

**Oracle® Retail Xstore Suite 18.0/Oracle Retail
Merchandising Foundation Cloud Service 16.0.030
and Oracle Retail Pricing Cloud Service 16.0.030**

Implementation Guide

Release 18.0/16.0.030

F14603-04

September 2020

Oracle Retail Xstore Suite 18.0/RMFCs 16.0.030 and RPCS16.0.030 Implementation Guide, Release 18.0/16.0.030

F14603-04

Copyright © 2020, Oracle and/or its affiliates. All rights reserved.

Primary Author: Gerlinde Rust

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

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

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

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

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

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

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

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

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

Value-Added Reseller (VAR) Language

Oracle Retail VAR Applications

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

- (i) the **MicroStrategy** Components developed and licensed by MicroStrategy Services Corporation (MicroStrategy) of McLean, Virginia to Oracle and imbedded in the MicroStrategy for Oracle Retail Data Warehouse and MicroStrategy for Oracle Retail Planning & Optimization applications.
- (ii) the **Wavelink** component developed and licensed by Wavelink Corporation (Wavelink) of Kirkland, Washington, to Oracle and imbedded in Oracle Retail Mobile Store Inventory Management.
- (iii) the software component known as **Access Via™** 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

Send Us Your Comments	xi
Preface	xiii
Audience.....	xiii
Documentation Accessibility	xiii
Related Documents	xiii
Customer Support	xiv
Review Patch Documentation	xiv
Improved Process for Oracle Retail Documentation Corrections	xiv
Oracle Retail Documentation on the Oracle Technology Network	xv
Conventions	xv
1 Overview	
2 Data Flow from Merchandising to Xstore using Ominchannel Data Service (OCDS)	
Conceptual Data Flow	2-1
Xadmin Deployments.....	2-3
Manual Refresh of an Xstore Database from OCDS.....	2-3
3 Transaction Flow from Xstore to ReSA	
Conceptual Data Flow	3-1
Technical Implementation	3-2
Xstore Broadcaster	3-2
RTLog Generator.....	3-3
ReSA saimptlog/i.....	3-3
4 OCDS Integration Configuration	
Xadmin-OCDS Integration Communication	4-1
Xadmin System Settings	4-1
Omnichannel Data Service Configuration Options.....	4-2
OCDS Database Tables	4-3
Seed Data for VAT and Non-VAT Organizations.....	4-3

Download: Immediate vs Store Close	4-4
--	-----

5 Integration Considerations

Foundation Data	5-1
Seed Data	5-1
Transaction Details.....	5-1
Currency Exchange Rates	5-2
Stores.....	5-2
Merchandise Hierarchy	5-2
Items	5-3
Merchandise Items	5-3
Non-Merchandise Items.....	5-3
Kit Items.....	5-3
Differentiators.....	5-3
Product Restrictions.....	5-4
Related Items	5-4
Other Item Attribute Notes.....	5-4
Tax	5-4
Value Added Tax (VAT)	5-5
US Sales Tax	5-5
Inventory	5-6
Serialized Inventory.....	5-6
Customer Orders	5-6
In-Store Orders	5-6
Recognition of a Sale.....	5-6
Pricing	5-7
Sales Audit	5-7
Register-level Balancing	5-7
Sales Person.....	5-7
Tender Type	5-7
Coupons.....	5-7
Promotion	5-7

6 RTLog Generator On-Premises

Configuration	6-1
Deployment	6-6
WebLogic Cluster Setup	6-8
Deployment of the RTLog Generator Application on a Cluster	6-19
Security Configuration	6-22
Container Level Security	6-22
Transport Level Security	6-25
Complete the Security Configuration	6-27

7 RTLog Generator Cloud

RTLog Generator Cloud	7-1
Configuration	7-1

Integration	7-1
Updating Mapping Configuration	7-1
Retrieving Published RTLog Files	7-3
Security Configuration	7-3
Acquiring IDCS Token	7-3
Provide IDCS Authentication	7-4

A Appendix: POSLog to RTLog Mapping Details

Transaction Type Mapping	A-1
Tender Type Mapping	A-3
Total Tender ID Mapping	A-5
Item Type Mapping	A-6
Reason Code Mapping	A-7
ReSA Reason Codes	A-7
ReSA Return Reason Codes.....	A-8
ReSA Discount Reason Codes	A-8
ReSA Item Price Override Reason Codes	A-9
Item Status/Sales Type Mapping	A-9

