<Level1Code dtype="String">GROUP</Level1Code>
<Level2Code dtype="String">DEPARTMENT</Level2Code>
<Level3Code dtype="String">CLASS</Level3Code>
<Level4Code dtype="String">SUBCLASS</Level4Code>
</MerchHierarchy>
```

Items

This section lists considerations regarding items.

Merchandise Items

Physical merchandise items should be mastered in RMS 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 ReSA where the item being sold or returned cannot be identified and accounted for in RMS.

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 RMS. 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 RMS with the same ID as that created in Xcenter. The item created in RMS should be set up as a non-merchandise item to prevent it from being re-exported to Xstore.

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

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 Items

Kits, or pack items in RMS, 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 RMS, so these items will appear as standard items in Xstore and the component details will not be available.

Styles and Differentiators

Differentiators are used in RMS 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. RMS 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 RMS with Xstore, as it will be ignored in the integration.

Additionally, in RMS 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 RMS will be ignored and will not be visible in Xstore.

It is possible in RMS to range a child item to a store without also ranging its parent. However, it should be noted that in this case because the parent is not ranged, child items (SKUs) associated with the style cannot be selected by scanning the style ID in Xstore; it can only be looked up by the child IDs (SKUs). In addition, when the style does not exist, the item dimension values are not displayed in the Xstore UI.

Finally, it should be noted that RMS sends parent item/location combinations as well as transaction item/location combinations to Xstore. If you have transaction items in RMS with a parent item that is not deemed a "style" by Xstore per the definition above, the parent item/location combination will be recorded in the Xstore item/location table (itm_item_options), but may not have a corresponding record in the item table (itm_item). This does not cause any systematic issues, but be aware that the data in the tables may not match in this case.

Product Restrictions

Product restrictions can be set up in RMS 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.

Related Items

RMS 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 RMS 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 RMS are ignored.

Other Item Attribute Notes

- Item Restocking - unlike Xstore, RMS 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 RMS. Therefore, this function would not be available in Xstore.
- RMS has the ability for retailers to extend the available item attribution by creating user defined attributes and custom flex attributes. Although included in the available data from RMS, these are currently not used by Xstore.
- Xstore uses the item level description (which is communicated in the primary RMS 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 RMS be updated to show the localized item description.

Tax

This section describes considerations regarding taxes.

Value Added Tax (VAT)

RMS 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 RMS and communicated as such to Xstore. For more information on configuration for VAT in Xstore, see the *Oracle Retail Xstore Technical Guide*.

When RMS 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 RMS 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

RMS 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.

In standalone mode, DataLoader has to be executed twice, first to import RMS files and second to import this `.mnt` file. If they are placed together in the download directory, `.mnt` files always get loaded first.

In an integrated environment with Xstore Office/Xenvironment, the retailer has to drop the `.momzip` file first, and wait until that zip file is processed by Xstore Office before dropping this `.mnt` file. This guarantees the `.mnt` file is imported into Xcenter after all RMS files, and is staged for store deployment with a deployment ID greater than the deployment ID of the RMS files.

After loading RMS 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 t`ax_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 RMS 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.

Inventory

Inventory functionality in Xstore should be disabled when implemented with RMS. No inventory information is integrated between Xstore and RMS, 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 RMS. 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 ReSA/RMS and Xstore is not supported.

Serialized Inventory

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

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

Customer Orders

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

In-Store Orders

Orders taken in the store on behalf of a customer that do not go through an Order Management System (OMS) for fulfillment will include only a customer order number, but not a fulfillment order number when it the transactions related to it are integrated to ReSA.

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 RMS 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 RMS.

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 RMS and RPM 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 RMS and RPM)
- Manufacturer's Recommended Retail (from RMS)

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

Multi-Unit Pricing

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

Promotions

RPM has a system option (`RPM_SYSTEM_OPTIONS.COMPLEX_PROMO_ALLOWED_IND`) to not allow users to use any complex promotions, which includes all Multi-Buy Threshold, Transaction, and Finance promotions. This system option can be used to prevent end users from creating complex promotions that are not supported by Xstore including. However, this system option would also prevent the ability to create some complex promotions that are supported in both solutions. So, consideration should be taken to determine whether that is the right approach or whether a business process should be used to prevent users from creating unsupported promotions.

Promotions supported in RPM but not in Xstore include the following:

- Multi-Buy promotions with a reward of cheapest free (Buy N and get Cheapest Free)
- Multi-Buy promotions that use the OR condition between buy lists and/or reward lists
- Threshold promotions with a qualifier of item level
- Finance promotions

- Customer Segment promotions - Xstore can support customer segment promotions, but it is not supported in the integration with RPM

Simple Promotions

When Simple Promotions are created in RPM, the promotional retail for each item/location is calculated based on the promotion criteria, rounding rules, and taking into consideration any promotion overlaps. The retail from RPM is provided to RMS, SIM, and Xstore. However, Xstore also calculates a retail price based on the promotion criteria and ignores the value sent by RPM. This could result a slight difference in expected promotional price between solutions.

Coupon-Based Promotions

RPM does not support tying a coupon to a promotion and the RMS coupon functionality is not integrated with Xstore. It is recommended that if coupons are required for your business that these be managed in Oracle Retail Customer engagement and leverage the base integration of that solution with Xstore. Or, if managed in an external solution, these could be integrated to Xstore as part of your implementation.

Sales Audit

This section describes sales audit considerations.

Register-Level Balancing

Xstore workstation and ReSA register are equivalent concepts; however ReSA does not have an entity equivalent to the Xstore till, which means that Xstore cannot be configured for till-level balancing when integrated with ReSA. When integrated with Xstore, ReSA should be configured with a balancing level of Register and Xstore will always sends the workstation ID as the register.

Sales Person

In Xstore, the sales person field length can be up to 60 characters in length, but ReSA 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 ReSA only supports one. Therefore only the transaction level sales associate is exported to ReSA.

Tender Type

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

Coupons

Bounce back coupon number length in Xstore can be 60 characters long, but ReSA 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.

6

RTLog Generator

This chapter describes how to install, deploy, and configure the RTLog Generator application.

RTLog Generator is a Java and XML based web application that exposes a Spring-JAXWS implemented SOAP web service. It is distributed as a web archive along with a configuration zip file ready to be deployed on an Oracle WebLogic 12c server.

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

Configuration

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

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

Starting from Xstore release 18.0, the RTLog Generator application is shipped with two configuration zip files:

- rtlog-gen-config-resa-cs.zip
- rtlog-gen-config-resa-onprem.zip

To integrate with ReSA on cloud, rtlog-gen-config-resa-cs.zip should be used to externally configure and extend the RTLog Generator's functionality.

To integrate with ReSA on premise, rtlog-gen-config-resa-onprem.zip should be used to externally configure and extend the RTLog Generator's functionality.

To set up the external configuration features:

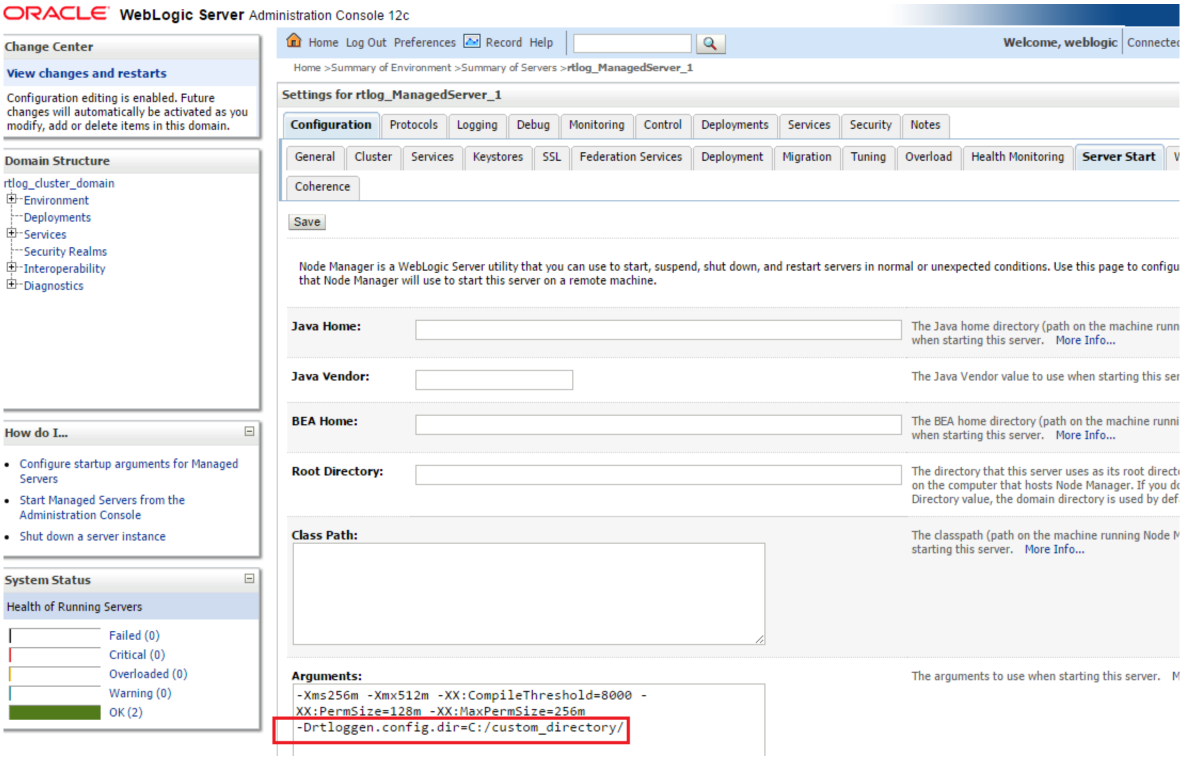
1. Extract the configuration file's content into the C:\<*rtlog-gen-config*> directory if installing on Microsoft Windows or /usr/local/<*rtlog-gen-config*> on Linux OS. These directories are the default locations where the RTLog Generator application will look for the configuration files. These default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

   - Pass a JVM argument to the server startup script and bounce the server:

     -Drtloggen.config.dir=C:/<*custom_directory*>/

If the WebLogic domain is created with a Node manager, the same argument can be passed from the Administration Console in the Arguments field. See Figure 6–1.

Figure 6–1 Administration Console Configuration Page


ORACLE WebLogic Server Administration Console 12c
Home Log Out Preferences Record Help
Welcome, weblogic Connected
Change Center
View changes and restarts
Domain Structure
rtlog_cluster_domain
Environment
Deployments
Services
Security Realms
Interoperability
Diagnostics
How do I...
System Status
Health of Running Servers
Failed (0)
Critical (0)
Overloaded (0)
Warning (0)
OK (2)
Home >Summary of Environment >Summary of Servers >rtlog_ManagedServer_1
Settings for rtlog_ManagedServer_1
Configuration Protocols Logging Debug Monitoring Control Deployments Services Security Notes
General Cluster Services Keystores SSL Federation Services Deployment Migration Tuning Overload Health Monitoring Server Start V
Coherence
Save

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

- Configure startup arguments for Managed Servers
- Start Managed Servers from the Administration Console
- Shut down a server instance

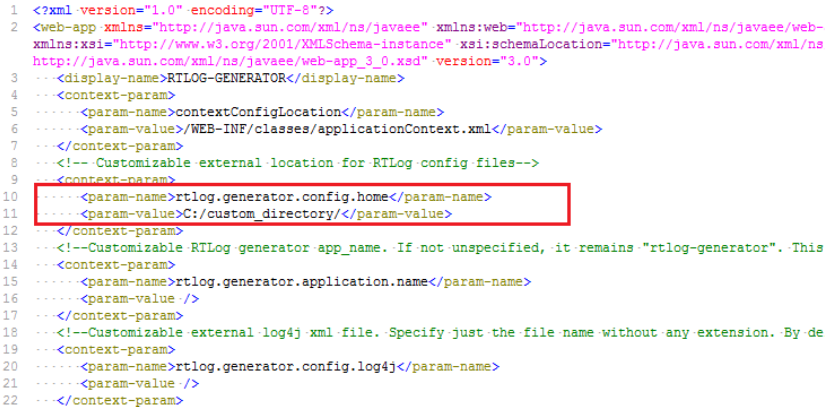
Node Manager is a WebLogic Server utility that you can use to start, suspend, shut down, and restart servers in normal or unexpected conditions. Use this page to configu that Node Manager will use to start this server on a remote machine.

Java Home:		The Java home directory (path on the machine runn when starting this server. More Info...
Java Vendor:		The Java Vendor value to use when starting this ser
BEA Home:		The BEA home directory (path on the machine runni when starting this server. More Info...
Root Directory:		The directory that this server uses as its root directo on the computer that hosts Node Manager. If you do Directory value, the domain directory is used by def
Class Path:		The classpath (path on the machine running Node M starting this server. More Info...
Arguments:	-Xms256m -Xmx512m -XX:CompileThreshold=8000 -XX:PermSize=128m -XX:MaxPermSize=256m -Drtloggen.config.dir=C:/custom_directory/	The arguments to use when starting this server. M

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

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

Figure 6–2 Example of context-param Field Update



```
<?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>
```

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

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

   - rtlogconfig.properties:

     This file contains three properties (key value pairs):

     – processingDir: This directory path specifies the location that RTLog Generator will use to build its RTLog files as it receives data from Xstore Office. This directory needs to be created manually.

     – resaFileDropDir: This directory path specifies the destination for the RTLog files this system is producing. It should be configured to the location where ReSA is looking to receive the RTLog files. This directory needs to be created manually.

     – clusterNodeNumber: This property should only be enabled when running in a clustered environment. For more information, see "WebLogic Cluster Setup".

     Following is an example of the three properties:

     ```
     processingDir = C:/RTLOG_Weblogic/Output/Store/RTLOGS
     resaFileDropDir = C:/RTLOG_Weblogic/Output/ReSA
     clusterNodeNumber = 1
     ```

   - RTLogFormatConfig.xml:

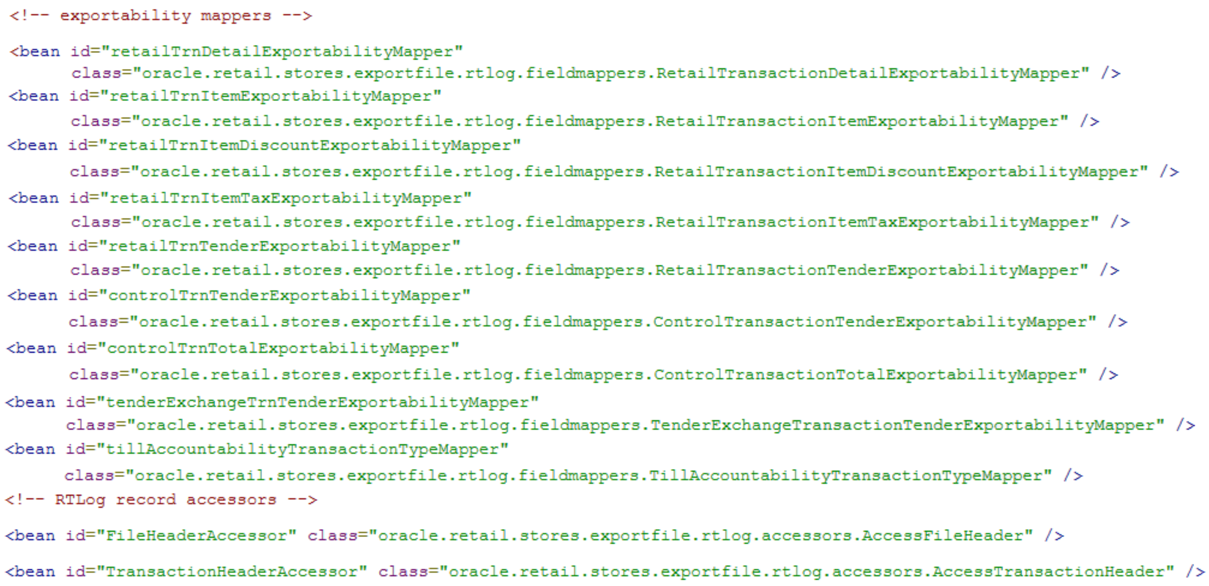
     This file specifies the format of the RTLog record as specified by ReSA. You do not make any changes to this file.

   - rtlog-generator-log4j.xml: This file configures the logging levels for the RTLog Generator application.

   - RTLogMappingBean.xml:

     This is a spring configuration XML file that provides metadata for the FieldMapper and Record Accessor beans which get injected into the RTLog Generator business logic classes. The following example is an excerpt from this file:

Figure 6–3 RTLogMappingBean.xml File Excerpt



```
<!-- exportability mappers -->

<bean id="retailTrnDetailExportabilityMapper"
      class="oracle.retail.stores.exportfile.rtlog.fieldmappers.RetailTransactionDetailExportabilityMapper" />
<bean id="retailTrnItemExportabilityMapper"
      class="oracle.retail.stores.exportfile.rtlog.fieldmappers.RetailTransactionItemExportabilityMapper" />
<bean id="retailTrnItemDiscountExportabilityMapper"
      class="oracle.retail.stores.exportfile.rtlog.fieldmappers.RetailTransactionItemDiscountExportabilityMapper" />
<bean id="retailTrnItemTaxExportabilityMapper"
      class="oracle.retail.stores.exportfile.rtlog.fieldmappers.RetailTransactionItemTaxExportabilityMapper" />
<bean id="retailTrnTenderExportabilityMapper"
      class="oracle.retail.stores.exportfile.rtlog.fieldmappers.RetailTransactionTenderExportabilityMapper" />
<bean id="controlTrnTenderExportabilityMapper"
      class="oracle.retail.stores.exportfile.rtlog.fieldmappers.ControlTransactionTenderExportabilityMapper" />
<bean id="controlTrnTotalExportabilityMapper"
      class="oracle.retail.stores.exportfile.rtlog.fieldmappers.ControlTransactionTotalExportabilityMapper" />
<bean id="tenderExchangeTrnTenderExportabilityMapper"
      class="oracle.retail.stores.exportfile.rtlog.fieldmappers.TenderExchangeTransactionTenderExportabilityMapper" />
<bean id="tillAccountabilityTransactionTypeMapper"
      class="oracle.retail.stores.exportfile.rtlog.fieldmappers.TillAccountabilityTransactionTypeMapper" />
<!-- RTLog record accessors -->

<bean id="FileHeaderAccessor" class="oracle.retail.stores.exportfile.rtlog.accessors.AccessFileHeader" />

<bean id="TransactionHeaderAccessor" class="oracle.retail.stores.exportfile.rtlog.accessors.AccessTransactionHeader" />
```

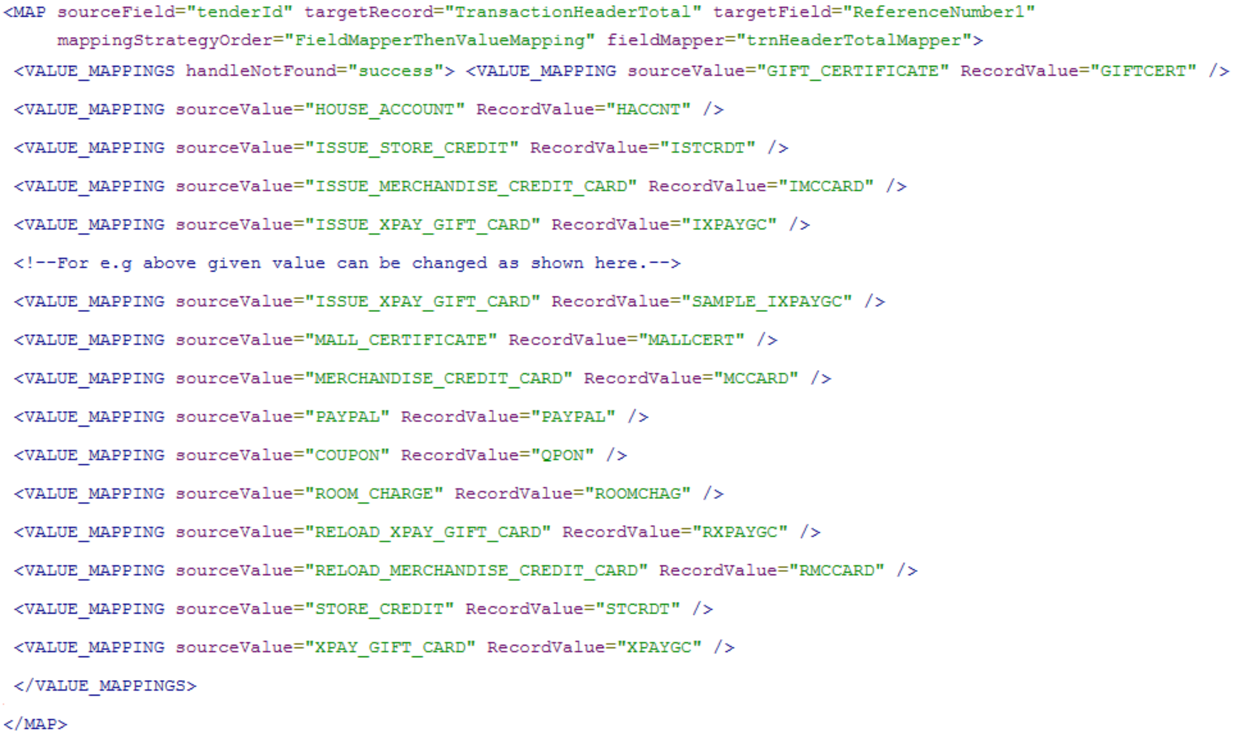
- RTLogMappingConfig.xml:

  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>
```

Figure 6–4 RTLogMappingConfig.xml Field Mapper Example 1

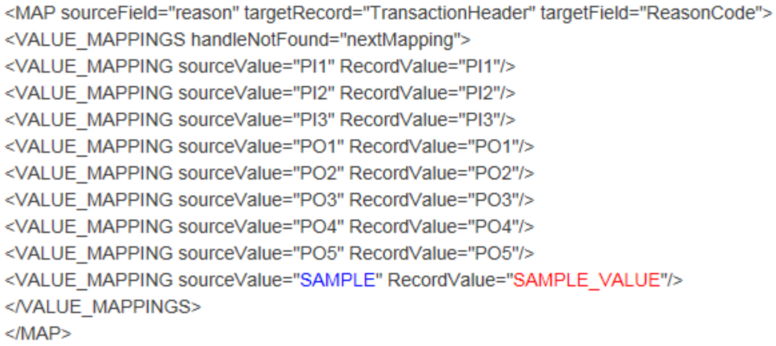


```
<MAP sourceField="tenderId" targetRecord="TransactionHeaderTotal" targetField="ReferenceNumber1"
     mappingStrategyOrder="FieldMapperThenValueMapping" fieldMapper="trnHeaderTotalMapper">
 <VALUE_MAPPINGS handleNotFound="success"> <VALUE_MAPPING sourceValue="GIFT_CERTIFICATE" RecordValue="GIFTCERT" />
 <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>
```

Figure 6–5 RTLogMappingConfig.xml Field Mapper Example 2

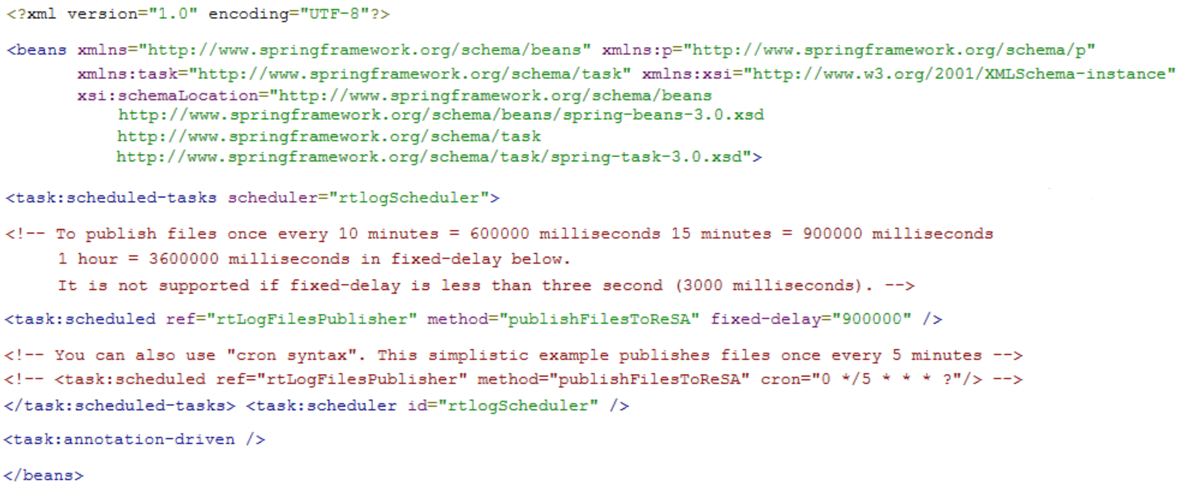


```
<MAP sourceField="reason" targetRecord="TransactionHeader" targetField="ReasonCode">
<VALUE_MAPPINGS handleNotFound="nextMapping">
<VALUE_MAPPING sourceValue="PI1" RecordValue="PI1"/>
<VALUE_MAPPING sourceValue="PI2" RecordValue="PI2"/>
<VALUE_MAPPING sourceValue="PI3" RecordValue="PI3"/>
<VALUE_MAPPING 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>
```

- 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 6–6 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>
```

Deployment

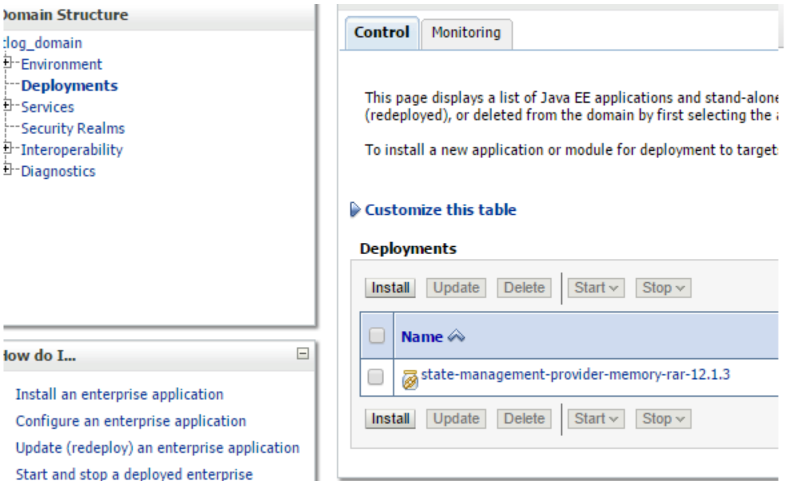
If you are deploying in a cluster, first set up a WebLogic cluster. For more information, see "WebLogic Cluster Setup".

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

To deploy the RTLog Generator application:

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

Figure 6–7 Administration Console Control Page



)omain Structure

:log_domain
- Environment
- Deployments
- Services
- Security Realms
- Interoperability
- Diagnostics

Iow do I...

Install an enterprise application

Configure an enterprise application

Update (redeploy) an enterprise application

Start and stop a deployed enterprise

This page displays a list of Java EE applications and stand-alone (redeployed), or deleted from the domain by first selecting the a

To install a new application or module for deployment to target

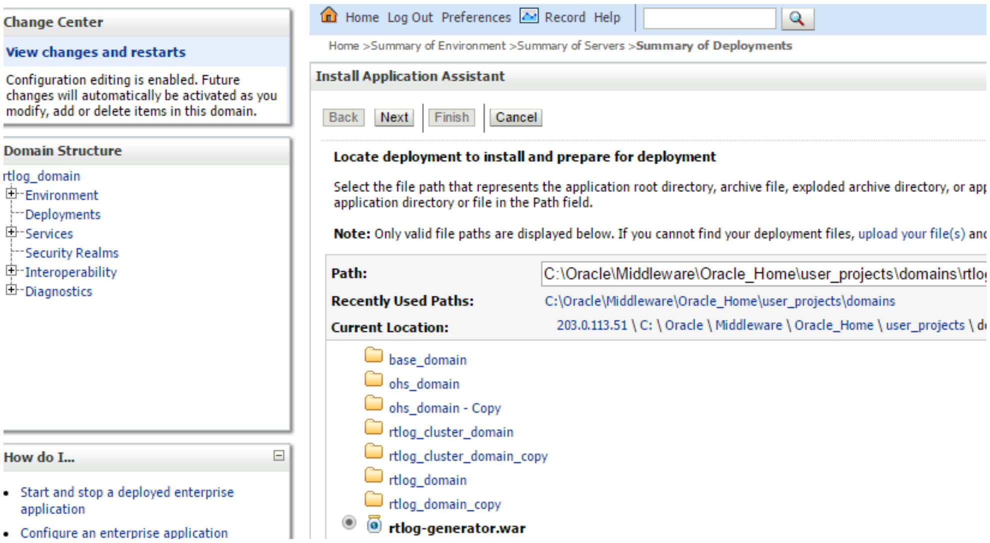
Customize this table

Deployments

□	Name
□	state-management-provider-memory-rar-12.1.3

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

Figure 6–8 Administration Console Install Application Assistant Page



Change Center

View changes and restarts

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

Domain Structure

rtlog_domain
- Environment
- Deployments
- Services
- Security Realms
- Interoperability
- Diagnostics

How do I...

- Start and stop a deployed enterprise application
- Configure an enterprise application

Home >Summary of Environment >Summary of Servers >Summary of Deployments

Install Application Assistant

Locate deployment to install and prepare for deployment

Select the file path that represents the application root directory, archive file, exploded archive directory, or app application directory or file in the Path field.

Note: Only valid file paths are displayed below. If you cannot find your deployment files, upload your file(s) and

Path: C:\Oracle\Middleware\Oracle_Home\user_projects\domains\rtlo

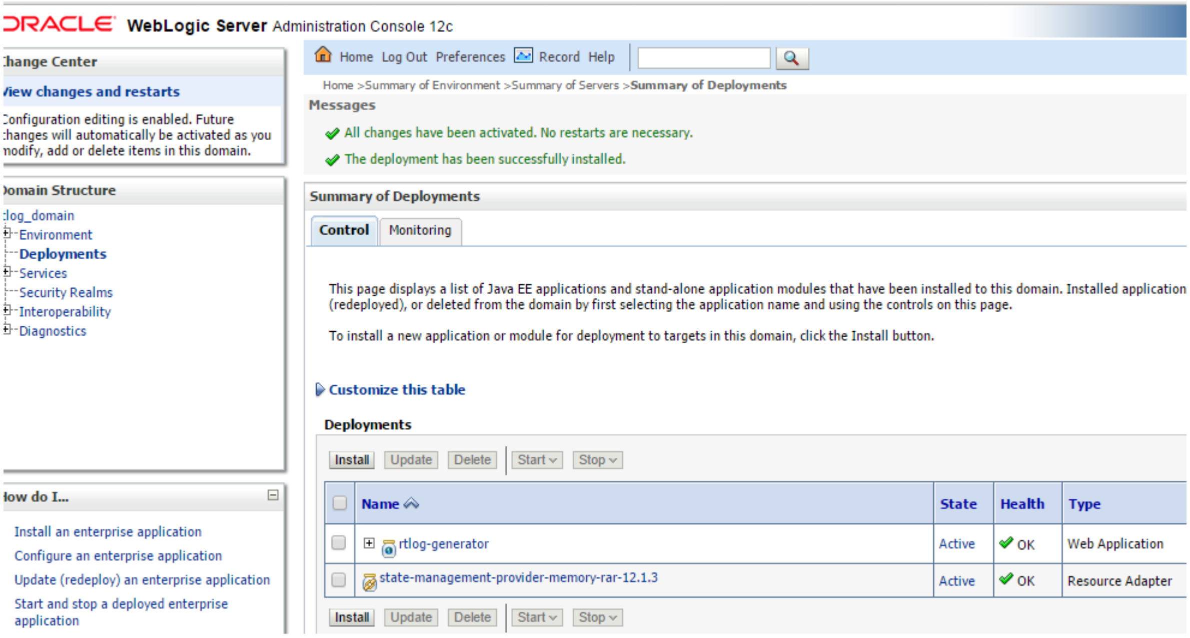
Recently Used Paths: C:\Oracle\Middleware\Oracle_Home\user_projects\domains

Current Location: 203.0.113.51 \ C: \ Oracle \ Middleware \ Oracle_Home \ user_projects \ d

- base_domain
- ohs_domain
- ohs_domain - Copy
- rtlog_cluster_domain
- rtlog_cluster_domain_copy
- rtlog_domain
- rtlog_domain_copy
- ● **rtlog-generator.war**

5. Click **Next** and then **Finish**. Once deployed, RTLog Generator should be listed as one of the deployed applications as shown in Figure 6–9.

Figure 6–9 Administration Console Summary of Deployments



WebLogic Server Administration Console 12c
Home Log Out Preferences Record Help
Home >Summary of Environment >Summary of Servers >Summary of Deployments
hange Center
iew changes and restarts
omain Structure
log_domain
Environment
Deployments
Services
Security Realms
Interoperability
Diagnostics
ow do I...
Install an enterprise application
Configure an enterprise application
Update (redeploy) an enterprise application
Start and stop a deployed enterprise application
Summary of Deployments
Control
Monitoring
Customize this table
Install
Update
Delete
Start
Stop

onfiguration editing is enabled. Future hanges will automatically be activated as you nodify, add or delete items in this domain.

Messages

✓ All changes have been activated. No restarts are necessary.

✓ The deployment has been successfully installed.

This page displays a list of Java EE applications and stand-alone application modules that have been installed to this domain. Installed application (redeployed), or deleted from the domain by first selecting the application name and using the controls on this page.

To install a new application or module for deployment to targets in this domain, click the Install button.

Deployments

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

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

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

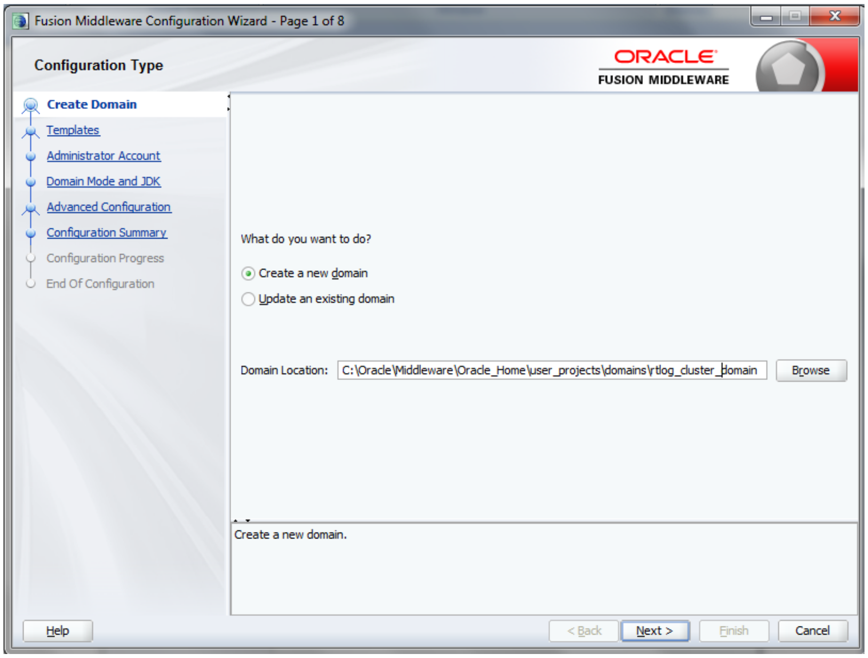
WebLogic Cluster Setup

> **Note:** WebLogic 12c must be installed on all the clustered machines and the exact same installed directory location must be used on all the machines.

To set up the cluster to use RTLog Generator:

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

Figure 6–10 Configuration Wizard Configuration Type Page



What do you want to do?

● Create a new domain

○ Update an existing domain

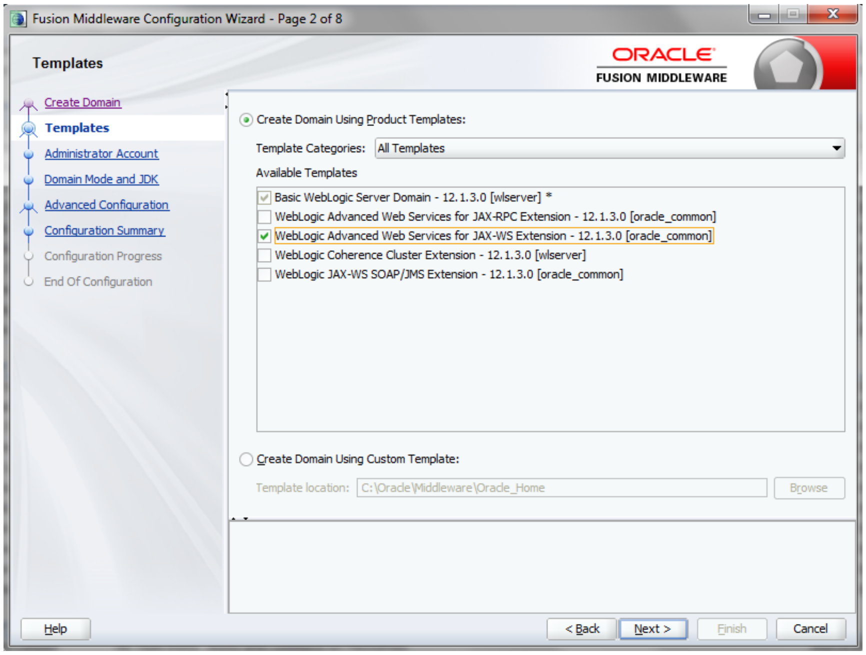
Domain Location: C:\Oracle\Middleware\Oracle_Home\user_projects\domains\rtlog_cluster_domain

Create a new domain.

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