List of Figures

2-1	OCDS Component Diagram.....	2-2
2-2	Conceptual Data Flow from Merchandising and Pricing to Xstore POS.....	2-2
2-3	Xstore Office Deployments	2-3
2-4	Xstore Office - Data Publisher Option	2-4
2-5	Xstore Office - Data to Publish.....	2-4
2-6	Xstore Office - Target Organization Node Pop Up.....	2-5
3-1	Xstore to ReSA Transaction Flow	3-2
4-1	Xstore Office - Ominichannel Data Service System Settings	4-2
4-2	OCDS Subtask Details Table	4-4
6-2	Example of context-param Field Update.....	6-2
6-5	RTLogMappingConfig.xml Field Mapper Example 2.....	6-6
6-7	Administration Console Control Page.....	6-7
6-8	Administration Console Install Application Assistant Page	6-7
6-10	Configuration Wizard Configuration Type Page.....	6-9
6-11	Configuration Wizard Templates Page	6-9
6-12	Configuration Wizard Administrator Account Page	6-10
6-13	Configuration Wizard Domain Mode and JDK.....	6-10
6-14	Configuration Wizard Advanced Configuration Page	6-11
6-15	Configuration Wizard Administration Server Page	6-11
6-16	Configuration Wizard Node Manager Page.....	6-12
6-17	Configuration Wizard Managed Servers Page.....	6-13
6-18	Configuration Wizard Clusters Page	6-13
6-19	Configuration Wizard Assign Servers to Clusters Page	6-14
6-20	Configuration Wizard HTTP Proxy Applications Page.....	6-14
6-21	Configuration Wizard Machines Page.....	6-15
6-22	Configuration Wizard Assign Servers to Machines Page.....	6-15
6-23	Configuration Wizard Configuration Summary Page	6-16
6-24	Administration Console Settings Page	6-18
6-25	Administration Console Configuration Page	6-18
6-27	Administration Console Deployments Page.....	6-19
6-28	Administration Console Install Application Assistant Page	6-20
6-29	Install Application Assistant Select Deployment Targets Page	6-20
6-30	Summary of Deployments Page	6-21
6-32	WebLogic Plugin Enabled Parameter	6-22
6-33	Administration Console Summary of Security Realms Page.....	6-22
6-34	Create a New Group Page	6-23
6-35	Create a New User Page	6-23
6-36	Users Page.....	6-24
6-37	User Settings Page.....	6-24
6-39	Administration Console Servers Page	6-25
6-40	Keystores Settings.....	6-26
6-41	Settings for the Administration Server	6-26

List of Tables

4-1	Properties - OCDS URL and Service Path Prefix.....	4-1
4-2	OCDS System Settings	4-2
4-3	OCDS Database Tables.....	4-3
7-1	REST Services related to the RTLogMappingConfig.xml.....	7-2
A-1	Transaction Type Mapping	A-1
A-2	Tender Type Mapping.....	A-4
A-3	Total Tender ID Mapping.....	A-6
A-4	Item Type Mapping	A-6
A-5	ReSA Reason Codes.....	A-7
A-6	ReSA Return Reason Codes.....	A-8
A-7	ReSA Discount Reason Codes.....	A-9
A-8	ReSA Item Price Override Reason Codes.....	A-9
A-9	ReSA Item Status/Sales Type Mapping	A-10

Send Us Your Comments

Oracle Retail Xstore Suite 18.0/RMFCS 16.0.030 and RPCS16.0.030 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.
<i>italic</i>	Italic type indicates book titles, emphasis, or placeholder variables for which you supply particular values.
monospace	Monospace type indicates commands within a paragraph, URLs, code in examples, text that appears on the screen, or text that you enter.

Overview

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

- Foundation and price data from Oracle Retail Merchandising Foundation Cloud Service (RMFCS) and Oracle Retail Pricing Cloud Service (RPCS) to Oracle Retail Suite flow into the Omnichannel Data Service (OCDS).

OCDS is a component of the Oracle Retail Integration Cloud Service (RICS).

Xstore Office communicates directly with OCDS to request data-changes reported to OCDS by RMFCS and RPCS. When changes to merchandising and pricing data are detected, Xstore .mnt files are automatically generated and distributed for data loading into Xcenter and Xstore databases.

- 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 using Omnichannel Data Service \(OCDS\)"](#): This chapter describes the flow of data from the Merchandising applications to Omnichannel Data Service to the Xstore Suite.
- [Chapter 3, "Transaction Flow from Xstore to ReSA"](#): This chapter describes the flow of transactions from Xstore Point of Service to ReSA.
- [Chapter 4, "OCDS Integration Configuration"](#): This chapter provides information on the configuration changes that can be made for the integration.
- [Chapter 5, "Integration Considerations"](#): This chapter covers functional and technical points about the integration that need to be taken into consideration when implementing the integration.
- [Chapter 6, "RTLog Generator On-Premises"](#): 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.



Data Flow from Merchandising to Xstore using Ominchannel Data Service (OCDS)

This chapter covers the data flow from Oracle Retail Merchandising Foundation Cloud Service (RMFCS) and Oracle Retail Pricing Cloud Service (RPCS) to OCDS, where merchandising and pricing data can be requested by Xstore Office for loading into the Xcenter and Xstore databases.

OCDS can provide the Xstore Suite the following categories of data:

- Merchandise hierarchy
- Organizational hierarchy
- Store (including addresses)
- Dimension type (derived from Item DIFF usage)
- Dimension value (derived from Item DIFF usage)
- Items
- VAT tax rules and item associations
- Related items
- Initial prices
- Price changes
- Clearance prices

Conceptual Data Flow

Oracle Omnichannel Cloud Data Service (OCDS) is a data hub, enabling the Oracle Retail Merchandising and Pricing applications to share information with the Oracle Retail Omnichannel applications. OCDS is composed of three major components:

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

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

Figure 2–1 OCDS Component Diagram

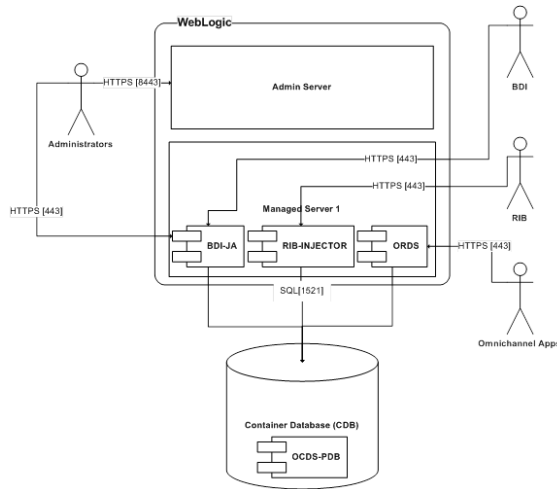
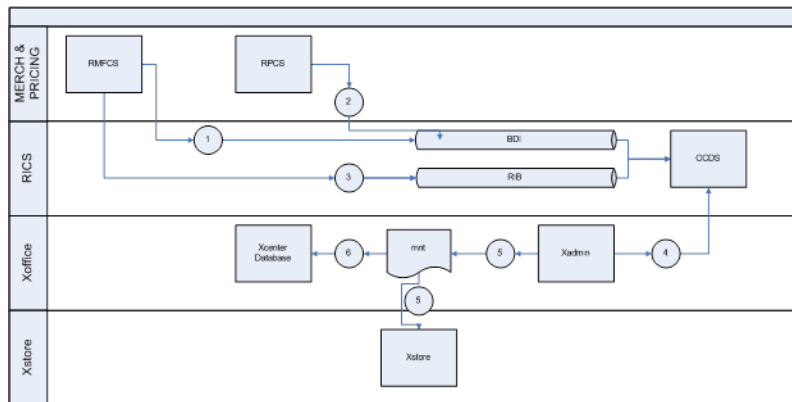


Figure 2–2 Conceptual Data Flow from Merchandising and Pricing to Xstore POS



The following steps describe the flow in Figure 2–2.

1. OCDS receives an initial load of merchandising foundation data using BDI as the data transport. This is generally a one-time push of data over BDI into OCDS.
2. OCDS starts to receive, on-going, regularly scheduled pricing data using BDI as the data transport.
3. OCDS starts to receive, on-going, near-real-time updates of merchandising data using the RIB as the data transport.
4. Xadmin starts polling OCDS, at a regularly scheduled interval, to check for updates (for example, additions, deletions, and modifications) of merchandising and pricing data used by the Xstore Suite. Xadmin communicates requests for changes to OCDS data by calling the OCDS REST web services.
5. When changes to OCDS data are detected, Xadmin generates .mnt files containing the commands to update Xstore Suite databases. When .mnt files are generated, they are automatically deposited into the Xstore Office auto-drop folder for data-loading and distribution.