```
WebLogic Advanced Web Services for JAX-WS Extension - 12.1.3.0 [oracle_common]
```

Figure 6–11 Configuration Wizard Templates Page



● Create Domain Using Product Templates:

Template Categories: All Templates

Available Templates

☑ Basic WebLogic Server Domain - 12.1.3.0 [wlserver] *

☐ WebLogic Advanced Web Services for JAX-RPC Extension - 12.1.3.0 [oracle_common]

☑ WebLogic Advanced Web Services for JAX-WS Extension - 12.1.3.0 [oracle_common]

☐ WebLogic Coherence Cluster Extension - 12.1.3.0 [wlserver]

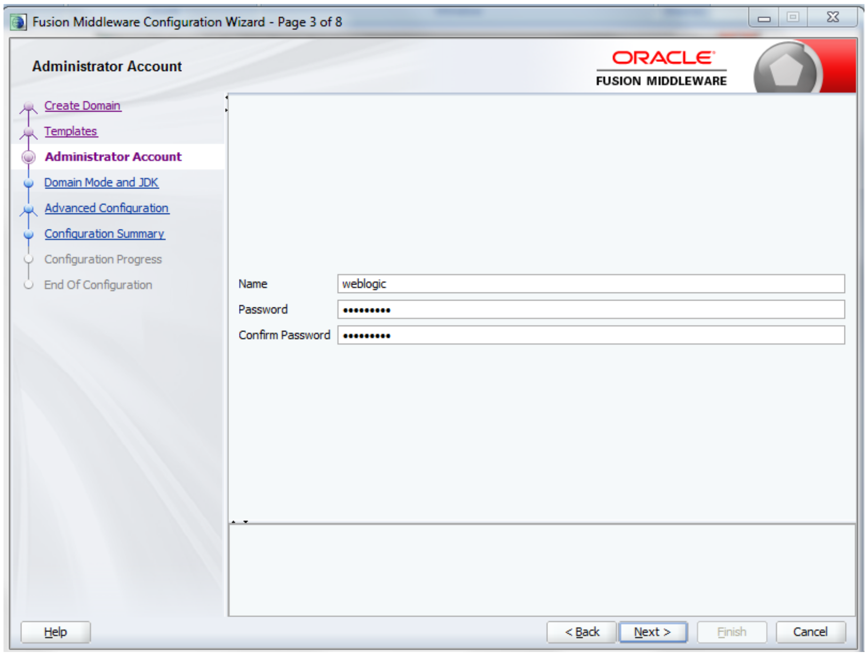
☐ WebLogic JAX-WS SOAP/JMS Extension - 12.1.3.0 [oracle_common]

○ Create Domain Using Custom Template:

Template location: C:\Oracle\Middleware\Oracle_Home

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

Figure 6–12 Configuration Wizard Administrator Account Page



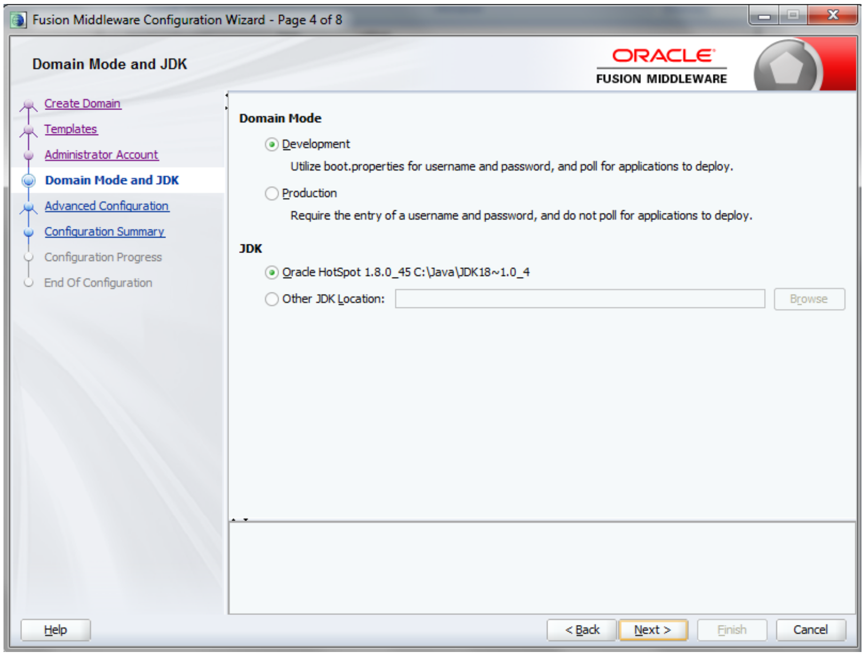
Name: weblogic

Password: •••••••••

Confirm Password: •••••••••

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

Figure 6–13 Configuration Wizard Domain Mode and JDK



Domain Mode

● Development

Utilize boot.properties for username and password, and poll for applications to deploy.

○ Production

Require the entry of a username and password, and do not poll for applications to deploy.

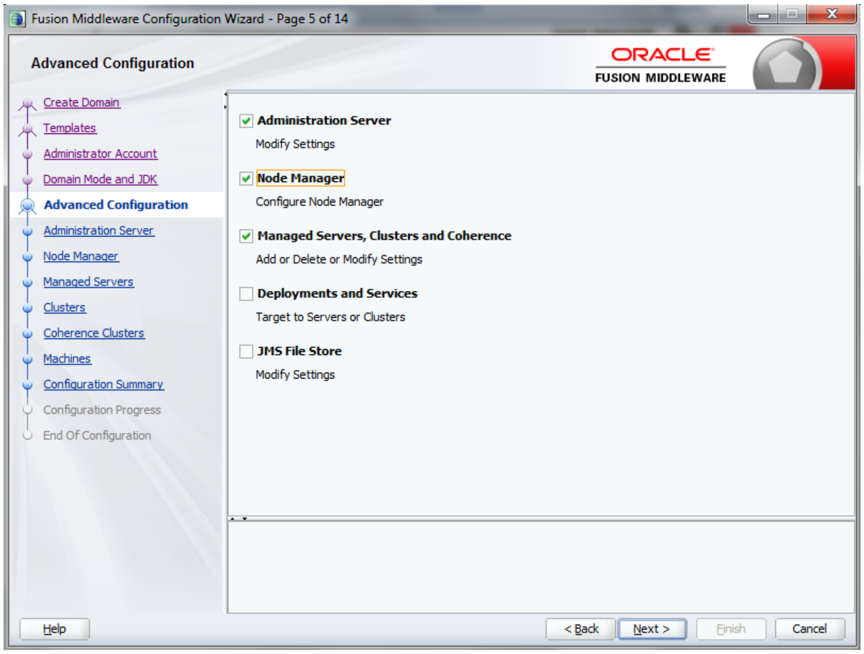
JDK

● Oracle HotSpot 1.8.0_45 C:\Java\JDK18~1.0_4

○ Other JDK Location:

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

Figure 6–14 Configuration Wizard Advanced Configuration Page



☑ **Administration Server**
Modify Settings

☑ **Node Manager**
Configure Node Manager

☑ **Managed Servers, Clusters and Coherence**
Add or Delete or Modify Settings

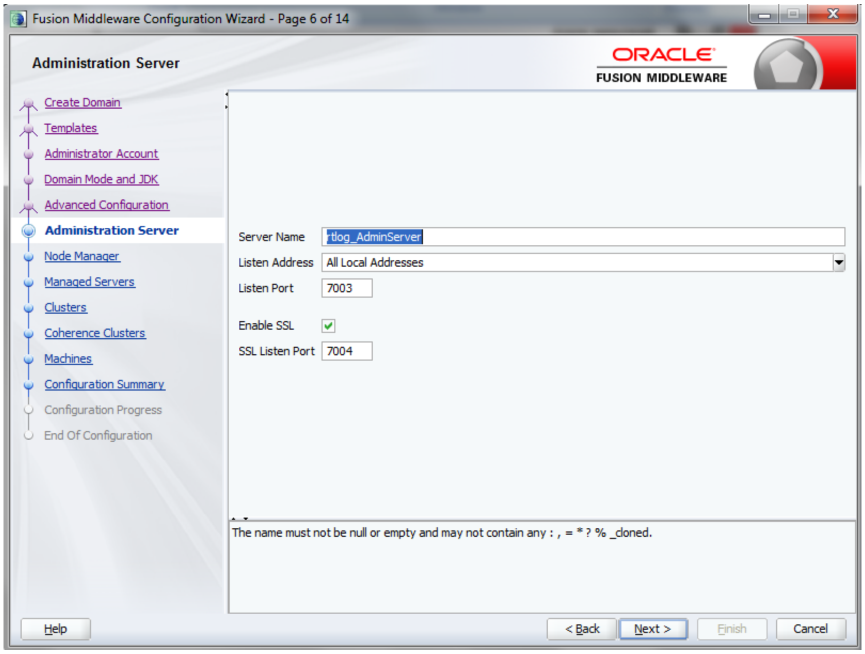
☐ **Deployments and Services**
Target to Servers or Clusters

☐ **JMS File Store**
Modify Settings

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

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

Figure 6–15 Configuration Wizard Administration Server Page

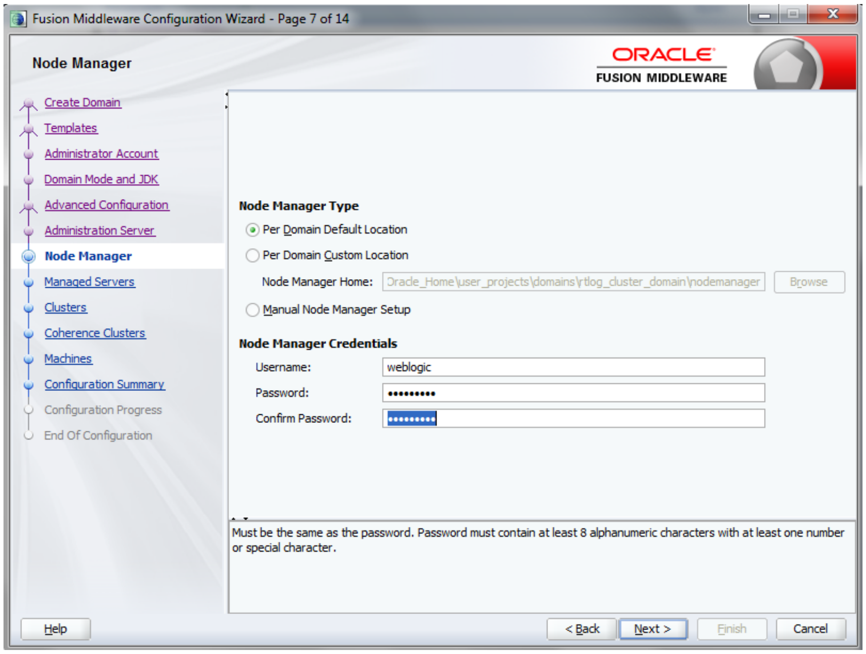


Server Name: rtlog_AdminServer
Listen Address: All Local Addresses
Listen Port: 7003
Enable SSL: ☑
SSL Listen Port: 7004

The name must not be null or empty and may not contain any : , = * ? % _cloned.

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

Figure 6–16 Configuration Wizard Node Manager Page



Fusion Middleware Configuration Wizard - Page 7 of 14

Node Manager

Node Manager Type

● Per Domain Default Location

○ Per Domain Custom Location

Node Manager Home: Oracle_Home\user_projects\domains\rtlog_cluster_domain\nodemanager

○ Manual Node Manager Setup

Node Manager Credentials

Username: weblogic

Password: •••••••••

Confirm Password: •••••••••

Must be the same as the password. Password must contain at least 8 alphanumeric characters with at least one number or special character.

9. On the Managed Servers page, add and configure each managed server:

   a. For the listen address, enter the IP address of the managed server. Do not select All local Addresses.

   b. rtlog_ManagedServer_1 will be running on Machine_1 in this configuration. Enter the Machine_1 IP address for the server.

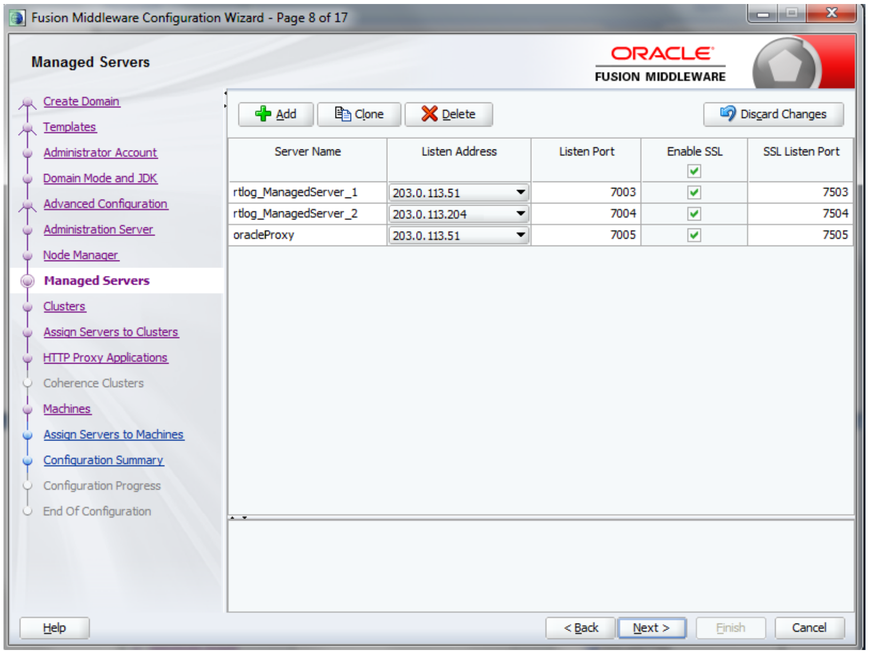
   c. rtlog_ManagedServer_2 will be running on Machine_2 in this configuration. Enter the Machine_2 IP address for this server.

   d. oracleProxy is running on Machine_1, but is not a part of the cluster. It is an Oracle proxy HTTP cluster servlet used for failover and load balancing purposes. Enter the Machine_1 IP address for this server.

   e. Enable SSL for all the managed servers.

   f. Click **Next**.

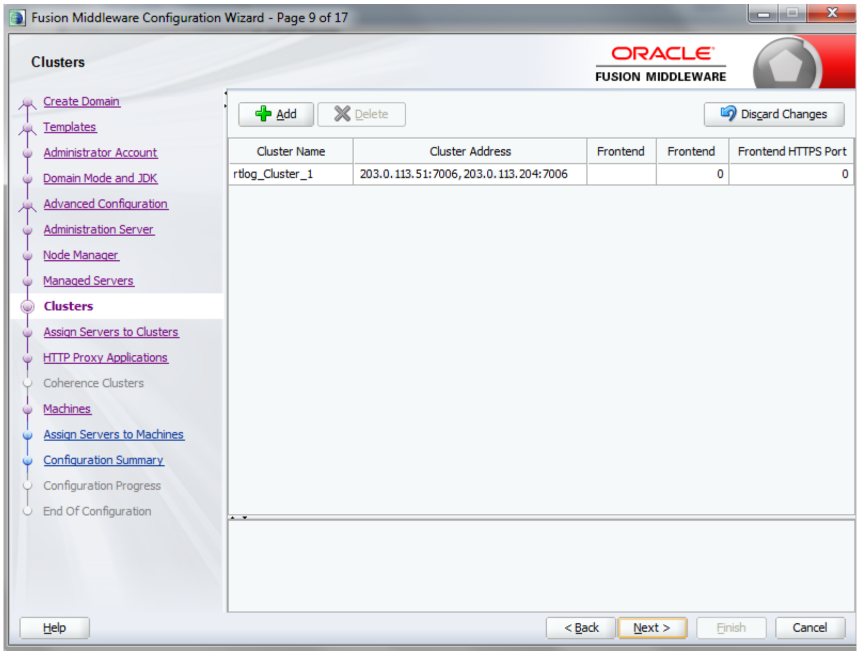
Figure 6–17 Configuration Wizard Managed Servers Page



Server Name	Listen Address	Listen Port	Enable SSL ☑	SSL Listen Port
rtlog_ManagedServer_1	203.0.113.51	7003	☑	7503
rtlog_ManagedServer_2	203.0.113.204	7004	☑	7504
oracleProxy	203.0.113.51	7005	☑	7505

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

Figure 6–18 Configuration Wizard Clusters Page

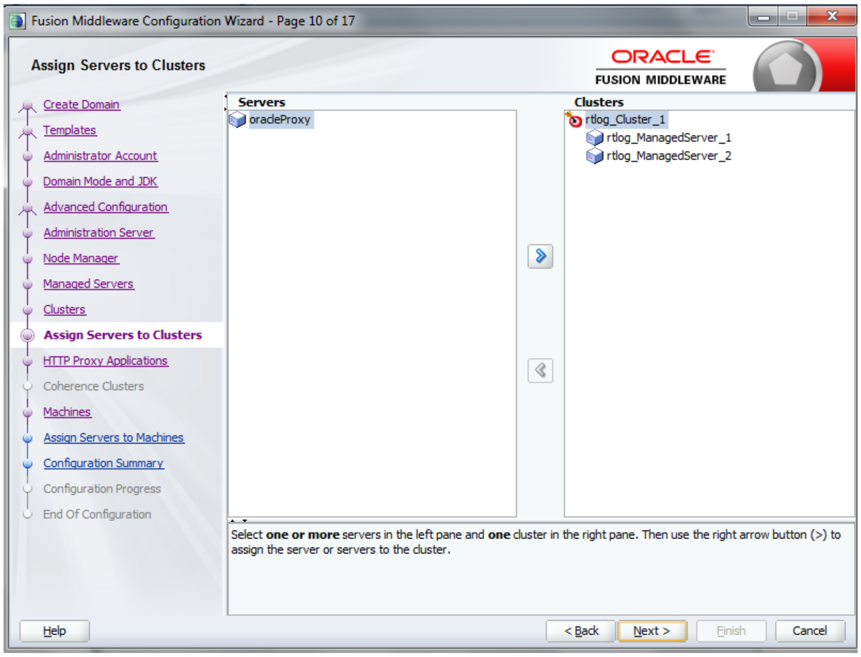


Cluster Name	Cluster Address	Frontend	Frontend	Frontend HTTPS Port
rtlog_Cluster_1	203.0.113.51:7006,203.0.113.204:7006		0	0

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

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

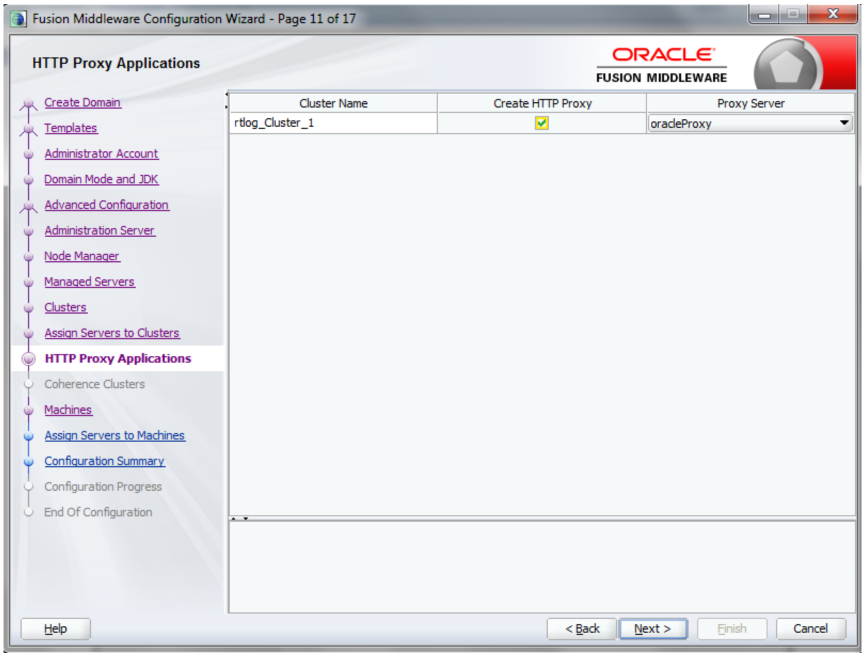
Figure 6–19 Configuration Wizard Assign Servers to Clusters Page



Select **one or more** servers in the left pane and **one** cluster in the right pane. Then use the right arrow button (>) to assign the server or servers to the cluster.

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

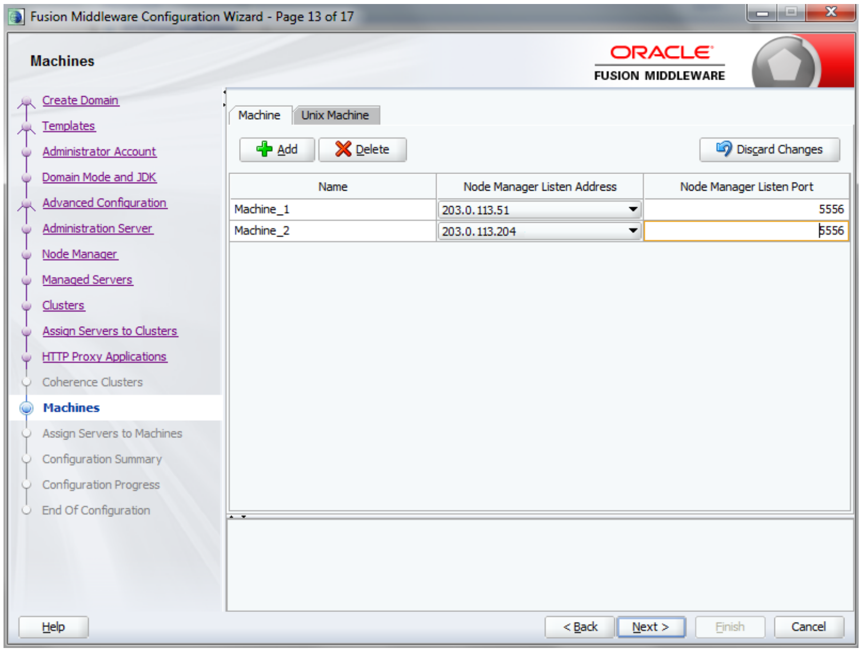
Figure 6–20 Configuration Wizard HTTP Proxy Applications Page



Cluster Name	Create HTTP Proxy	Proxy Server
rtlog_Cluster_1	☑	oracleProxy

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

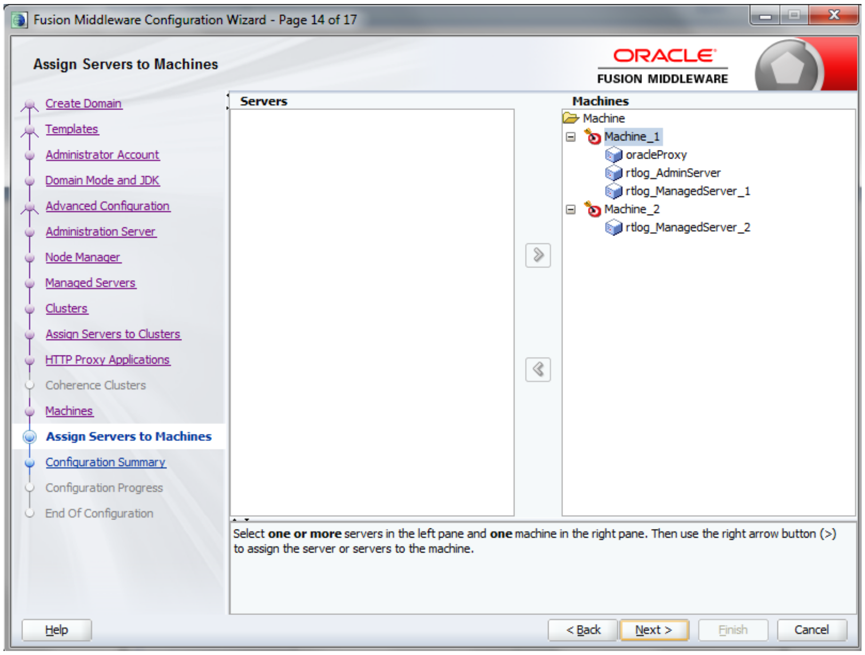
Figure 6–21 Configuration Wizard Machines Page



Name	Node Manager Listen Address	Node Manager Listen Port
Machine_1	203.0.113.51	5556
Machine_2	203.0.113.204	5556

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

Figure 6–22 Configuration Wizard Assign Servers to Machines Page

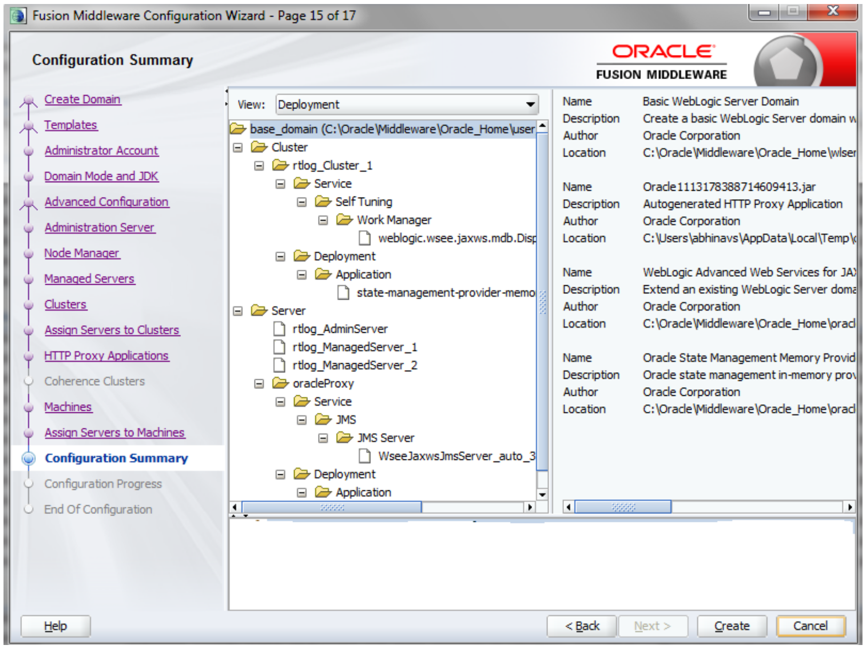


```
Machine
  Machine_1
    oracleProxy
    rtlog_AdminServer
    rtlog_ManagedServer_1
  Machine_2
    rtlog_ManagedServer_2
```

Select **one or more** servers in the left pane and **one** machine in the right pane. Then use the right arrow button (>) to assign the server or servers to the machine.

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

Figure 6–23 Configuration Wizard Configuration Summary Page



```
base_domain (C:\Oracle\Middleware\Oracle_Home\user
  Cluster
    rtlog_Cluster_1
      Service
        Self Tuning
          Work Manager
            weblogic.wsee.jaxws.mdb.Disp
      Deployment
        Application
          state-management-provider-memo
  Server
    rtlog_AdminServer
    rtlog_ManagedServer_1
    rtlog_ManagedServer_2
    oracleProxy
      Service
        JMS
          JMS Server
            WseeJaxwsJmsServer_auto_3
      Deployment
        Application
```

Name	Basic WebLogic Server Domain
Description	Create a basic WebLogic Server domain w
Author	Oracle Corporation
Location	C:\Oracle\Middleware\Oracle_Home\wlser
Name	Oracle1113178388714609413.jar
Description	Autogenerated HTTP Proxy Application
Author	Oracle Corporation
Location	C:\Users\abhinavs\AppData\Local\Temp\
Name	WebLogic Advanced Web Services for JAX
Description	Extend an existing WebLogic Server doma
Author	Oracle Corporation
Location	C:\Oracle\Middleware\Oracle_Home\oracl
Name	Oracle State Management Memory Provid
Description	Oracle state management in-memory prov
Author	Oracle Corporation
Location	C:\Oracle\Middleware\Oracle_Home\oracl

To complete the configuration of the cluster:

1. Start and stop the node manager. You can find the start up script inside the newly created domain, that is, the <*rtlog_clust_domain*>\bin directory.

2. In the nodemanager.properties file, set SecureListener=false. This file is found in the <*rtlog_clust_domain*>\nodemanager directory.

3. Edit the <*rtlog_clust_domain*>\config\config.xml file. Use plain communication for the node managers by updating the communication type for the node managers as shown in the following example:

```
<machine>
       <name>Machine_1</name>
       <node-manager>
      <name>Machine_1</name>
      <nm-type>Plain</nm-type>
      <listen-address>203.0.113.51</listen-address>
    </node-manager>
 </machine>
 <machine>
    <name>Machine_2</name>
    <node-manager>
      <name>Machine_2</name>
      <nm-type>Plain</nm-type>
      <listen-address>203.0.113.204</listen-address>
    </node-manager>
  </machine>
```

4. If the <*rtlog_clust_domain*> is created with the production mode option:

   a. Run <*rtlog_clust_domain*>\startWeblogic.cmd for the first time. This creates the servers folders under the domain. Enter the administration user name and password.

   b. Create a folder named security under the <*rtlog_clust_domain*>\servers\Admin server.

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

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

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

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

5. Pack the created domain:

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

<*WL_HOME*>\wlserver\common\bin\pack.cmd

Run the following command:

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

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

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

unpack.cmd -template=<*WL_HOME*>\user_projects\domains\<*user_templates*>\rtlog_cluster_domain.jar -domain=<*WL_HOME*>\user_projects\domains\rtlog_cluster_domain

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

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

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

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

c. Run the following command:

connect ('adminServer_username', 'adminServer_password','t3://Machine_1_IPAddress:Admin_server_unsecured_port')

For example: connect ('weblogic','weblogic1','t3://203.0.113.51:7003')

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

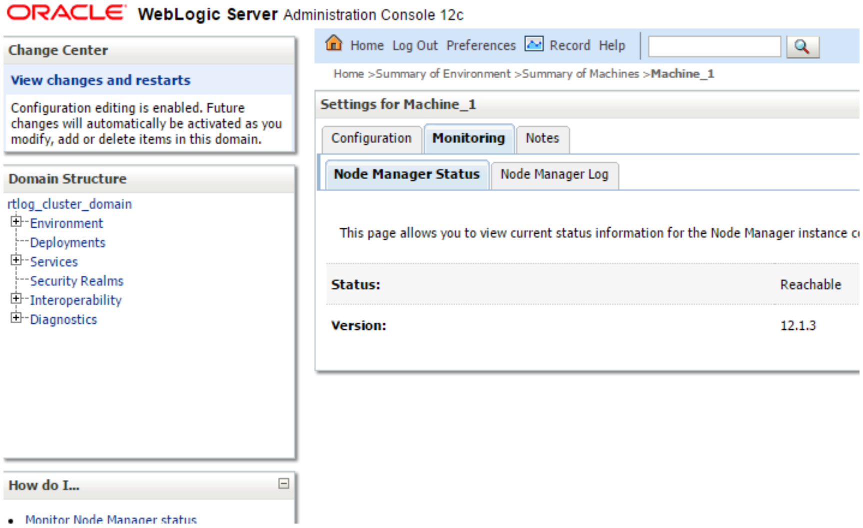
nmEnroll ('<*WL_HOME*>/user_projects/domains/<*rtlog_cluster_domain*>','<*WL_HOME*>/user_projects/domains/<*rtlog_cluster_domain*>/nodemanager')

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

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

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

Figure 6–24 Administration Console Settings Page



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

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

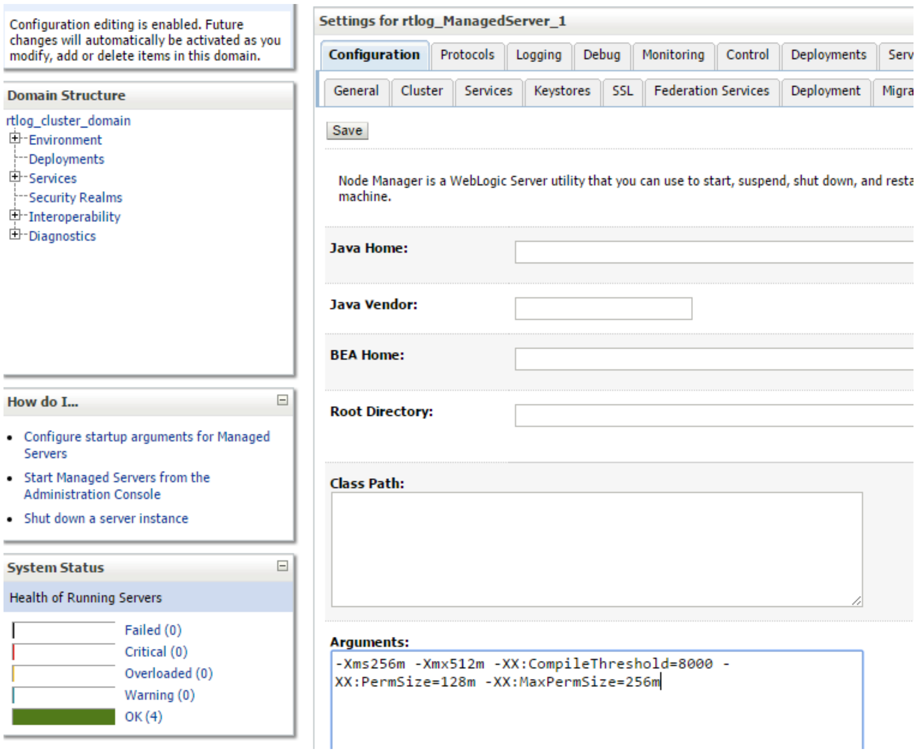
Status: Reachable

Version: 12.1.3

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

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

Figure 6–25 Administration Console Configuration Page



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

Java Home:

Java Vendor:

BEA Home:

Root Directory:

Class Path:

Arguments:

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

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

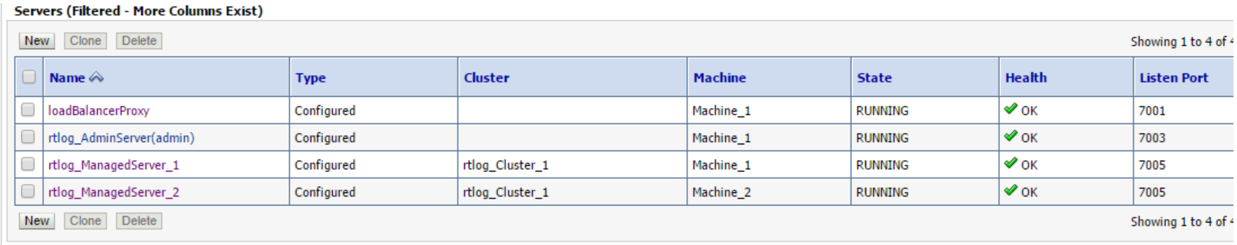
-Drtloggen.config.dir=C:/*<rtlog-gen-config_1>*/

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

---

9. Start all the managed servers including the Oracle proxy. Figure 6–26 shows an example of the list of managed servers.

Figure 6–26 Administration Console List of Servers



□	Name	Type	Cluster	Machine	State	Health	Listen Port
□	loadBalancerProxy	Configured		Machine_1	RUNNING	✓ OK	7001
□	rtlog_AdminServer(admin)	Configured		Machine_1	RUNNING	✓ OK	7003
□	rtlog_ManagedServer_1	Configured	rtlog_Cluster_1	Machine_1	RUNNING	✓ OK	7005
□	rtlog_ManagedServer_2	Configured	rtlog_Cluster_1	Machine_2	RUNNING	✓ OK	7005

Deployment of the RTLog Generator Application on a Cluster

To deploy the application:

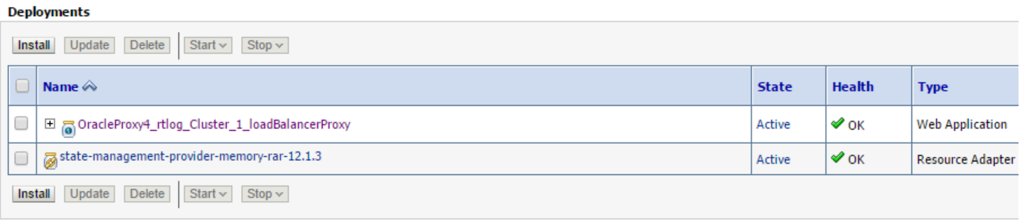
1. Oracle proxy creates a web application by creating the web.xml and weblogic.xml files which can be found in the following directory:

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

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

2. Navigate to the Administration Console home page and click Deployments in the left navigation menu. Figure 6–27 shows an example of the page before deploying the RTLog Generator application.

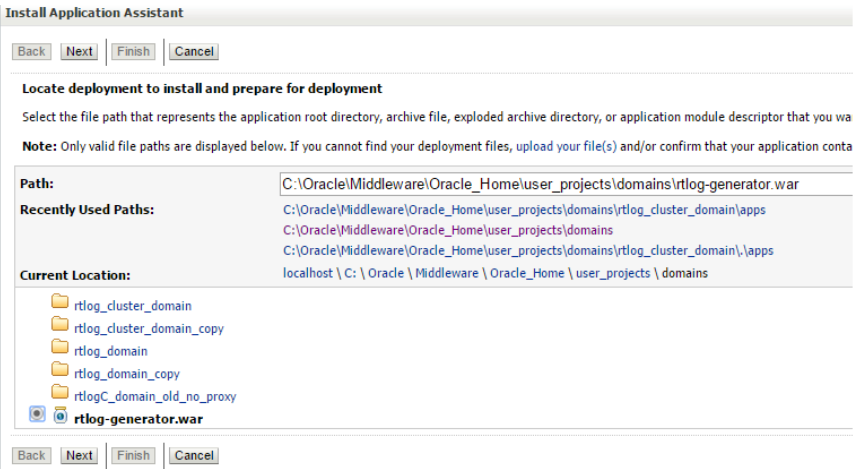
Figure 6–27 Administration Console Deployments Page



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

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

Figure 6–28 Administration Console Install Application Assistant Page



Locate deployment to install and prepare for deployment

Select the file path that represents the application root directory, archive file, exploded archive directory, or application module descriptor that you wa

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

Path: C:\Oracle\Middleware\Oracle_Home\user_projects\domains\rtlog-generator.war

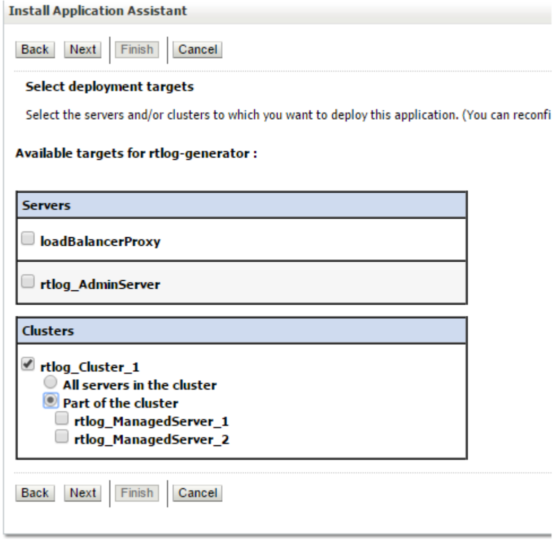
Recently Used Paths:
C:\Oracle\Middleware\Oracle_Home\user_projects\domains\rtlog_cluster_domain\apps
C:\Oracle\Middleware\Oracle_Home\user_projects\domains
C:\Oracle\Middleware\Oracle_Home\user_projects\domains\rtlog_cluster_domain\.\apps

Current Location: localhost \ C: \ Oracle \ Middleware \ Oracle_Home \ user_projects \ domains

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

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

Figure 6–29 Install Application Assistant Select Deployment Targets Page



Select deployment targets

Select the servers and/or clusters to which you want to deploy this application. (You can reconfi

Available targets for rtlog-generator :

Servers

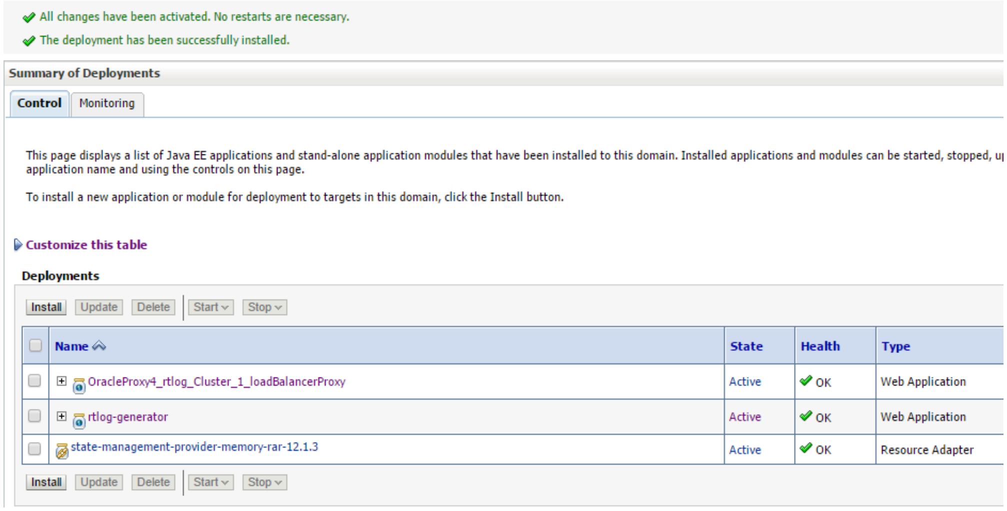
- □ loadBalancerProxy
- □ rtlog_AdminServer

Clusters

- ☑ rtlog_Cluster_1
  - ○ All servers in the cluster
  - ● Part of the cluster
    - □ rtlog_ManagedServer_1
    - □ rtlog_ManagedServer_2

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

Figure 6–30 Summary of Deployments Page



✓ All changes have been activated. No restarts are necessary.

✓ The deployment has been successfully installed.

Summary of Deployments

This page displays a list of Java EE applications and stand-alone application modules that have been installed to this domain. Installed applications and modules can be started, stopped, u application name and using the controls on this page.

To install a new application or module for deployment to targets in this domain, click the Install button.

Deployments

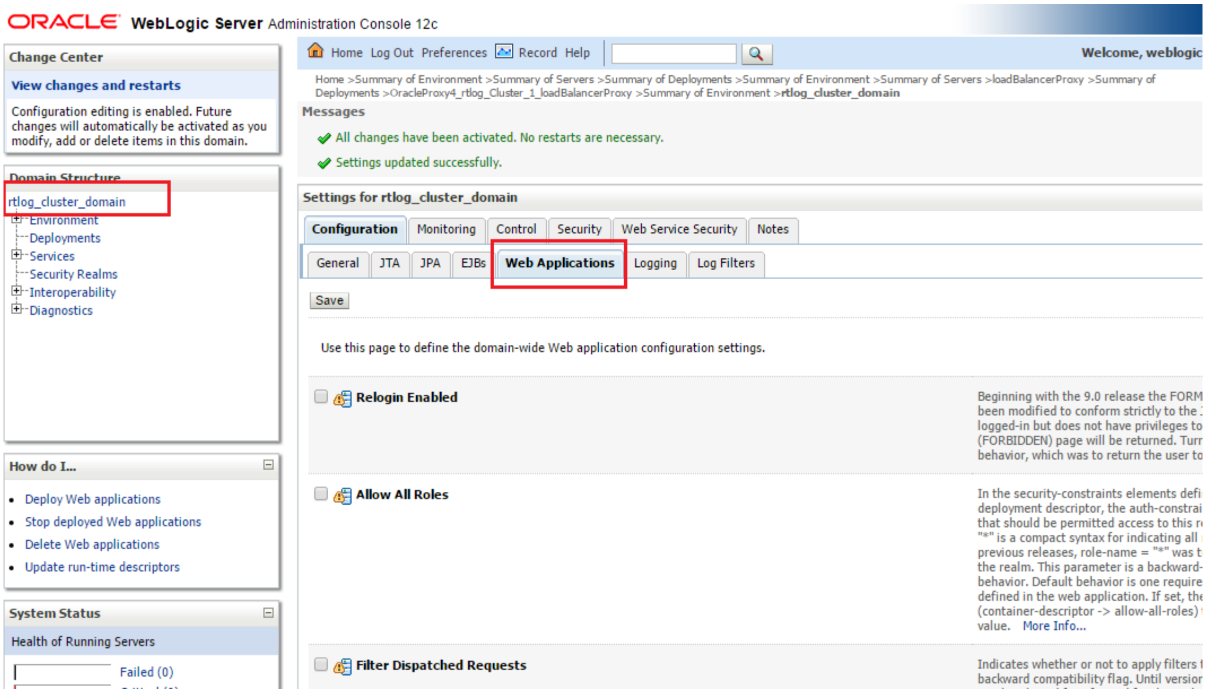
□	Name	State	Health	Type
□	⊞ OracleProxy4_rtlog_Cluster_1_loadBalancerProxy	Active	✓ OK	Web Application
□	⊞ rtlog-generator	Active	✓ OK	Web Application
□	state-management-provider-memory-rar-12.1.3	Active	✓ OK	Resource Adapter

5. To enable container and transport level security, see "Security Configuration".

6. To enable the WebLogic Plugin Enabled parameter from the cluster domain:

   a. Click the <*rtlog_cluster_domain*> link in the left navigation menu. Navigate to the Web Application tab.

Figure 6–31 Administration Console Settings Page



Home >Summary of Environment >Summary of Servers >Summary of Deployments >Summary of Environment >Summary of Servers >loadBalancerProxy >Summary of Deployments >OracleProxy4_rtlog_Cluster_1_loadBalancerProxy >Summary of Environment >**rtlog_cluster_domain**

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

Messages

✓ All changes have been activated. No restarts are necessary.

✓ Settings updated successfully.

Settings for rtlog_cluster_domain

Use this page to define the domain-wide Web application configuration settings.

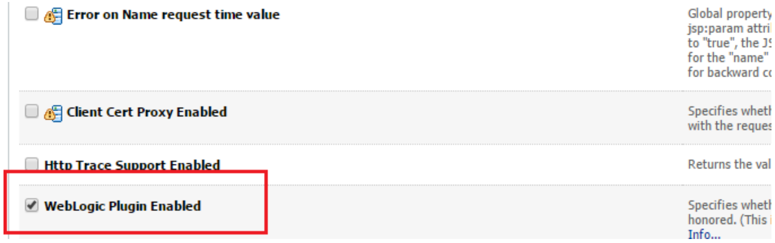
□ **Relogin Enabled** — Beginning with the 9.0 release the FORM been modified to conform strictly to the logged-in but does not have privileges to (FORBIDDEN) page will be returned. Turr behavior, which was to return the user to

□ **Allow All Roles** — In the security-constraints elements defi deployment descriptor, the auth-constrai that should be permitted access to this r "*" is a compact syntax for indicating all previous releases, role-name = "*" was t the realm. This parameter is a backward- behavior. Default behavior is one require defined in the web application. If set, the (container-descriptor -> allow-all-roles) value. More Info...

□ **Filter Dispatched Requests** — Indicates whether or not to apply filters backward compatibility flag. Until versior

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

Figure 6–32 WebLogic Plugin Enabled Parameter



□ **Error on Name request time value** — Global property jsp:param attri to "true", the JS for the "name" for backward co

□ **Client Cert Proxy Enabled** — Specifies wheth with the reques

□ **Http Trace Support Enabled** — Returns the val

☑ **WebLogic Plugin Enabled** — Specifies wheth honored. (This Info...

Security Configuration

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

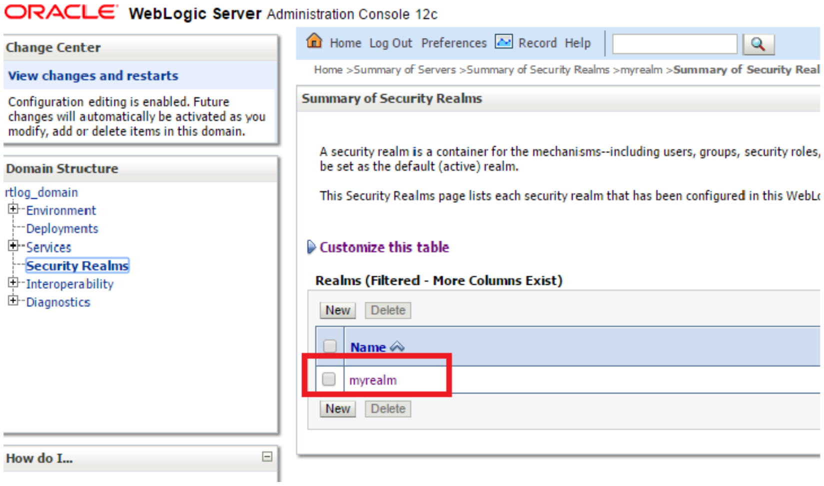
- Container level security: Basic HTTP authentication by setting up the security realm in WebLogic. To configure this security, see "Container Level Security".
- Transport level security: SOAP requests are sent over the secured protocol (HTTPS) by configuring the keystore/truststore in the WebLogic domain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

Container Level Security

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

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

Figure 6–33 Administration Console Summary of Security Realms Page



Home >Summary of Servers >Summary of Security Realms >myrealm >Summary of Security Real

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

Summary of Security Realms

A security realm is a container for the mechanisms--including users, groups, security roles, be set as the default (active) realm.

This Security Realms page lists each security realm that has been configured in this WebL

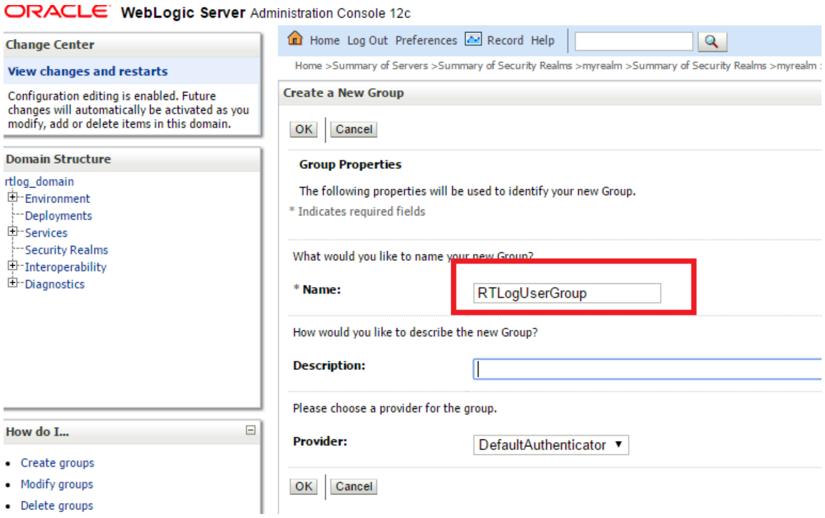
Realms (Filtered - More Columns Exist)

Name
myrealm

3. In the list of realms on the Summary of Security Realms page, select myrealm.

4. Select Users and Groups and then the Groups tab. To create a new group, click **New**. Enter a group name, for example RTLogUserGroup, and click **OK**.

Figure 6–34 Create a New Group Page



Home >Summary of Servers >Summary of Security Realms >myrealm >Summary of Security Realms >myrealm :

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

Create a New Group

Group Properties

The following properties will be used to identify your new Group.

* Indicates required fields

What would you like to name your new Group?

* Name: RTLogUserGroup

How would you like to describe the new Group?

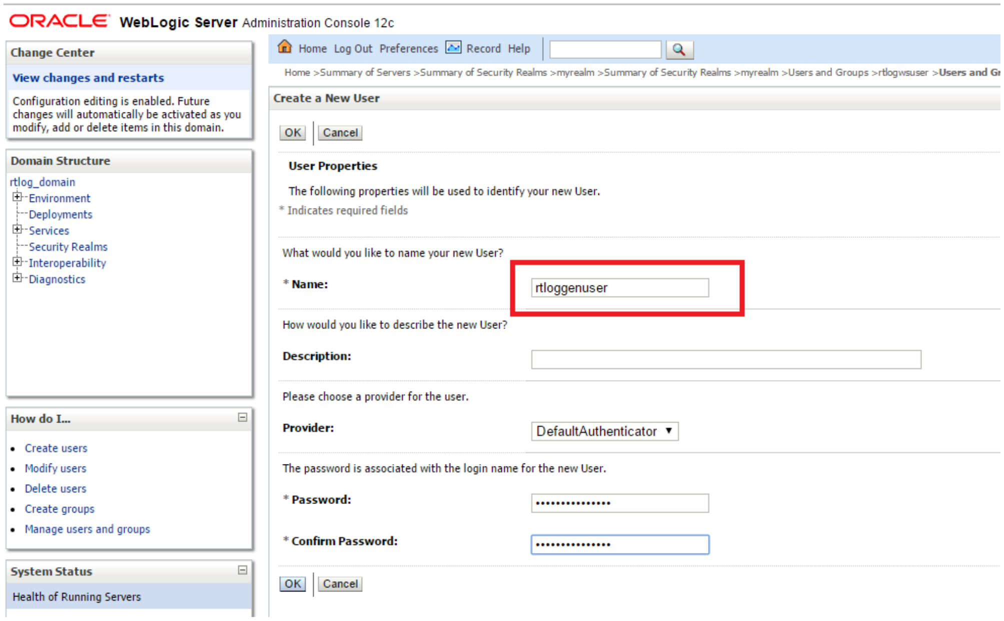
Description:

Please choose a provider for the group.

Provider: DefaultAuthenticator ▾

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

Figure 6–35 Create a New User Page



Home >Summary of Servers >Summary of Security Realms >myrealm >Summary of Security Realms >myrealm >Users and Groups >rtlogwsuser >**Users and Gr**

Create a New User

User Properties

The following properties will be used to identify your new User.

* Indicates required fields

What would you like to name your new User?

* **Name:** rtloggenuser

How would you like to describe the new User?

Description:

Please choose a provider for the user.

Provider: DefaultAuthenticator

The password is associated with the login name for the new User.

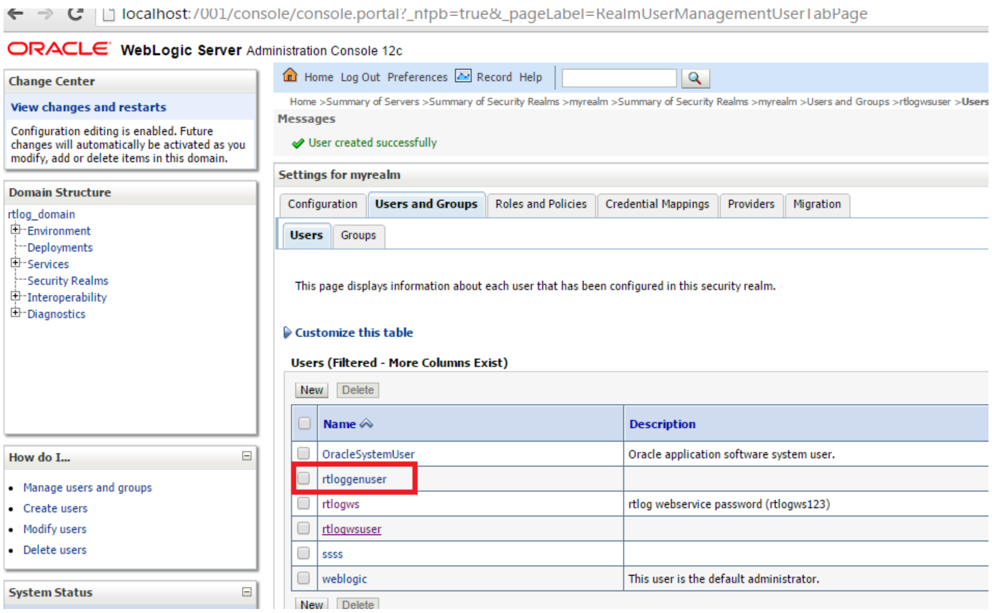
* **Password:** •••••••••••••••

* **Confirm Password:** •••••••••••••••

ORACLE WebLogic Server Administration Console 12c
Home Log Out Preferences Record Help
Change Center
View changes and restarts
Configuration editing is enabled. Future changes will automatically be activated as you modify, add or delete items in this domain.
Domain Structure
rtlog_domain
Environment
Deployments
Services
Security Realms
Interoperability
Diagnostics
How do I...
Create users
Modify users
Delete users
Create groups
Manage users and groups
System Status
Health of Running Servers
OK
Cancel

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

Figure 6–36 Users Page



localhost:7001/console/console.portal?_nfpb=true&_pageLabel=RealmUserManagementUserTabPage

Home >Summary of Servers >Summary of Security Realms >myrealm >Summary of Security Realms >myrealm >Users and Groups >rtlogwsuser >**Users**

Messages

✓ User created successfully

Settings for myrealm

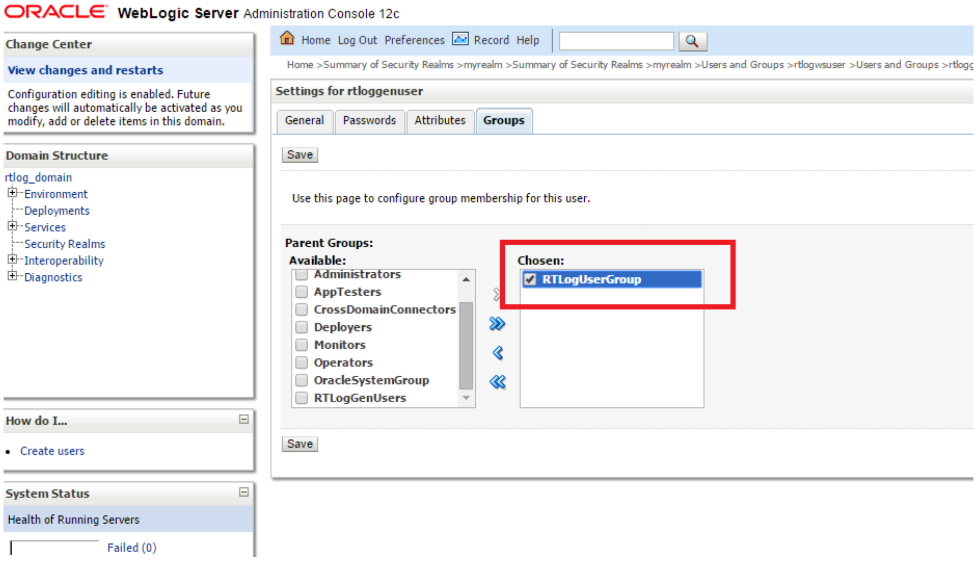
This page displays information about each user that has been configured in this security realm.

Users (Filtered - More Columns Exist)

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

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

Figure 6–37 User Settings Page



ORACLE WebLogic Server Administration Console 12c

Change Center

View changes and restarts

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

Domain Structure

rtlog_domain
- Environment
- Deployments
- Services
- Security Realms
- Interoperability
- Diagnostics

How do I...

- Create users

System Status

Health of Running Servers

Failed (0)

Home Log Out Preferences Record Help

Home >Summary of Security Realms >myrealm >Summary of Security Realms >myrealm >Users and Groups >rtlogwsuser >Users and Groups >rtlogg

Settings for rtloggenuser

General | Passwords | Attributes | Groups

Save

Use this page to configure group membership for this user.

Parent Groups:

Available:

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

Chosen:

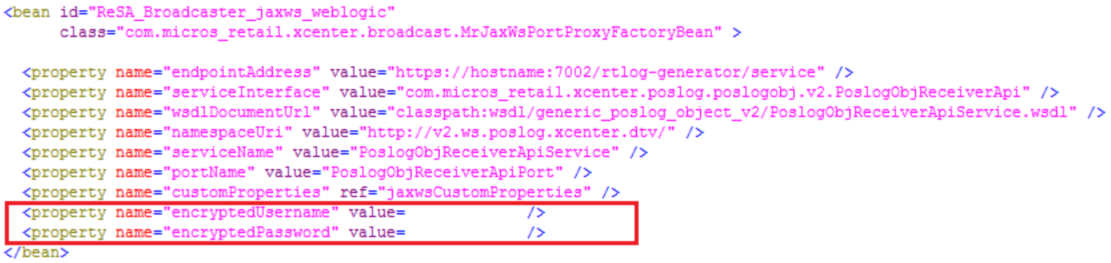
☑ RTLogUserGroup

Save

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

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

Figure 6–38 Example of MrJaxWsPortProxyFactoryBean Update



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

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

Transport Level Security

To configure transport level security:

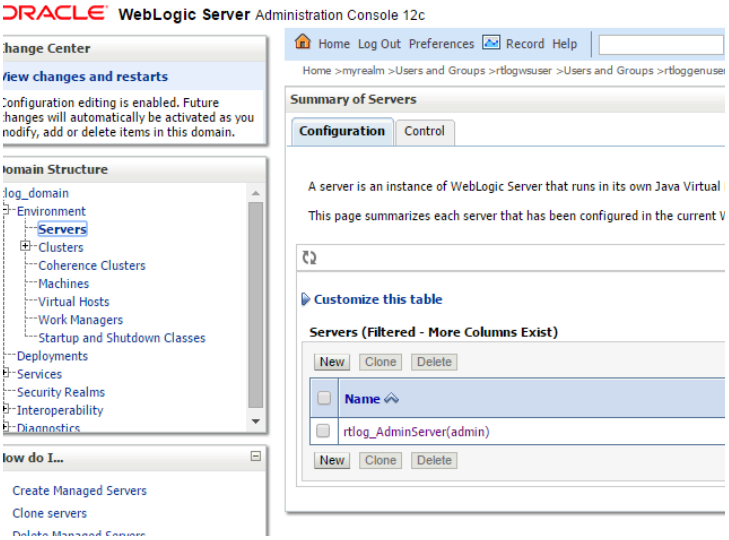
1. Create keystore.jks using a keytool utility. For information on keytool utilities, see the *Oracle Retail Xstore Point of Service Implementation Guide*.

2. Export the public certificate into a truststore.jks file. These files are needed to configure the custom key and trust store for Step 3.

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

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

Figure 6–39 Administration Console Servers Page



ORACLE WebLogic Server Administration Console 12c

hange Center

iew changes and restarts

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

omain Structure

log_domain
- Environment
  - Servers
  - Clusters
  - Coherence Clusters
  - Machines
  - Virtual Hosts
  - Work Managers
  - Startup and Shutdown Classes
- Deployments
- Services
- Security Realms
- Interoperability
- Diagnostics

ow do I...

Create Managed Servers

Clone servers

Home Log Out Preferences Record Help

Home >myrealm >Users and Groups >rtlogwsuser >Users and Groups >rtloggenuse

Summary of Servers

Configuration | Control

A server is an instance of WebLogic Server that runs in its own Java Virtual

This page summarizes each server that has been configured in the current V

Customize this table

Servers (Filtered - More Columns Exist)

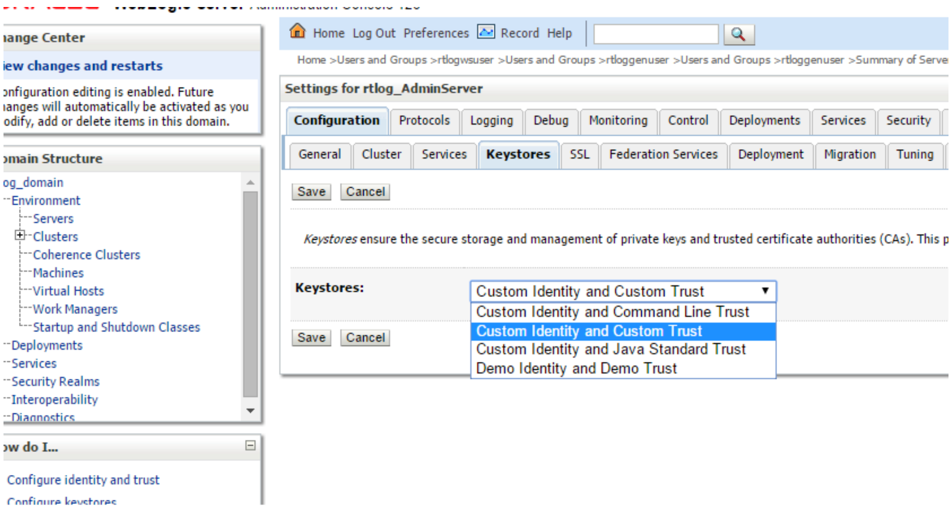
New Clone Delete

□	Name
□	rtlog_AdminServer(admin)

New Clone Delete

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

Figure 6–40 Keystores Settings



ange Center

ew changes and restarts

onfiguration editing is enabled. Future anges will automatically be activated as you odify, add or delete items in this domain.

omain Structure

og_domain
- Environment
  - Servers
  - Clusters
  - Coherence Clusters
  - Machines
  - Virtual Hosts
  - Work Managers
  - Startup and Shutdown Classes
- Deployments
- Services
- Security Realms
- Interoperability
- Diagnostics

ow do I...

Configure identity and trust

Configure keystores

Home Log Out Preferences Record Help

Home >Users and Groups >rtlogwsuser >Users and Groups >rtloggenuser >Users and Groups >rtloggenuser >Summary of Serve

Settings for rtlog_AdminServer

Configuration | Protocols | Logging | Debug | Monitoring | Control | Deployments | Services | Security

General | Cluster | Services | Keystores | SSL | Federation Services | Deployment | Migration | Tuning

Save Cancel

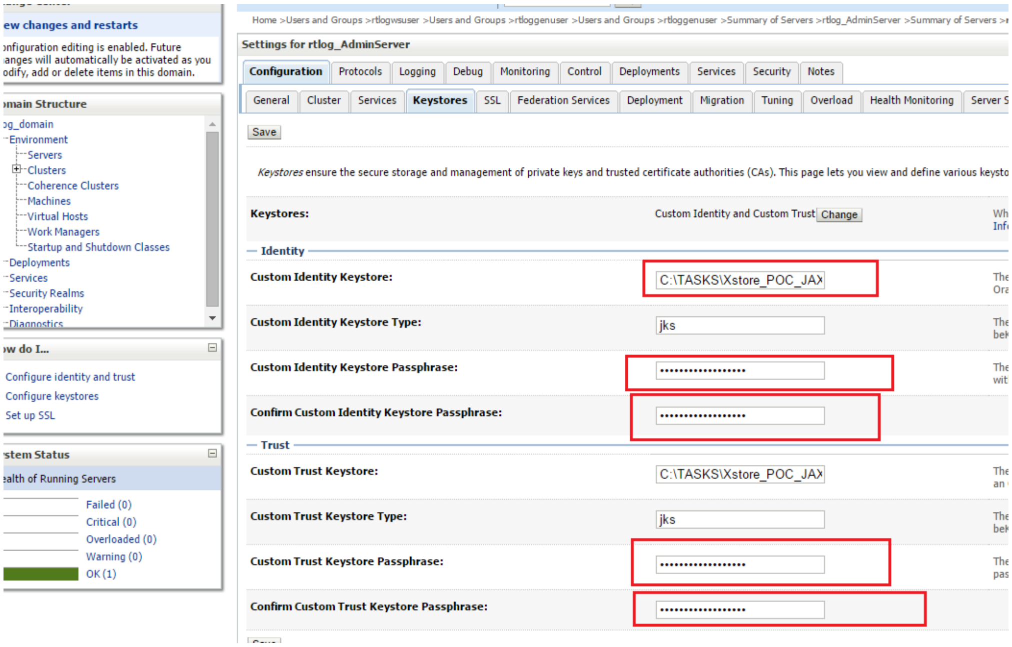
Keystores ensure the secure storage and management of private keys and trusted certificate authorities (CAs). This p

Keystores: Custom Identity and Custom Trust
- Custom Identity and Command Line Trust
- Custom Identity and Custom Trust
- Custom Identity and Java Standard Trust
- Demo Identity and Demo Trust

Save Cancel

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

Figure 6–41 Settings for the Administration Server



Settings for rtlog_AdminServer

Keystores ensure the secure storage and management of private keys and trusted certificate authorities (CAs). This page lets you view and define various keysto

Keystores: Custom Identity and Custom Trust

Identity

Custom Identity Keystore: C:\TASKS\Xstore_POC_JAX

Custom Identity Keystore Type: jks

Custom Identity Keystore Passphrase: ••••••••••••••••••

Confirm Custom Identity Keystore Passphrase: ••••••••••••••••••

Trust

Custom Trust Keystore: C:\TASKS\Xstore_POC_JAX

Custom Trust Keystore Type: jks

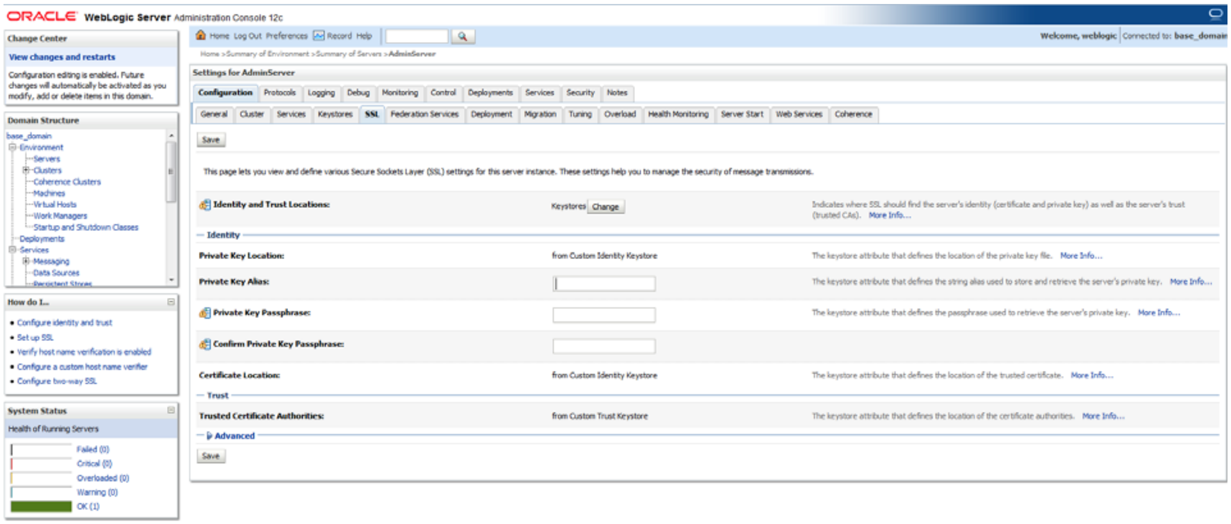
Custom Trust Keystore Passphrase: ••••••••••••••••••

Confirm Custom Trust Keystore Passphrase: ••••••••••••••••••

6. Enter the path to keystore.jks, including the file name, and enter the custom Identity Keystore passphrase you created for the keystore. Repeat this for trustore.jks, but enter the appropriate passphrase for the truststore. For an example, see Figure 6–41.

7. Switch to the SSL tab. Enter the alias name and private keyphrase as created during the certificate generation. To save the changes, click **Save**.

Figure 6–42 Save Settings for Administration Server



Settings for AdminServer

This page lets you view and define various Secure Sockets Layer (SSL) settings for this server instance. These settings help you to manage the security of message transmissions.

Identity and Trust Locations: Keystores — Indicates where SSL should find the server's identity (certificate and private key) as well as the server's trust (trusted CAs). More Info...

Identity

Private Key Location: from Custom Identity Keystore — The keystore attribute that defines the location of the private key file. More Info...

Private Key Alias: — The keystore attribute that defines the string alias used to store and retrieve the server's private key. More Info...

Private Key Passphrase: — The keystore attribute that defines the passphrase used to retrieve the server's private key. More Info...

Confirm Private Key Passphrase:

Certificate Location: from Custom Identity Keystore — The keystore attribute that defines the location of the trusted certificate. More Info...

Trust

Trusted Certificate Authorities: from Custom Trust Keystore — The keystore attribute that defines the location of the certificate authorities. More Info...

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

Complete the Security Configuration

Test both the container and transport level security using SOAPUI.

To set up the unlimited strength JCE files:

1. Download and install the correct version of the unlimited strength JCE files. For more information, see the *Oracle Retail Xstore Point of Service Implementation Guide*.
2. Configure WebLogic 12c with the Xstore suite of product's supported cipher suites. To configure it, update the <*domain*>\<*domain_name*>\config\config.xml file and add the following inside the ssl block:

   ```
   <ciphersuite>TLS_RSA_WITH_AES_256_CBC_SHA</ciphersuite>
   <ciphersuite>TLS_RSA_WITH_AES_256_CBC_SHA</ciphersuite
   <ciphersuite>TLS_RSA_WITH_AES_256_CBC_SHA256</ciphersuite>
   <ciphersuite>TLS_ECDHE_RSA_WITH_AES_256_CBC_SHA</ciphersuite>
   <ciphersuite>TLS_ECDHE_ECDSA_WITH_AES_256_CBC_SHA</ciphersuite>
   <ciphersuite>TLS_ECDHE_RSA_WITH_AES_256_CBC_SHA384</ciphersuite>
   <ciphersuite>TLS_ECDHE_ECDSA_WITH_AES_256_CBC_SHA384</ciphersuite>
   ```

3. Disable the schema validation in WebLogic by passing the JVM argument in the WebLogic startup script:

   -Dweblogic.configuration.schemaValidationEnabled=false

4. Xstore Office's RTLog Generator broadcaster end point should be configured to use the secured (HTTPS) URL for configuring the container level security section:

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

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

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

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

   - Configure endpointAddress of the ReSA_Broadcaster_jaxws bean.

7 RTLog Generator Cloud

This chapter describes the RTLog Generator on cloud.

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 distributed as a web archive along with a configuration zip file ready to be deployed on an Oracle WebLogic 12c server. It is usually deployed alongside the other Xstore office cloud applications.

The RTLog Generator on cloud can be integrated with a ReSA application deployed either on-premises or on cloud.

Configuration

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

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

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

The RTLog Generator application relies heavily on the XML-based mapping specified in the `RTLogMappingConfig.xml` file to provide 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.

The following elements of the mapping XML file can be customized:

- `FieldMapperThenValueMapping`:

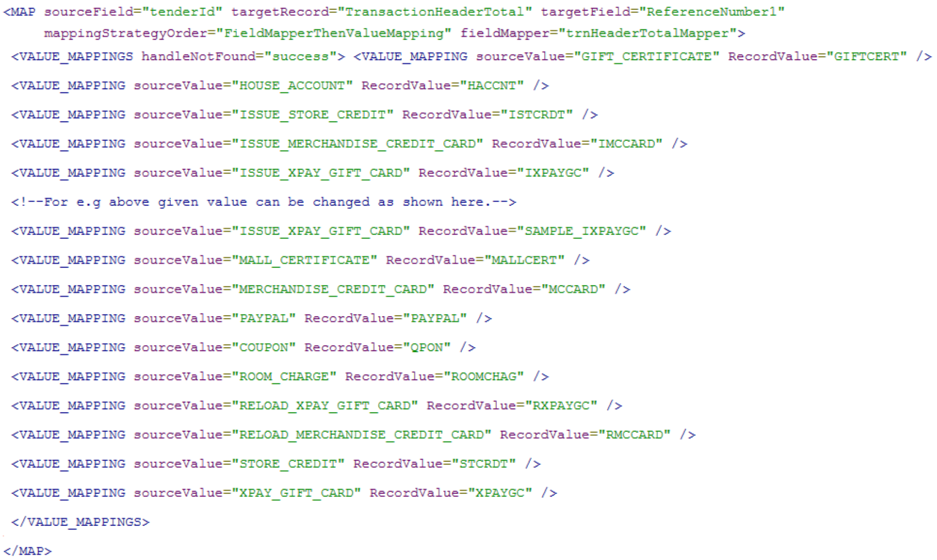
  The `RecordValue` attribute values as shown in the following example can be changed:

  ```
  <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>
```

Figure 7–1 RTLogMappingConfig.xml Field Mapper Example 1



```
<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>
```

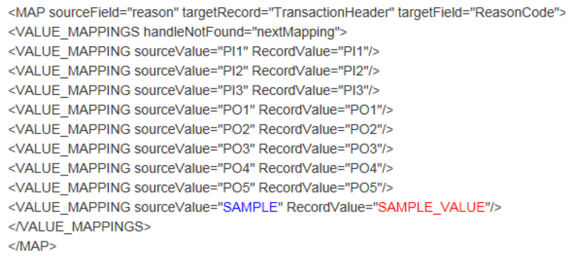
- No `mappingStrategyOrder` and `fieldMapper` attributes are defined:

  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>
```

Figure 7–2 RTLogMappingConfig.xml Field Mapper Example 2



```
<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>
```

Integration

This section describes the RTlog Generator Cloud integration.

Updating Mapping Configuration

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

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

A new property `configUploadDir` is added to `rtlogconfig.properties`.

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. This directory specifies the upload directory to host the customized mapping file. In a cluster environment with multiple RTLogGen nodes, all nodes must be configured to point to the same config upload directory on shared file system. This is to ensure that once a mapping file is uploaded, it is visible to all the nodes.

```
####
#### (uncomment and configure this if and only if you are setting up a cluster of
RTLogGen nodes on cloud)
####
####configUploadDir =  /somewhere/rtlogconfig_upload
```