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

Xadmin Deployments

When the OCDS integration is enabled, Xadmin will automatically generate .mnt files with instructions for updating the Xstore Suite databases when merchandising or pricing data changes are detected. Deployments of generated .mnt files, to be loaded at the store, are automatically created for either immediate or scheduled distribution. Each deployment status of the files is displayed in the Xstore Office Deployments screen.

Figure 2–3 Xstore Office Deployments

ID	Name	Type	Plan	Launch Date	Waves Complete	Status
10239	2_dimensionValue_1542664871	Auto Deploy	Single Wave	Nov 19, 2018	0 out of 1	Scheduled
10238	2_dimensionType_1542664869	Auto Deploy	Single Wave	Nov 19, 2018	0 out of 1	Scheduled
10237	5_price_1542664518196_STOF	Auto Deploy	Single Wave	Nov 19, 2018	0 out of 1	Scheduled
10236	5_price_1542664516788_STOF	Auto Deploy	Single Wave	Nov 19, 2018	1 out of 1	Complete
10235	5_price_1542664516268_STOF	Auto Deploy	Single Wave	Nov 19, 2018	0 out of 1	Scheduled
10234	5_price_1542664513595_STOF	Auto Deploy	Single Wave	Nov 19, 2018	0 out of 1	Scheduled
10233	4_itemLoc_1542664517424_ST	Auto Deploy	Single Wave	Nov 19, 2018	0 out of 1	Scheduled
10232	4_itemLoc_1542664515629_ST	Auto Deploy	Single Wave	Nov 19, 2018	0 out of 1	Scheduled
10231	4_itemLoc_1542664515492_ST	Auto Deploy	Single Wave	Nov 19, 2018	0 out of 1	Scheduled
10230	4_itemLoc_1542664513014_ST	Auto Deploy	Single Wave	Nov 19, 2018	0 out of 1	Scheduled
10229	3_item_1542664517312_STOR	Auto Deploy	Single Wave	Nov 19, 2018	0 out of 1	Scheduled
10228	3_item_1542664515522_STOR	Auto Deploy	Single Wave	Nov 19, 2018	0 out of 1	Scheduled
10227	3_item_1542664515253_STOR	Auto Deploy	Single Wave	Nov 19, 2018	0 out of 1	Scheduled
10226	3_item_1542664512850_STOR	Auto Deploy	Single Wave	Nov 19, 2018	0 out of 1	Scheduled
10225	5_price_1542663078013_STOF	Auto Deploy	Single Wave	Nov 19, 2018	0 out of 1	Scheduled
10224	5_price_1542663076679_STOF	Auto Deploy	Single Wave	Nov 19, 2018	1 out of 1	Complete

Manual Refresh of an Xstore Database from OCDS

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

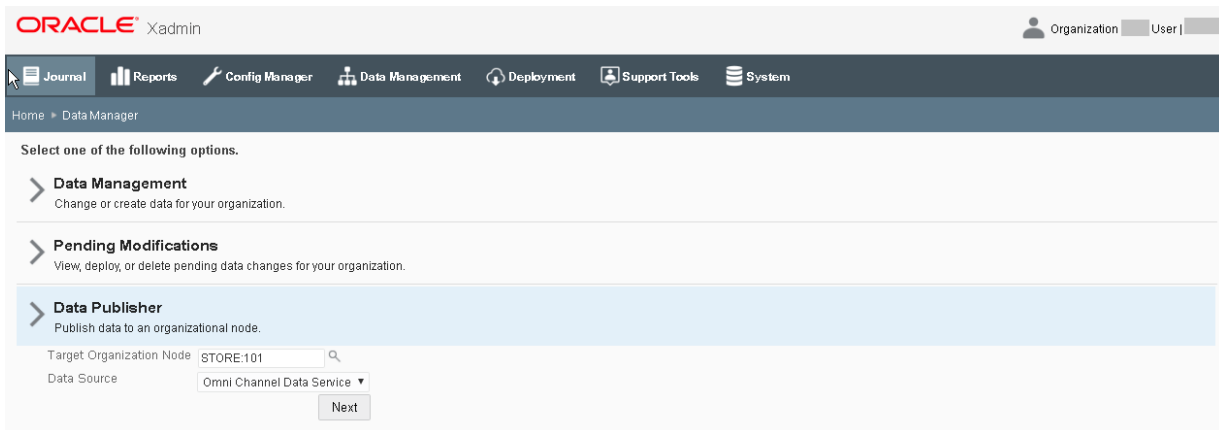
Use of the Data Publisher to replenish one or more types of OCDS data at a store will result in the purging of all existing OCDS-sourced data, followed by the loading of a full set of the most recent OCDS data.

When the "OCDS Enabled" option in Xadmin's System Settings is set to true, the DataManager screen's "Data Publisher" option will include a "Data Source" drop down list, which includes the list option: Omni Channel Data Service.

To publish OCDS data to one or more stores:

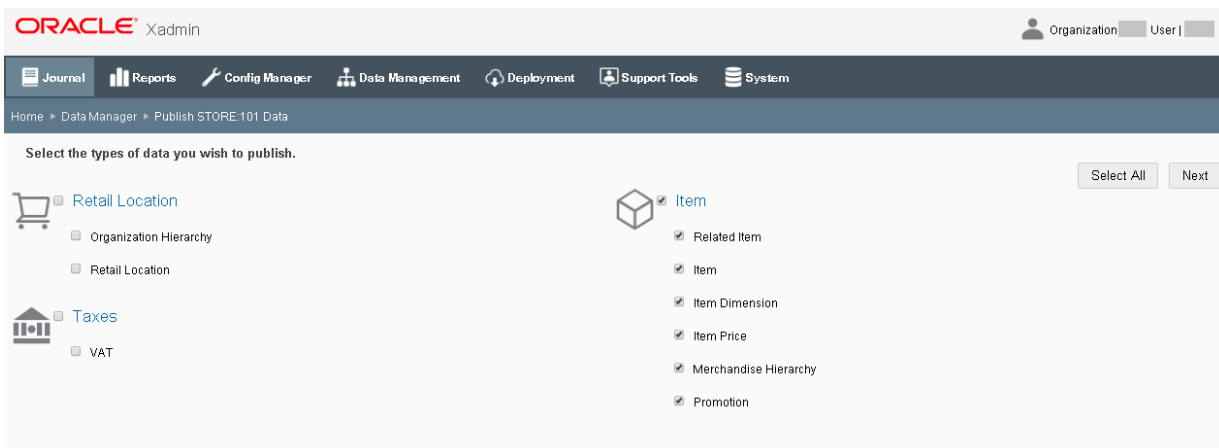
1. Choose the desired Organization Node for the target stores, select Omni Channel Data Service, and click **Next**.

Figure 2–4 Xstore Office - Data Publisher Option



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

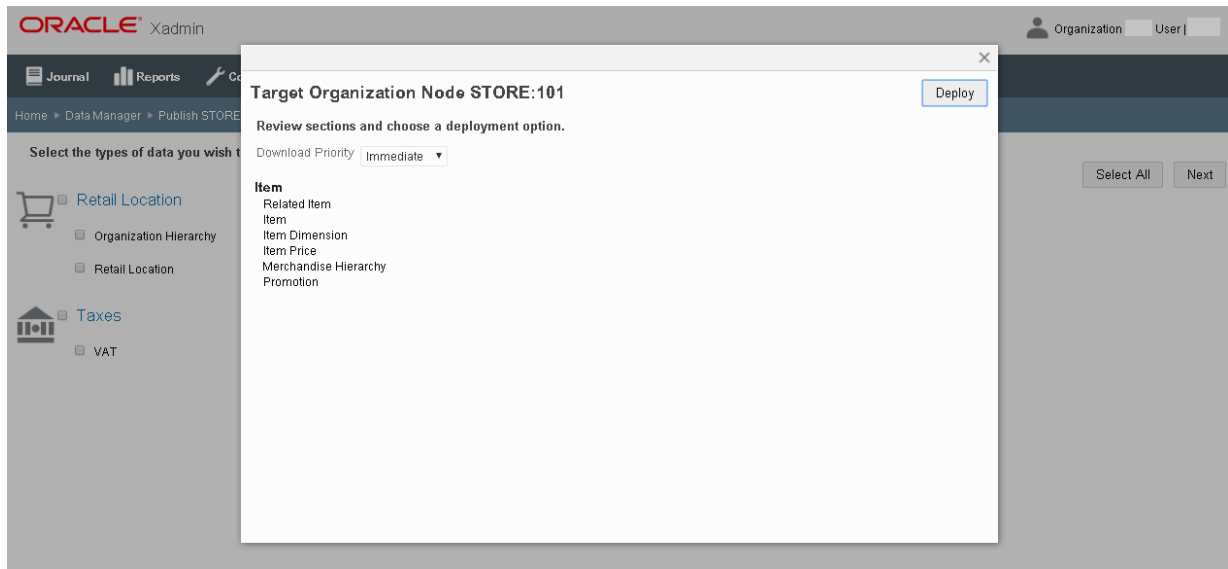
Figure 2–5 Xstore Office - Data to Publish



Note: The flow of Promotion data into and out of OCDS is not currently supported. The Promotion check-box in [Figure 2–5](#) should not be selected.