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 hasn't 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/RTLogMappingConf
ig"  > 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/RTLogMappingConf
ig"
```

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

Retrieving Published RTLog Files

RTLog Generator Cloud application's ability to provide a mechanism to retrieve the published RTLog files varies depending on the type of ReSA application that it is integrated with.

- For ReSA on cloud, SFTP process is used to transfer the files
- For ReSA on-premises, REST service provides the way to download the files

ReSA on Cloud:

- SFTP credentials to connect to the ReSA application's SFTP directory on cloud are made available to the RTLog Generator Cloud deployment team.
- SFTP connectivity utilizes public/private key based authentication.
- Cloud Application Management is responsible for SFTP credentials rotation.

ReSA on-premises:

- RTLog Generator provides a REST service to download the published RTLog files in compressed format (zip files).
- OAuth2 token is required for communicating with the REST web service.
- Returns a zip file if one available on the published RTLog files directory.
- Content type of the response will be application/octet-stream
- Content-disposition response header will contain the name of the attached zip file
- Returns HTTP 204 code when no zip files are available for download.

The following example shows how to retrieve the published RTLog files using curl.

Example 7–3 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. The ReSA application is required to do the same in order to communicate with the RTLog Generator application via REST web services.

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 ReSA application).
- 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 7–4 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 7–5 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 7–6 Provide Oauth2 Token - Provide OAuth2 Token for REST Services

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

A

Appendix: POSLog to RTLog Mapping Details

The mapping from the POSLog format to the RTLog format is defined in the Xstore configuration file RTLogMappingConfig.xml. This appendix provides details on the following mappings:

- Transaction Type Mapping
- Tender Type Mapping
- Total Tender ID Mapping
- Item Type Mapping
- Reason Code Mapping
- Item Status/Sales Type Mapping

Transaction Type Mapping

- The ReSA transaction type values are defined in code_type TRAT.
- The ReSA sub-transaction type values are defined in code_type TRAS.

Table A–1 describes the Xstore to ReSA transaction type mapping.

Table A–1 Transaction Type Mapping

Xstore Transaction Type	ReSA Transaction Type TRAT	ReSA Sub-Transaction Type TRAS	Description
ACCOUNT_LOOKUP	OTHER	OTHER	ACCOUNT_LOOKUP transactions are passed from Xstore to ReSA for full visibility audit, but not otherwise implemented in ReSA.
BALANCE_INQUIRY	OTHER	OTHER	BALANCE_INQUIRY transactions are passed from Xstore to ReSA for full visibility audit, but not otherwise implemented in ReSA.
CREDIT_ APPLICATION	OTHER	OTHER	CREDIT_APPLICATION transactions are passed from Xstore to ReSA for full visibility audit, but not otherwise implemented in ReSA.
ESCROW	OTHER	OTHER	ESCROW transactions are passed from Xstore to ReSA for full visibility audit, but not otherwise implemented in ReSA.

Table A–1 (Cont.) Transaction Type Mapping

Xstore Transaction Type	ReSA Transaction Type TRAT	ReSA Sub-Transaction Type TRAS	Description
EXCHANGE_RATE	OTHER	OTHER	EXCHANGE_RATE transactions are passed from Xstore to ReSA for full visibility audit, but not otherwise implemented in ReSA.
GNRIC	OTHER	OTHER	GNRIC transactions are passed from Xstore to ReSA for full visibility audit, but not otherwise implemented in ReSA.
INVENTORY_ CONTROL	OTHER	OTHER	INVENTORY_CONTROL transactions are mapped from Xstore to ReSA for full visibility audit, but not otherwise implemented in ReSA. Xstore should be configured so that inventory control transactions are not generated, and therefore not sent to ReSA.
INVENTORY_ SUMMARY_COUNT	OTHER	OTHER	INVENTORY_SUMMARY_COUNT transactions are mapped from Xstore to ReSA for full visibility audit, but not otherwise implemented in ReSA. Xstore should be configured so that inventory summary count transactions are not generated, and therefore not sent to ReSA.
MOVEMENT_ PENDING	OTHER	OTHER	MOVEMENT_PENDING transactions are mapped from Xstore to ReSA for full visibility audit, but not otherwise implemented in ReSA. Xstore should be configured so that inventory summary count transactions are not generated, and therefore not sent to ReSA.
NO_SALE	NOSALE	NOSALE	NA
POST_VOID	PVOID	VOID	NA
RETAIL_SALE (can be mapped to multiple ReSA transaction types depending on other conditions)	SALE	SALE	Regular transaction.
	NOSALE	SUSPND	Suspend transaction.
	VOID	CANCEL	Cancel transaction.
	VOID	CANCEL	Cancel orphaned transaction.
SESSION_CONTROL	OTHER	OTHER	Issue till.
	OTHER	OTHER	Assign till/assign till tender transfer.
	OTHER	OTHER	Attach till.
	OTHER	OTHER	Remove till.
	OTHER	OTHER	Return till.
SYSTEM_CLOSE	CLOSE	CSTORE	Close store.
SYSTEM_OPEN	OPEN	OSTORE	Open store.

Table A–1 (Cont.) Transaction Type Mapping

Xstore Transaction Type	ReSA Transaction Type TRAT	ReSA Sub-Transaction Type TRAS	Description
TENDER_CONTROL (can be mapped to multiple ReSA 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.

Tender Type Mapping

- The ReSA tender type groups are defined in code_type TENT.

- The ReSA tenders are defined in the seeded data table POS_TENDER_TYPE_HEAD.

Table A–2 describes the Xstore to ReSA transaction tender type mapping.

Table A–2 Tender Type Mapping

Xstore		Xstore POS Log Tender Group Type		ReSA RTLog	
TenderTypeCode	**TenderTypeID**	**Tender Type**	**Tender ID**	**TenderTypeGroup**	**TenderTypeID**
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.

Table A–2 (Cont.) Tender Type Mapping

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

Total Tender ID Mapping

Table A–3 describes the ReSA mapping for the total ID record in the transaction header.

Table A–3 Total Tender ID Mapping

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

Item Type Mapping

ReSA tender type values are defined in code SAIT and used in the following:

- RTLOG TITEM record, item type field
- Sa_tran_item.item_type

Table A–4 describes the Xstore item type mapping.

Table A–4 Item Type Mapping

Xstore Item Type	ReSA Item Type	Description
Alteration	NMITEM	NA
Deposit	NMITEM	NA
dtv:GiftCertificate	GCN	Gift Card and Gift Certificate

Table A–4 (Cont.) Item Type Mapping

Xstore Item Type	ReSA Item Type	Description
dtv:NonMerchandise	NMITEM	NA
dtv:Payment	NMITEM	NA
Fee	NMITEM	NA
ItemCollection	ITEM	NA
Service	NMITEM	NA
Stock	ITEM	NA
Warranty	NMITEM	NA

Reason Code Mapping

Xstore has a single set of reason codes, used both for reason codes, price override codes, and other modifications. ReSA separates these concepts into individual sets used in different RTLog fields and saved to different database table/fields. Because reason codes can be mixed coming out of Xstore, ReSA has mapped some code values to multiple code types to avoid the possibility of errors.

ReSA Reason Codes

ReSA reason codes:

- Code type REAC

> **Note:** ReSA supports a number of other transaction level reason codes. Only reason codes related to Xstore integration are listed here.

- SA_TRAN_HEAD.REASON_CODE
- Used for further information on a number of transaction types.
- Mapped to Xstore miscellaneous reason codes.

Table A–5 describes the reason code mapping.

Table A–5 ReSA Reason Codes

Xstore Reason Code	ReSA 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

Table A–5 (Cont.) ReSA Reason Codes

Xstore Reason Code	ReSA Reason Code	Description
PAID_IN	PI1	Change from Paid Out
PAID_IN	PI2	Found Money
PAID_IN	PI3	Drawer Loan 1
PAID_IN	TENDEX	Tender exchange
PAID_OUT	PO1	Stocks
PAID_OUT	PO2	Delivery
PAID_OUT	PO3	Postage
PAID_OUT	PO4	Contractor Services
PAID_OUT	PO5	Store Incentives

ReSA Return Reason Codes

ReSA return reason codes:

- Code type = SARR
- SA_TRAN_ITEM. RETURN_REASON_CODE

Table A–6 describes the return reason code mapping.

Table A–6 ReSA Return Reason Codes

Xstore Reason Code	ReSA 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

ReSA Discount Reason Codes

ReSA discount reason codes:

- Code type SADT

Note: ReSA supports a number of other discount types. Only discount types related to Xstore integration are listed here.

- SA_TRAN_DISC.DISC_TYPE

Table A–7 describes the discount reason code mapping.

Table A–7 ReSA Discount Reason Codes

Xstore Reason Code	ReSA Reason Code	Description
DC1	S	Incorrect Label
DC2	MS	Manager Discretion
DC3	CP	Price Guarantee
DC4	D	Damage Adjustment
NEW_PRICE_RULE	NEWPRC	New Price Rule
DOCUMENT	DOC	Document
MANUFACTURER_ COUPON	MCOUP	Manufacturer Coupon
REFUND_PRORATION	REFUND	Refund Proration
CALCULATED_ WARRANTY_PRICE	CALWAR	Warranty Price

ReSA Item Price Override Reason Codes

ReSA item price override reason codes:

- Code type ORRC
- SA_TRAN_ITEM.OVERRIDE_REASON

Table A–8 describes the item price override reason code mapping.

Table A–8 ReSA Item Price Override Reason Codes

Xstore Reason Code	ReSA 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/Sales Type Mapping

ReSA item status:

- Code type SASI

- SA_TRAN_ITEM.ITEM_STATUS

Valid values for the ReSA item status are shown in the following table:

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

ReSA sales type:

- Code type SASY
- SA_TRAN_ITEM.SALES_TYPE

Valid values for the ReSA sales type are shown in the following table:

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

Table A–9 describes the item status and sales type mapping.

Table A–9 ReSA Item Status/Sales Type Mapping

Xstore Item	Xstore Action	ReSA Item Status	ReSA 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

Table A–9 (Cont.) ReSA Item Status/Sales Type Mapping

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

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

RMS Diff Group Detail

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

Table B–1 RMS Diff Group Detail Mapping

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

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

	RMS				Xstore				
	Field Name	**Description**	**Field Type**	**Max Width**	**Table**	**Column**	**Data Type**	**Default Value**	**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		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
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
4	Display Seq	Optional sequence to describe the order in which diffs within the diff group should be displayed in user interfaces. Data is optional in this field.	Character	4	rms_diff_group_detail	DISPLAY_SEQ	NUMBER(4)		NA
	NA	NA	NA	NA	rms_diff_group_detail	CREATE_DATE	TIMESTAMP(6)	now()	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

	RMS				Xstore				
	Field Name	**Description**	**Field Type**	**Max Width**	**Table**	**Column**	**Data Type**	**Default Value**	**Comment**
0	Family	Value is always DIFFGRPHDR. Data will always be present in this field.	Character	20	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		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

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

	RMS				Xstore				
	Field Name	Description	Field Type	Max Width	Table	Column	Data Type	Default Value	Comment
3	DiffGroupDesc	Description of the differentiator group (for example: Men's Shirt Sizes, Women's Shoe Sizes, Girls Dress Sizes, Shower Gel Scents, Yogurt Flavors, and so on). Data will always be present in this field in the DIFFGRPHDRCRE and DIFFGRPHDRMOD records.	Character	120	rms_diff_group_head	DIFF_GROUP_DESC	VARCHAR2(120 CHAR)		NA
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
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		Ignored. It is already captured in the diff id tab.
	NA	NA	NA	NA	rms_diff_group_head	CREATE_DATE	TIMESTAMP(6)		NA
	NA	NA	NA	NA	rms_diff_group_head	UPDATE_DATE	TIMESTAMP(6)		NA

RMS Diffs

Table B–3 describes the RMS Diffs mapping.

Table B–3 RMS Diffs Mapping

	RMS				Xstore				
	Field Name	**Description**	**Field Type**	**Max Width**	**Table**	**Column**	**Data Type**	**Default Value**	**Comment**
0	Family	Value is always DIFFS. Data will always be present in this field.	Character	20	NA	NA	NA		Ignored.
1	Type	Value is always FULL when the program is run in full mode. If the program is run in delta mode, the type will vary based on the type of change (CRE, MOD, DEL). Valid values are: DIFFCRE - Creation of diff information. DIFFMOD - Modification of diff information. DIFFDEL - Deletion of diff information. Data will always be present in this field.	Character	15	NA	NA	NA		Determines action.
2	DiffId	Contains the unique ID of the diff. Data will always be present in this field.	Character	10	rms_diff_ids	DIFF_ID	VARCHAR2(10 CHAR)		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
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)		Also used to populate rms_diff_group_head for the default Diff Group.

Table B–3 (Cont.) RMS Diffs Mapping

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

Table B–3 (Cont.) RMS Diffs Mapping

	RMS				Xstore				
	Field Name	Description	Field Type	Max Width	Table	Column	Data Type	Default Value	Comment
	NA	NA	NA	NA	rms_diff_ids	CREATE_DATE	TIMESTAMP(6)	now()	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

	RMS				Xstore				
	Field Name	Description	Field Type	Max Width	Table	Column	Data Type	Default Value	Comment
0	Family	Value is always ITEMS. Data will always be present in this field.	Character	20	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		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)	"*"	Dimension Value and Dimension Type are only used for Style items.
3	Item	ID of the item. This field will always have data.	Character	25	itm_item, itm_item_dimension_value, itm_item_dimension_type	ITEM_ID, DIMENSION_SYSTEM	VARCHAR(60)		DIMENSION_SYSTEM is populated with ItemID for Style items.

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

	RMS				Xstore				
	Field Name	**Description**	**Field Type**	**Max Width**	**Table**	**Column**	**Data Type**	**Default Value**	**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)		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		Ignored.

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

	RMS				Xstore				
	Field Name	**Description**	**Field Type**	**Max Width**	**Table**	**Column**	**Data Type**	**Default Value**	**Comment**
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		Ignored.
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		Ignored.

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

	RMS				Xstore				
	Field Name	Description	Field Type	Max Width	Table	Column	Data Type	Default Value	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		Used to distinguish Item from Style, and UPC from Item.

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

	RMS				Xstore				
	Field Name	Description	Field Type	Max Width	Table	Column	Data Type	Default Value	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		Used to distinguish Item from Style, and UPC from Item.

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

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

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

	RMS				Xstore				
	Field Name	Description	Field Type	Max Width	Table	Column	Data Type	Default Value	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		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)		Only used when Diff1Level is provided.

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

	RMS				Xstore				
	Field Name	Description	Field Type	Max Width	Table	Column	Data Type	Default Value	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
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		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

	RMS				Xstore				
	Field Name	Description	Field Type	Max Width	Table	Column	Data Type	Default Value	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)		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

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

	RMS				Xstore				
	Field Name	**Description**	**Field Type**	**Max Width**	**Table**	**Column**	**Data Type**	**Default Value**	**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		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)		Only used when Diff3Level is provided.

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

	RMS				Xstore				
	Field Name	Description	Field Type	Max Width	Table	Column	Data Type	Default Value	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
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		Ignored.

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

	RMS				Xstore				
	Field Name	**Description**	**Field Type**	**Max Width**	**Table**	**Column**	**Data Type**	**Default Value**	**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		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		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)		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

	RMS				Xstore				
	Field Name	**Description**	**Field Type**	**Max Width**	**Table**	**Column**	**Data Type**	**Default Value**	**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		Ignored.
25	Unique Class	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 configure d level)	VARCHA R2(60 CHAR)		C is always appended to the value.

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

	RMS				Xstore				
	Field Name	**Description**	**Field Type**	**Max Width**	**Table**	**Column**	**Data Type**	**Default Value**	**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		Ignored.
27	Unique Subclass	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)		S is always appended to the value.

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

	RMS				Xstore				
	Field Name	Description	Field Type	Max Width	Table	Column	Data Type	Default Value	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		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)		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		Ignored.
31	ShortDescription	Shortened item description. This field will always have data for the ITEMHDRCRE and ITEMHDRMOD records.	Character	120	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		Ignored.

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

	RMS				Xstore				
	Field Name	**Description**	**Field Type**	**Max Width**	**Table**	**Column**	**Data Type**	**Default Value**	**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		If N, then NON_ PHYSICAL item so the record is skipped.

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

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

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

	RMS				Xstore				
	Field Name	Description	Field Type	Max Width	Table	Column	Data Type	Default Value	Comment
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		Ignored.
38	Package Size	Holds the size of the product printed on any packaging (for example, 24 ounces). This field can be used for reporting purposes and to determine same sized and different sized items. Data is optional in this field.	Number	12	NA	NA	NA		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		Ignored.

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

	RMS				Xstore				
	Field Name	**Description**	**Field Type**	**Max Width**	**Table**	**Column**	**Data Type**	**Default Value**	**Comment**
40	StoreOrdMulti	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		Ignored.
41	Forecast Ind	Indicates if sales forecasts will be produced for this item. Valid values are: Y, N. This field will always have data for the ITEMHDRCRE and ITEMHDRMOD records.	Character	1	NA	NA	NA		Ignored.
42	Original Retail	The original retail price of the item per unit. This field is stored in the primary currency of the Merchandising systems. Data is optional in this field.	Number	20	NA	NA	NA		Ignored.
43	OriginalRetailCurrencyCode	The currency code of the OriginalRetail. Data is optional in this field.	Character	3	NA	NA	NA		Ignored.
44	MfgRec Retail	Data is optional in this field.	Number	20	itm_item	LIST_PRICE	NUMBER(17,6)		NA
45	MfgRecRetailCurrencyCode	The currency code of the MfgRecRetail. Data is optional in this field.	Character	3	NA	NA	NA		Ignored.

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

	RMS				Xstore				
	Field Name	**Description**	**Field Type**	**Max Width**	**Table**	**Column**	**Data Type**	**Default Value**	**Comment**
46	RetailLabelType	This field indicates any special label type associated with an item (that is, pre-priced or cents off). This field is used in RMS for reporting purposes only. Data is optional in this field.	Character	6	NA	NA	NA		Ignored.
47	RetailLabelValue	This field represents the value associated with the RetailLabelType. Data is optional in this field.	Number	20	NA	NA	NA		Ignored.
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		Ignored.

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

	RMS				Xstore				
	Field Name	Description	Field Type	Max Width	Table	Column	Data Type	Default Value	Comment
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		Ignored.
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		Ignored.

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

	RMS				Xstore				
	Field Name	Description	Field Type	Max Width	Table	Column	Data Type	Default Value	Comment
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		Ignored.
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		Ignored.

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

	RMS				Xstore				
	Field Name	Description	Field Type	Max Width	Table	Column	Data Type	Default Value	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-13 NDC - 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		Ignored.

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

	RMS				Xstore				
	Field Name	**Description**	**Field Type**	**Max Width**	**Table**	**Column**	**Data Type**	**Default Value**	**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		Ignored.

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

	RMS				Xstore				
	Field Name	Description	Field Type	Max Width	Table	Column	Data Type	Default Value	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		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		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		Ignored.

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

	RMS				Xstore				
	Field Name	**Description**	**Field Type**	**Max Width**	**Table**	**Column**	**Data Type**	**Default Value**	**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		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		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		Ignored.

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

	RMS				Xstore				
	Field Name	Description	Field Type	Max Width	Table	Column	Data Type	Default Value	Comment
61	Default WastePct	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		Ignored.
62	ConstantDimInd	This field will always have data.	Character	1	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		Ignored.
64	Sellable Ind	Indicates if the item may be sold as a unit. If it is Y, the item will have its own unique unit retail. If it is N, the item will not have a unit retail. This field will always have data for the ITEMHDRCRE and ITEMHDRMOD records.	Character	1	NA	NA	NA		Ignored.

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

	RMS				Xstore				
	Field Name	Description	Field Type	Max Width	Table	Column	Data Type	Default Value	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		Ignored.

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

	RMS				Xstore				
	Field Name	Description	Field Type	Max Width	Table	Column	Data Type	Default Value	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		Ignored.

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

	RMS				Xstore				
	Field Name	**Description**	**Field Type**	**Max Width**	**Table**	**Column**	**Data Type**	**Default Value**	**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		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		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		Ignored.

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

	RMS				Xstore				
	Field Name	**Description**	**Field Type**	**Max Width**	**Table**	**Column**	**Data Type**	**Default Value**	**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		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		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)		If Y, then 1.

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

	RMS				Xstore				
	Field Name	**Description**	**Field Type**	**Max Width**	**Table**	**Column**	**Data Type**	**Default Value**	**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		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		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		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		Ignored.

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

	RMS				Xstore				
	Field Name	**Description**	**Field Type**	**Max Width**	**Table**	**Column**	**Data Type**	**Default Value**	**Comment**
77	Deposit ItemTyp e	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		Ignored.
78	Contain erItem	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		Ignored.

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

	RMS				Xstore				
	Field Name	**Description**	**Field Type**	**Max Width**	**Table**	**Column**	**Data Type**	**Default Value**	**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		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		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		Ignored.

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

	RMS				Xstore				
	Field Name	**Description**	**Field Type**	**Max Width**	**Table**	**Column**	**Data Type**	**Default Value**	**Comment**
82	Comments	Holds any free-form comments associated with the item. Data is optional in this field.	Character	2000	NA	NA	NA		Ignored.
	NA	NA	NA	NA	itm_item	MERCH_LEVEL_1	VARCHAR2(60 CHAR)		Populated by looking up the parent of the merch_level_2 value in loc_org_hierarchy.
	NA	NA	NA	NA	itm_item	ITEM_TYPCODE	VARCHAR2(30 CHAR)	STANDARD	NA
	NA	NA	NA	NA	itm_item	DESCRIPTION	VARCHAR2(254 CHAR)		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	itm_item	NAME	VARCHAR2(254 CHAR)		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

	RMS				Xstore				
	Field Name	**Description**	**Field Type**	**Max Width**	**Table**	**Column**	**Data Type**	**Default Value**	**Comment**
	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	itm_item, itm_item_dimension_value, itm_item_dimension_type	CREATE_DATE	TIMESTAMP(6)	now()	Not set when Type =ITEMHDRMOD.
	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	itm_item, itm_item_dimension_value, itm_item_dimension_type	UPDATE_DATE	TIMESTAMP(6)	now()	Not set when Type =FULLHDR.
	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

	RMS				Xstore				
	Field Name	Description	Field Type	Max Width	Table	Column	Data Type	Default Value	Comment
0	Family	Value is always ITEMLOC. Data will always be present in this field.	Character	20	NA	NA	NA		Ignored.
1	Type	Value is always FULLITEMLOC when the program is run in full mode. In delta mode, valid values are: ITEMLOCCRE ITEMLOCMOD ITEMLOCDEL Data will always be present in this field.	Character	15	NA	NA	NA		Determines action.
2	LocTyp e	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	VARCHA R2(30 CHAR)	STORE	If not S, entire record is ignored.
3	Locatio n	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	VARCHA R2(30 CHAR)		NA
4	Item	ID of the item. Data will always be present in this field.	Character	25	itm_item_ options	ITEM_ID	VARCHA R2(60 CHAR)		NA
5	ItemPar ent	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		Ignored, obtained from ItemHdr feed.

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

	RMS				Xstore				
	Field Name	**Description**	**Field Type**	**Max Width**	**Table**	**Column**	**Data Type**	**Default Value**	**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		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		Ignored.

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

	RMS				Xstore				
	Field Name	**Description**	**Field Type**	**Max Width**	**Table**	**Column**	**Data Type**	**Default Value**	**Comment**
8	Selling UnitRetail	Data is optional in this field for some type of items. Data is only included in this field for the ITEMLOCCRE and FULLITEMLOC records. After initial item creation, unit retail and other pricing information should come from RPM. If this program is run in full mode and produces FULLITEMLOC records, a full extract from RPM of pricing should also be performed to update all price records.	Number	20	itm_item_prices	PRICE	NUMBER (17,6)		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		Ignored.
10	Selling UOM	Data is optional in this field for some type of items. Data is only included in this field for the ITEMLOCCRE and FULLITEMLOC records.	Character	4	itm_item_options	UNIT_OF_MEASURE_CODE	VARCHAR2(30 CHAR)		NA
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

	RMS				Xstore				
	Field Name	Description	Field Type	Max Width	Table	Column	Data Type	Default Value	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)		If translated, ItemDescriptionsEnabled is true (see dataloader-beans.xml), and then com_translations record is created using ItemDesc. It is recommended that translated ItemDescriptionsEnabled be set to false when loading Xcenter because non-local descriptions are used in translation records when loading ItemHdr.
13	LocalShorDesc	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		Ignored.

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

	RMS				Xstore				
	Field Name	**Description**	**Field Type**	**Max Width**	**Table**	**Column**	**Data Type**	**Default Value**	**Comment**
14	Ti	Number of shipping units (cases) that make up one tier of a pallet of this item for this location. A location may use a local pallet configuration if there are special space or receiving configurations that make typical pallets impractical. Multiply TI by HI to get the total number of cases for a pallet. Data is optional in this field.	Number	12	NA	NA	NA		Ignored.
15	Hi	Number of tiers that make up a complete pallet (height). A location may use a local pallet configuration if there are special space or receiving configurations that make typical pallets impractical. Multiply TI by HI to get the total number of cases for a pallet. Data is optional in this field.	Number	12	NA	NA	NA		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		Ignored.

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

	RMS				**Xstore**				
	Field Name	**Description**	**Field Type**	**Max Width**	**Table**	**Column**	**Data Type**	**Default Value**	**Comment**
17	Status	Current status of item at the location. Valid values are: A - Active, item is valid and can be ordered and sold. I - Inactive, item is valid, but cannot be ordered or sold. C - Discontinued, item is valid and sellable, but no longer orderable . D - Delete, item is pending delete and cannot be ordered or sold. This field will always have data for FULLITEMLOC, ITEMLOCCRE, and ITEMLOCMOD.	Character	1	itm_item_ options	ITEM_ AVAILAB ILITY_ CODE, STOCK_ STATUS	VARCHA R2(30 CHAR, VARCHA R2(60 CHAR)		If itemLoc:sta tus=A or C, the item_ availability _code is Available. If itemLoc:sta tus=I or D, the item_ availability _code is NA. If itemLoc:sta tus=C, the stock_ status is DISCONTI NUED. Otherwise, stock_ status is null.
18	DailyW astePct	Average percentage lost from inventory on a daily basis due to natural wastage. Data is optional in this field.	Number	12	NA	NA	NA		Ignored.
19	Measur eOfEac h	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		Ignored.
20	Measur eOfPric e	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		Ignored.

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

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

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

	RMS				Xstore				
	Field Name	**Description**	**Field Type**	**Max Width**	**Table**	**Column**	**Data Type**	**Default Value**	**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		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		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		Ignored.

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

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

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

	RMS				Xstore				
	Field Name	**Description**	**Field Type**	**Max Width**	**Table**	**Column**	**Data Type**	**Default Value**	**Comment**
29	Source Wh	This value is used by the ad-hoc PO/Transfer creation process to determine from which warehouse to fill the stores request. It is also used by the Allocation process to support warehouse to warehouse allocations. A value is required in this field if the sourcing method is Warehouse. Data is optional in this field.	Number	10	NA	NA	NA		Ignored.
30	UinTyp e	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		Used to determine a Serialized Item. If empty, itm_ item.seriali zed_item_ flag is set to 0. Otherwise, itm_ item.seriali zed_item_ flag is set to 1.