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

Figure 2-6 Xstore Office - Target Organization Node Pop Up



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

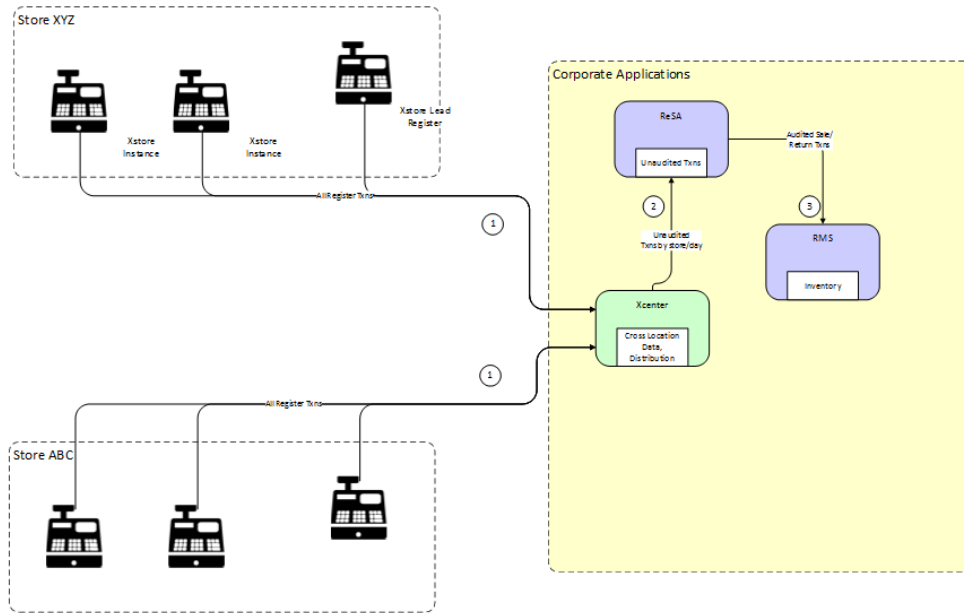
All transactions produced in Xstore are sent to ReSA. ReSA processing is primarily concerned with transactions that alter inventory or contain payment. ReSA loads other types of Xstore transactions (such as entering training mode, gift registry creation, and so on) into an OTHER transaction type for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

OCDS Integration Configuration

This chapter describes configuration options for the OCDS integration.

Xadmin-OCDS Integration Communication

Xadmin requests changes to merchandising and pricing data in OCDS by calling the REST web services.

[ON-PREMISES ONLY] The Xstore Office `xcenter.properties` file contains properties for the OCDS URL and service path prefix.

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

Note: The Integration Management feature only available for Xstore Office Cloud Service Release 18.1.

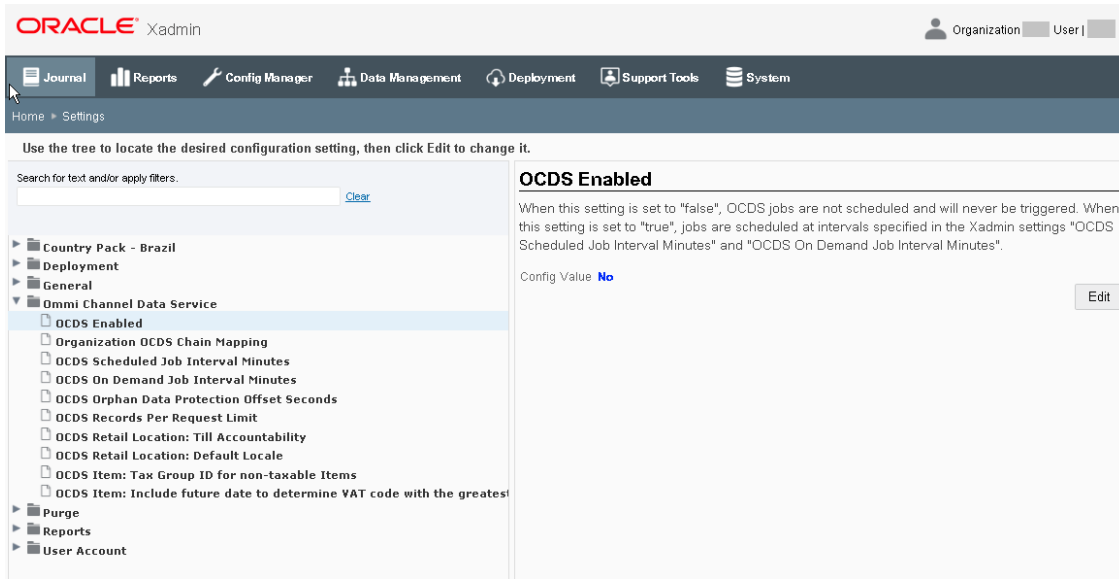
Table 4–1 *Properties - OCDS URL and Service Path Prefix*

Properties	Description
<code>ocds.connectionURL</code>	A URL describing protocol, host name, and port of the OCDS Web Services. Example: <code>https://HOSTNAME:xxx</code>
<code>ocds.connection.servicePath.prefix</code>	The root service path for OCDS. Example: <code>/ords/ocds/omnichannel/v1</code>
<code>ocds.connection.username</code>	The encrypted username used to authenticate web service communication with OCDS. This property is only necessary for on-prem installations, cloud installations store this information in the Oracle Enterprise Manager (OEM).
<code>ocds.connection.password</code>	The encrypted password used to authenticate web service communication with OCDS. This property is only necessary for on-prem installations, cloud installations store this information in the Oracle Enterprise Manager (OEM).

Xadmin System Settings

Xadmin's System Settings screen includes a category for Omnichannel Data Service (OCDS) configuration options. These settings govern how Xadmin integrates with OCDS.

Figure 4–1 Xstore Office - Omnichannel Data Service System Settings



Omnichannel Data Service Configuration Options

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

Table 4–2 OCDS System Settings

Setting	Description
OCDS Enabled	<p>When this setting is set to <code>false</code>, OCDS jobs are not scheduled and will never be triggered. When this setting is set to <code>true</code>, jobs are scheduled at intervals specified in the Xadmin settings "OCDS Scheduled Job Interval Minutes" and "OCDS On Demand Job Interval Minutes".</p> <p>Note: The OCDS Enabled setting is only available for Xstore Office and Xstore Office Cloud Service Release18.0.</p> <p>[CLOUD ONLY] For Xstore Office Cloud Service Release 18.1 this system setting is not available. The integration with OCDS is enabled through the Xadmin Integration Management flow.</p>
Organization OCDS Chain mapping	Specifies a comma separated Xstore organization and OCDS chain mappings. For example, "1000:1,2000:2" maps organization 1000 to chain 1, and organization 2000 to chain 2.
OCDS Scheduled Job Interval Minutes	This is the frequency with which Xadmin requests data from OCDS. The value should be greater than or equal to 30 minutes.
OCDS On Demand Job Interval Minutes	This is the maximum amount of time that will elapse between when an on-demand job is created and is executed. The value should be greater than or equal to 5 minutes.
OCDS Orphan Data Protection Offset Seconds	This is the minimum amount of time data must age in OCDS before it can be visible to Xcenter. The offset helps to prevent related-data from becoming orphaned due to system latency. The value should be greater than or equal to 30 seconds.
OCDS Records Per Request Limit	This specifies the number of records to request in calls to OCDS. If no limit is defined then OCDS will determine the maximum number of records.
OCDS Retail Location: Till Accountability	This specifies if Retail Locations created from OCDS data are to use Till Accountability or not.

Table 4–2 (Cont.) OCDS System Settings

Setting	Description
OCDS Retail Location: Default Locale	This specifies the Locale to use in Retail Locations created from OCDS data when a LANG_ISO_CODE is not defined for a location. OCDS Item: Tax Group ID for non-taxable items This specifies the Tax Group ID assigned to non-taxable items created from OCDS data.
OCDS Item: Tax Group ID for non-taxable Items	This specifies the Tax Group ID assigned to non-taxable items created from OCDS data.
OCDS Item: Include future date to determine VAT code with the greatest active date?	This specifies if future date should be included to determine VAT code with the greatest active date.

OCDS Database Tables

There are three database tables in the Xadmin database that are exclusively used for integration with OCDS. See the *Oracle® Retail Xstore Point-of-Service Software Database Dictionary* for complete details on these tables.

Table 4–3 OCDS Database Tables

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

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

Seed Data for VAT and Non-VAT Organizations

Seed data for VAT and non-VAT organizations

The OCDS_SUBTASK_DETAILS table contains metadata that controls how .mnt files are named and deployed for data-loading into Xstore Suite databases. In general, the out-of-the box seed data values for this table are appropriate for most organizations; however, if not using VAT then the ACTIVE column for the VAT SUBTASK_ID should be set to 0.