31	UinLab el	This column contains the label for the UIN when displayed in SIM. Data is optional in this field.	Character	6	NA	NA	NA		Ignored.
32	Capture TimeIn Proc	This column indicates when the UIN should be captured for an item during transaction processing. Data is optional in this field.	Character	6	NA	NA	NA		Ignored.

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

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

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

	RMS				Xstore				
	Field Name	Description	Field Type	Max Width	Table	Column	Data Type	Default Value	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_ QUANTI TY_ FLAG, FORCE_ QUANTI TY_OF_ ONE_ FLAG	NUMBER (1,0)		PROMPT_ FOR_ QUANTIT Y_FLAG - 1 if R, otherwise 0. FORCE_ QUANTIT Y_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)		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 ReSA and RMS. Data is optional in this field.	Character	6	NA	NA	NA		Ignored.

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

	RMS				Xstore				
	Field Name	**Description**	**Field Type**	**Max Width**	**Table**	**Column**	**Data Type**	**Default Value**	**Comment**
41	FoodStampInd	Indicates whether the item is approved for food stamps at the location. This indicator does not impact processing in ReSA and RMS. Data is optional in this field.	Character	1	itm_item_options	FOODSTAMP_ELIGIBLE_FLAG	NUMBER (1,0)		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 ReSA and RMS. Data is optional in this field.	Character	1	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		Ignored.
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		Ignored.

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

	RMS				Xstore				
	Field Name	Description	Field Type	Max Width	Table	Column	Data Type	Default Value	Comment
45	FixedTareUom	Holds the unit of measure value associated with the tare value. Data is optional in this field.	Character	4	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		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		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		Ignored.
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		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		Ignored.

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

	RMS				Xstore				
	Field Name	**Description**	**Field Type**	**Max Width**	**Table**	**Column**	**Data Type**	**Default Value**	**Comment**
51	Report Code	Code to determine which reports the location should run. Data is optional in this field.	Character	6	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		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		Ignored.
54	IBShelf Life	Holds the Investment Buy-specific shelf life for the item/location in days. Data is optional in this field.	Number	4	NA	NA	NA		Ignored.
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		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		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		Ignored.

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

	RMS				Xstore				
	Field Name	**Description**	**Field Type**	**Max Width**	**Table**	**Column**	**Data Type**	**Default Value**	**Comment**
58	InStore Market Basket	Holds the in-store market basket code for this item/location combination. Data is optional in this field.	Character	6	NA	NA	NA		Ignored.
59	Storage Locatio n	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		Ignored.
60	AltStora geLocat ion	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		Ignored.
61	Returna bleInd	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_ RETURN ABLE_ FLAG	NUMBER (1,0)		1 if N, otherwise 0.
62	Refund ableInd	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		Ignored.
63	BackOr derInd	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		Ignored.

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

	RMS				Xstore				
	Field Name	Description	Field Type	Max Width	Table	Column	Data Type	Default Value	Comment
64	MerchandiseInd	Note that the MerchandiseInd does not vary by locations (items are not physical entities in one location and fees in another). The MerchandiseInd is denormalized to the item/location level because it is a common use case for some downstream systems to care only about physical or non-physical items. This field is denormalized to assist those systems in their filtering logic. All items, both merchandise and non-merchandise, are exported from RMS. This field will always have data for all records.	Character	1	NA	NA	NA		If N, then NON_PHYSICAL item so record is skipped.
	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.
	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

	RMS				Xstore				
	Field Name	**Description**	**Field Type**	**Max Width**	**Table**	**Column**	**Data Type**	**Default Value**	**Comment**
0	Family	Value is always MERCHHIERARCHY.	Character	20	ITM_MERCH_HEIRARCHY	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

	RMS				Xstore				
	Field Name	**Description**	**Field Type**	**Max Width**	**Table**	**Column**	**Data Type**	**Default Value**	**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		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

	RMS				Xstore				
	Field Name	**Description**	**Field Type**	**Max Width**	**Table**	**Column**	**Data Type**	**Default Value**	**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)		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)		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)		node name

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

	RMS				Xstore				
	Field Name	Description	Field Type	Max Width	Table	Column	Data Type	Default Value	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		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)		parent heirarchy id

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

	RMS				Xstore				
	Field Name	**Description**	**Field Type**	**Max Width**	**Table**	**Column**	**Data Type**	**Default Value**	**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
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

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

	RMS				Xstore				
	Field Name	Description	Field Type	Max Width	Table	Column	Data Type	Default Value	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	ITM_MERCH_HEIRARCHY	CREATE_DATE	TIMESTAMP(6)		NA
	NA	NA	NA	NA	ITM_MERCH_HEIRARCHY	CREATE_USER_ID	VARCHAR2(30 CHAR)		NA
	NA	NA	NA	NA	ITM_MERCH_HEIRARCHY	UPDATE_DATE	TIMESTAMP(6)		NA
	NA	NA	NA	NA	ITM_MERCH_HEIRARCHY	UPDATE_USER_ID	VARCHAR2(30 CHAR)		NA

RMS Organizational Hierarchy

Table B–7 describes the RMS Organizational Hierarchy mapping.

Table B–7 RMS Organizational Hierarchy Mapping

	RMS				Xstore				
	Field Name	**Description**	**Field Type**	**Max Width**	**Table**	**Column**	**Data Type**	**Default Value**	**Comment**
0	Family	Value is always ORGHIERARCHY.	Character	20	NA	NA	NA		Ignored.

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

	RMS				Xstore				
	Field Name	**Description**	**Field Type**	**Max Width**	**Table**	**Column**	**Data Type**	**Default Value**	**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 (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		Determines action.

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

	RMS				Xstore				
	Field Name	**Description**	**Field Type**	**Max Width**	**Table**	**Column**	**Data Type**	**Default Value**	**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.							
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)		For the root COMPANY node, the column is hard-coded as "*".

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

	RMS				Xstore				
	Field Name	**Description**	**Field Type**	**Max Width**	**Table**	**Column**	**Data Type**	**Default Value**	**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)		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

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

	RMS				Xstore				
	Field Name	**Description**	**Field Type**	**Max Width**	**Table**	**Column**	**Data Type**	**Default Value**	**Comment**
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)		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	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)		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

	RMS				Xstore				
	Field Name	**Description**	**Field Type**	**Max Width**	**Table**	**Column**	**Data Type**	**Default Value**	**Comment**
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
8	CurrencyCode	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	loc_org_hierarchy	level_order	NUMBER (10,0)		0 - COMPANY 10 - CHAIN 20 - AREA 30 - REGION 40 - DISTRICT 1000 - STORE 2000 - Warehouse
	NA	NA	NA	NA	loc_org_hierarchy	sort_order	NUMBER (10,0)		Always set to 0.

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

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

RMS Related Item Detail

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

Table B–8 RMS Related Item Detail Mapping

	RMS				Xstore				
	Field Name	Description	Field Type	Max Width	Table	Column	Data Type	Default Value	Comment
0	Family	Value is always ITEMS. Data will always be present in this field.	Character	20	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		Determines Action. RelationshipType is looked up from rms_ related_ item_head by RelationshipId, Location, and OrganizationId.

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

	RMS				Xstore				
	Field Name	**Description**	**Field Type**	**Max Width**	**Table**	**Column**	**Data Type**	**Default Value**	**Comment**
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
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
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
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		Ignored.

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

	RMS				Xstore				
	Field Name	**Description**	**Field Type**	**Max Width**	**Table**	**Column**	**Data Type**	**Default Value**	**Comment**
6	StartDate	Optional start date that the item should be used in the relationship in DD-MON-YYYY format. If defined, this date indicates the first date the item should be offered as an upsell, substitute, or cross sell. Data is optional in this field.	Character	11	itm_substitute_items, itm_attached_items	BEGIN_DATETIME	TIMESTAMP(6)		NA
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	itm_substitute_items, itm_attached_items	ORGANIZATION_ID	NUMBER(10,0)	${dtv.location.organizationId}	Always uses default value.
	NA	NA	NA	NA	itm_substitute_items, itm_attached_items	CREATE_DATE	TIMESTAMP(6)	now()	Not set when Type =RELITEMDETMOD.
	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	itm_substitute_items, itm_attached_items	UPDATE_DATE	TIMESTAMP(6)	now()	Not set when Type =FULLRELITEMDET.

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

	RMS				Xstore				
	Field Name	**Description**	**Field Type**	**Max Width**	**Table**	**Column**	**Data Type**	**Default Value**	**Comment**
	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	itm_attached_items	PROMPT_TO_ADD_FLAG	NUMBER (1,0)	TRUE	Configurable in dataloader-beans.xml.
	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

	RMS				Xstore				
	Field Name	**Description**	**Field Type**	**Max Width**	**Table**	**Column**	**Data Type**	**Default Value**	**Comment**
0	Family	Value is always ITEMS. Data will always be present in this field.	Character	20	NA	NA	NA		Ignored.
1	Type	Value is always FULLRELITEMHDR when the program is run in full mode. If the program is run in delta mode, the type will vary based on the type of change (CRE, MOD, DEL). Valid values are: RELITEMHEADCRE RELITEMHEADMOD RELITEMHEADDEL Data will always be present in this field.	Character	15	NA	NA	NA		Determines action. MOD will move records between itm_substute_items and item_attached_items when the modified record represents a relationship type change between CRSL/UPSL and SUBS.

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

	RMS				Xstore				
	Field Name	Description	Field Type	Max Width	Table	Column	Data Type	Default Value	Comment
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)		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)		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)		Used as levelValue in itm_substute_items and item_attached_items records.
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)		Ignored.

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

	RMS				Xstore				
	Field Name	**Description**	**Field Type**	**Max Width**	**Table**	**Column**	**Data Type**	**Default Value**	**Comment**
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)		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

	RMS				Xstore				
	Field Name	**Description**	**Field Type**	**Max Width**	**Table**	**Column**	**Data Type**	**Default Value**	**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 RMS/ReSA 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)		No logic is based on this data.

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

	RMS				Xstore				
	Field Name	Description	Field Type	Max Width	Table	Column	Data Type	Default Value	Comment
	NA	NA	NA	NA	rms_related_item_head	ORGANIZATION_ID	NUMBER (10,0)	${dtv.location.organizationId}	Always uses default value.
	NA	NA	NA	NA	rms_related_item_head	CREATE_DATE	TIMESTAMP(6)	now()	Not set when Type =ITEMHDRMOD.
	NA	NA	NA	NA	rms_related_item_head	UPDATE_DATE	TIMESTAMP(6)	now()	Not set when Type =FULLRELITEMHDR
	NA	NA	NA	NA	item_item_options	ATTACHED_ITEMS_FLAG	NUMBER (1,0)		Set to 1 when CRSL or UPSL, otherwise 0 for Item.
	NA	NA	NA	NA	item_item_options	SUBSTITUTE_AVAILABLE_FLAG	NUMBER (1,0)		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

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

Table B–10 (Cont.) RMS Store Mapping

	RMS				Xstore				
	Field Name	**Description**	**Field Type**	**Max Width**	**Table**	**Column**	**Data Type**	**Default Value**	**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		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
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
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
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		Ignored.

Table B–10 (Cont.) RMS Store Mapping

	RMS				Xstore				
	Field Name	Description	Field Type	Max Width	Table	Column	Data Type	Default Value	Comment
6	StoreNameSec	Contains an secondary name for this store. Data in this field is optional.	Character	150	NA	NA	NA		Ignored.
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		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		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
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		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		Ignored.
12	Aquire Date	Contains the date on which the store was acquired. Data in this field is optional. Format is YYYYMMDD.	Date	8	NA	NA	NA		Ignored.

Table B–10 (Cont.) RMS Store Mapping

	RMS				Xstore				
	Field Name	**Description**	**Field Type**	**Max Width**	**Table**	**Column**	**Data Type**	**Default Value**	**Comment**
13	RemodelDate	Contains the date on which the store was last remodeled. Data in this field is optional. Format is YYYYMMDD.	Date	8	NA	NA	NA		Ignored.
14	FaxNumber	Contains the fax number for the store. Data in this field is optional.	Character	20	loc_rtl_loc	TELEPHONE_2	VARCHAR(32)		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)		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)		Optional email address.
17	TotalSqFeet	Contains the total square footage of the store. Data in this field is optional.	Number	8	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		Ignored.
19	LinearDistance	Holds the total merchandisable space of the location. Data in this field is optional.	Number	8	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
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		Ignored.

Table B–10 (Cont.) RMS Store Mapping

	RMS				Xstore				
	Field Name	**Description**	**Field Type**	**Max Width**	**Table**	**Column**	**Data Type**	**Default Value**	**Comment**
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		Ignored.
23	ChannelId	Contains the channel with which the store is associated. Data in this field is optional.	Number	4	NA	NA	NA		Ignored.
24	ChannelName	Contains the name of the channel. Data in this field is optional.	Character	120	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		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		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		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		Ignored.
29	Transfer Zone	NA	Number	4	NA	NA	NA		Ignored.

Table B–10 (Cont.) RMS Store Mapping

	RMS				Xstore				
	Field Name	**Description**	**Field Type**	**Max Width**	**Table**	**Column**	**Data Type**	**Default Value**	**Comment**
30	Default Wh	Contains the number of the warehouse that may be used as the default for creating cross-dock masks.	Number	10	NA	NA	NA		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		Ignored.
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		Ignored.
33	CurrencyCode	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)		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		Ignored.

Table B–10 (Cont.) RMS Store Mapping

	RMS				Xstore				
	Field Name	**Description**	**Field Type**	**Max Width**	**Table**	**Column**	**Data Type**	**Default Value**	**Comment**
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		Ignored.
36	IntPOSInd	Indicates whether or not the POS system at the store is integrated. Data will always be present in this field for creation and modification records.	Character	1	NA	NA	NA		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		Ignored.
38	DunsLoc	Legacy, generally not used. Data in this field is optional.	Character	4	NA	NA	NA		Ignored.

Table B–10 (Cont.) RMS Store Mapping

	RMS				Xstore				
	Field Name	**Description**	**Field Type**	**Max Width**	**Table**	**Column**	**Data Type**	**Default Value**	**Comment**
39	SisterStore	This field holds a store number which is used to relate the current store to the historical data of an existing store. Data in this field is optional.	Number	10	NA	NA	NA		Ignored.
40	TsfEntityId	Data in this field is optional.	Number	10	NA	NA	NA		Ignored.
41	OrgUnitId	Column contains the organizational unit ID value. Data in this field is optional.	Number	15	NA	NA	NA		Ignored.
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		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		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		Ignored.

Table B–10 (Cont.) RMS Store Mapping

	RMS				Xstore				
	Field Name	Description	Field Type	Max Width	Table	Column	Data Type	Default Value	Comment
45	WFCustomer	Numeric ID of the customer. Data in this field is optional.	Number	10	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		Ignored.
47	CustomerOrderLocInd	This column determines whether the location is a customer order location. If the indicator is Y, the location can be used by OMS for sourcing/ fulfillment or both. Otherwise, it cannot be used. It is used only for the company stores.	Character	1	NA	NA	NA		Ignored.
	NA	NA	NA	NA	loc_rtl_loc	STORE_NBR	VARCHAR(254)		Always RMS StoreId.
	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	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	loc_rtl_loc	use_till_accountability_flag	NUMBER(1,0)	FALSE	Value can be configured in dataloader-beans-xml.

Table B–10 (Cont.) RMS Store Mapping

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

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

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

	RMS				Xstore				
	Field Name	**Description**	**Field Type**	**Max Width**	**Table**	**Column**	**Data Type**	**Default Value**	**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: 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		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
3	AddrType	Contains the code used to identify the address type. Data will always be present in this field.	Character	2	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		Ignored.

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

	RMS				Xstore				
	Field Name	**Description**	**Field Type**	**Max Width**	**Table**	**Column**	**Data Type**	**Default Value**	**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		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
7	Add2	Contains the second line of the address. Optional.	Character	240	loc_rtl_loc	ADDRESS2	VARCHAR2(254 CHAR)		NA
8	Add3	Contains the third line of the address. Optional.	Character	240	loc_rtl_loc	ADDRESS3	VARCHAR2(254 CHAR)		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
10	County	Contains the county name for the location. Optional.	Character	250	NA	NA	NA		NA
11	State	Contains the state abbreviation for the address. Optional.	Character	3	loc_rtl_loc	STATE	VARCHAR2(30 CHAR)		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
13	PostCode	Contains the postal code name for the location. Optional.	Character	30	loc_rtl_loc	POSTAL_CODE	VARCHAR2(30 CHAR)		NA
14	JurisdictionCode	Contains the tax jurisdiction code for the location. Optional.	Character	10	NA	NA	NA		Ignored.

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

	RMS				Xstore				
	Field Name	**Description**	**Field Type**	**Max Width**	**Table**	**Column**	**Data Type**	**Default Value**	**Comment**
15	Contact Name	Contains the name of the contact person at this address. Optional.	Character	120	NA	NA	NA		Ignored.
16	Contact Phone	Contains a phone number of the contact person at this address. Optional.	Character	20	NA	NA	NA		Ignored.
17	Contact Fax	Contains a fax number of the contact person at this address. Optional.	Character	20	NA	NA	NA		Ignored.
18	Contact Email	Contains an email for the contact person at this address. Optional.	Character	100	NA	NA	NA		Ignored.
	NA	NA	NA	NA	loc_rtl_loc	ORGANIZATION_ID	NUMBER (10,0)	${dtv.location.organizationId}	Always uses default value.
	NA	NA	NA	NA	loc_rtl_loc	UPDATE_DATE	TIMESTAMP(6)	now()	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

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

Table B–12 (Cont.) RMS VAT Mapping

	RMS				Xstore				
	Field Name	**Description**	**Field Type**	**Max Width**	**Table**	**Column**	**Data Type**	**Default Value**	**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		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)		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)		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

Table B–12 (Cont.) RMS VAT Mapping

	RMS				Xstore				
	Field Name	Description	Field Type	Max Width	Table	Column	Data Type	Default Value	Comment
5	VatCodeDesc	This field contains the description of the VAT Code. This field will always have data in the VATCRE and VATMOD records.	Character	120	tax_tax_group, tax_tax_group_rule	NAME, DESCRIPTION	VARCHAR(254)		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
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	tax_tax_group_rule	TAX_TYPCODE	VARCHAR(30)	VAT	Always uses default value.
	NA	NA	NA	NA	tax_tax_group_rule	TAX_RULE_SEQ_NBR	NUMBER (10,0)	1	Always uses default value. The assumption is that there can only be one tax rule per VAT region/VAT code combination.

Table B–12 (Cont.) RMS VAT Mapping

	RMS				Xstore				
	Field Name	Description	Field Type	Max Width	Table	Column	Data Type	Default Value	Comment
	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

	RMS				Xstore				
	Field Name	**Description**	**Field Type**	**Max Width**	**Table**	**Column**	**Data Type**	**Default Value**	**Comment**
	NA	NA	NA	NA	tax_tax_ rate_rule	EXPR_ DATETIME	TIMESTAMP(6)		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

	RMS				Xstore				
	Field Name	**Description**	**Field Type**	**Max Width**	**Table**	**Column**	**Data Type**	**Default Value**	**Comment**
						EXPR_ DATETIME (Continued)			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	tax_tax_ rate_rule	TAX_ RULE_ SEQ_ NBR		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	tax_tax_ rate_rule	TAX_ RATE_ RULE_ SEQ	NUMBER (10,0)		This has to be derived by the integration layer. For a newly created VAT rate, its sequence number is one after the maximum rate rule sequence number in the table.

Table B–12 (Cont.) RMS VAT Mapping

	RMS				Xstore				
	Field Name	**Description**	**Field Type**	**Max Width**	**Table**	**Column**	**Data Type**	**Default Value**	**Comment**
	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.
	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	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	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	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.

Table B–12 (Cont.) RMS VAT Mapping

	RMS				Xstore				
	Field Name	**Description**	**Field Type**	**Max Width**	**Table**	**Column**	**Data Type**	**Default Value**	**Comment**
	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.
	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	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

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

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

	RMS				Xstore				
	Field Name	**Description**	**Field Type**	**Max Width**	**Table**	**Column**	**Data Type**	**Default Value**	**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		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
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		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		Ignored.

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

	RMS				Xstore				
	Field Name	**Description**	**Field Type**	**Max Width**	**Table**	**Column**	**Data Type**	**Default Value**	**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		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)		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		Ignored.

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

	RMS				Xstore				
	Field Name	**Description**	**Field Type**	**Max Width**	**Table**	**Column**	**Data Type**	**Default Value**	**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		Ignored.
	NA	NA	NA	NA	itm_item_options	UPDATE_USER_ID	VARCHAR2(30 CHAR)	DATALOADER	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

	RPM			Xstore				
	Field Name	**Description**	**Field Type**	**Table**	**Column**	**Data Type**	**Default Value**	**Comment**
0	Record Descriptor	File Detail Marker (1 per clearance create/modify)	Char(5)	NA	NA	NA		Used to determine unit type.
1	Line Id	Unique line identification	Number(10)	NA	NA	NA		Ignored.
2	Event Type	CRE = Create, MOD = Modify	Char(3)	NA	NA	NA		Used to determine the action type.
3	Id	Clearance identifier	Number(15)	itm_item_prices	EXTERNAL_ID	VARCHAR2(60 CHAR)		NA
4	Item	Item identifier	Char(25)	itm_item_prices	ITEM_ID	VARCHAR2(60 CHAR)		NA
5	Effective Date	Clearance Effective Date (YYYYMMDDHH24MISS)	Date	itm_item_prices	EFFECTIVE_DATE	TIMESTAMP(6)		NA

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

	RPM			Xstore				
	Field Name	**Description**	**Field Type**	**Table**	**Column**	**Data Type**	**Default Value**	**Comment**
6	Selling Retail	Selling retail with price change applied	Number(20,4)	itm_item_prices	PRICE	NUMBER(17,6)		NA
7	Selling Retail UOM	Selling retail unit of measure	Char(4)	NA	NA	NA		Ignored
8	Selling Retail Currency	Selling retail currency	Char(3)	NA	NA	NA		Ignored (assumed to be base Currency).
9	Reset Clearance ID	Clearance reset identification	Number(15)	NA	NA	NA		Used to detect clearance reset. Clearance reset is when this value exactly matches the ID field's value.
	NA	NA	NA	itm_item_prices	ORGANIZATION_ID	NUMBER(10,0)	${dtv.location.organizationId}	Always uses default value.
	NA	NA	NA	itm_item_prices	PROPERTY_CODE	VARCHAR2(60 CHAR)		CLEARANCE_PRICE when Clearance event, or REGULAR_PRICE when Clearance Reset event.
	NA	NA	NA	itm_item_prices	CREATE_DATE	TIMESTAMP(6)	now()	Not set when EventType = Mod.
	NA	NA	NA	itm_item_prices	CREATE_USER_ID	VARCHAR2(30 CHAR)	DATALOADER	Not set when EventType = Mod.
	NA	NA	NA	itm_item_prices	UPDATE_DATE	TIMESTAMP(6)	now()	Always set.
	NA	NA	NA	itm_item_prices	UPDATE_USER_ID	VARCHAR2(30 CHAR)	DATALOADER	Always set.
	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

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

	RPM			Xstore				
	Field Name	**Description**	**Field Type**	**Table**	**Column**	**Data Type**	**Default Value**	**Comment**
0	Record Descriptor	File Detail Marker (1 per price change create or modify)	Char(5)	NA	NA	NA		Type of event.
1	Line Id	Unique line identification	Number(10)	NA	NA	NA		Ignored.
2	Event Type	CRE = Create, MOD = Modify	Char(3)	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
4	Item	Item identifier	Char(25)	itm_item_prices	ITEM_ID	VARCHAR2(60 CHAR)		NA

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

	RPM			Xstore				
	Field Name	**Description**	**Field Type**	**Table**	**Column**	**Data Type**	**Default Value**	**Comment**
5	Effective Date	Effective Date of the price change (YYYYMMDDHH24MISS)	NA	itm_item_prices	EFFECTIVE_DATE	TIMESTAMP(6)		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		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
8	Selling Retail UOM	Selling retail unit of measure	Char(4)	itm_item	UNIT_OF_MEASURE_CODE	VARCHAR2(30 BYTE)		Ignored.
9	Selling Retail Currency	Selling retail currency	Char(3)	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
11	Multi-Units	NA	Number(12,4)	NA	NA	NA		NA
12	Multi-Unit Retail	Number of multi-units	Number(20,4)	NA	NA	NA		NA
13	Multi-Unit UOM	Multi-Unit Retail unit of measure	Char(4)	NA	NA	NA		NA
14	Multi-Unit Currency	Multi-Unit Retail Currency	Char(3)	NA	NA	NA		NA
	NA	NA	NA	itm_item_prices	ORGANIZATION_ID	NUMBER(10,0)	${dtv.location.organizationId}	Always uses the default value.
	NA	NA	NA	itm_item_prices	PROPERTY_CODE	VARCHAR2(60 CHAR)		REGULAR_PRICE will be the value.

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

	RPM			Xstore				
	Field Name	**Description**	**Field Type**	**Table**	**Column**	**Data Type**	**Default Value**	**Comment**
	NA	NA	NA	itm_item_prices	CREATE_DATE	TIMESTAMP(6)	now()	Current time stamp. Not set when the event type =MOD.
	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	itm_item_prices	UPDATE_DATE	TIMESTAMP(6)	now()	Current time stamp. Field value only set when event type=MOD.
	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	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

	RPM			Xstore				
	Field Name	**Description**	**Field Type**	**Table**	**Column**	**Data Type**	**Default Value**	**Comment**
0	Record Descriptor	File Detail Delete Marker (1per price change delete)	Char(5)	NA	NA	NA		Used by the transformer to determine the unit type.

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

	RPM			Xstore				
	Field Name	**Description**	**Field Type**	**Table**	**Column**	**Data Type**	**Default Value**	**Comment**
1	Line Id	Unique line identification	Number(10)	NA	NA	NA		Ignored.
2	Id	Price change identifier	Number(15)	itm_item_prices	EXTERNAL_ID	VARCHAR2(60 CHAR)		NA
3	Item	Item identifier	Char(25)	itm_item_prices	ITEM_ID	VARCHAR2(60 CHAR)		NA
	NA	NA	NA	itm_item_prices	ORGANIZATION_ID	NUMBER(10,0)	${dtv.location.organizationId}	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					Xstore		
Record Name	**Field Name**	**Field Type**	**Default Value**	**Description**	**Table**	**Field**	**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
	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.

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

RPM					Xstore		
Record Name	**Field Name**	**Field Type**	**Default Value**	**Description**	**Table**	**Field**	**Comment**
	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	descriptio n	NA
	Promo Type	Number(2)		Valid values: 0 = Multi-Buy Promotion 1=- Simple Promotion 2 = Threshold Promotion 3 = Finance Promotion (formerly tied to a value of 6) 4 = Transaction Promotion	NA	NA	Determines structure of resulting deal
	Promo Comp Detail ID	Number(10)		Promotion Component Detail identifier	prc_deal	deal_id (partially)	NA
	Date Start	Date		Start Date of Promotion Component Detail (YYYYMMDDHH24 MISS)	prc_deal	effective_ date	NA
	End Date	Date		End Date of Promotion Component Detail (YYYYMMDDHH24 MISS)	prc_deal	end_date	NA

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

RPM					Xstore		
Record Name	**Field Name**	**Field Type**	**Default Value**	**Description**	**Table**	**Field**	**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					Xstore		
Record Name	Field Name	Field Type	Default Value	Description	Table	Field	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					Xstore		
Record Name	Field Name	Field Type	Default Value	Description	Table	Field	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					Xstore		
Record Name	**Field Name**	**Field Type**	**Default Value**	**Description**	**Table**	**Field**	**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 - YYYYMMDDHH24 MISS	NA	NA	Ignored.
	Selling Retail Currency	Char(3)		Selling retail currency	NA	NA	Ignored.
TPCDT	Record Descriptor	Char(5)	TPCDT	Credit Detail	NA	NA	Ignored.
	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.

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

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