Figure 4–2 OCDS Subtask Details Table

ORGANIZATION_ID	SUBTASK_ID	FILENAME_PREFIX	QUERY_BY_CHAIN	QUERY_BY_ORG_NODE	DESTINATION	DOWNLOAD_TIME	APPLY_IMMEDIATELY	ACTIVE	FAMILY
1	CLEARANCE_PRICE_CHANGE	5_clrpc	0	1	ALL	IMMEDIATE		1	1 PRICE
1	DIMENSION_TYPE	2_dimensionType	0	0	ALL	STORE_CLOSE		0	1 DIMENSION
1	DIMENSION_VALUE	2_dimensionValue	0	0	ALL	STORE_CLOSE		0	1 DIMENSION
1	ENTERPRISE_ITEM	3_itemCorp	0	0	XCENTER_ONLY (null)			0	1 ITEM
1	ITEM	3_item	0	1	XSTORE_ONLY	STORE_CLOSE		0	1 ITEM
1	ITEM_LOC	4_itemLoc	0	1	ALL	STORE_CLOSE		0	1 ITEM
1	ITEM_NODE	4_itemNode	0	1	XCENTER_ONLY (null)			0	0 ITEM
1	MERCH_HIERARCHY	2_merchHier	0	0	ALL	STORE_CLOSE		0	1 MERCH_HIERARCHY
1	ORG_HIERARCHY	1_orgHier	1	0	ALL	STORE_CLOSE		0	1 ORG_HIERARCHY
1	PRICE	5_price	0	1	ALL	IMMEDIATE		1	1 PRICE
1	PROMOTION	5_promo	0	1	XSTORE_ONLY	STORE_CLOSE		0	1 PROMOTION
1	REGULAR_PRICE_CHANGE	5_respc	0	1	ALL	IMMEDIATE		1	1 PRICE
1	RELATED_ITEM	4_relatedItem	0	1	ALL	STORE_CLOSE		0	1 RELATED_ITEM
1	RETAIL_LOCATION	2_retailLoc	1	0	ALL	STORE_CLOSE		0	1 RETAIL_LOCATION
1	VAT	2_vat	0	0	ALL	STORE_CLOSE		0	1 VAT

Download: Immediate vs Store Close

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

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.

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, "Appendix: POSLog to RTLog Mapping Details"](#). It is critical that the mappings are complete. If additional valid values are configured for Xstore in the RTLogMappingConfig.xml, they must also be configured for 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 RMFCS should be made to be in synch, with any changes made post-implementation continuing to be made in both solutions. The entities in this category include:

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

See the [Appendix A, "Appendix: 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 RMFCS and Xstore, as RMFCS 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 RMFCS will also be used to load those rates into Xstore, in order to ensure both solutions are operating with the same information and to prevent a financial impact from occurring due to differences in the rates used. Tender exchange transactions that occur in Xstore, where a customer is given USD in exchange for CAD, will be mapped to the transaction type OTHER in Sales Audit.

Stores

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

Additionally, latitude and longitude information that is used by Xstore to determine nearby stores for its inventory lookup function are not available as part of the integration from RMFCS. 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 RMFCS - group, department, class, and subclass. In Merchandising, the class ID displayed to users is unique only when combined with its department ID. Similarly, the displayed subclass ID is only unique when combined with its department and class. However, instead of using the composite key in the integration with Xstore, the unique key that is held in the RMFCS tables for class and subclass used in the OCDS and is written into .mnt files. This unique ID is not visible to users of RMFCS.

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

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

The creation of the RMFCS item in Merchandising 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 RMFCS, 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 RMFCS, so these items will appear as standard items in Xstore and the component details will not be available.

Differentiators

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

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

Product Restrictions

Product restrictions can be set up in RMFCS 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

RMFCS 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 RMFCS 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 RMFCS are ignored.

Other Item Attribute Notes

- Item Restocking - unlike Xstore, RMFCS 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 RMFCS. Therefore, this function would not be available in Xstore.
- RMFCS 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 RMFCS, these are currently not used by Xstore.
- Translated item descriptions are available from RMFCS as part of the integration but are not currently used by Xstore. Xstore uses the item level description (which is communicated in the primary RMFCS 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 RMFCS be updated to show the localized item description.

Tax

This section describes considerations regarding taxes.

Value Added Tax (VAT)

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

When RMFCS 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 RMFCS 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

RMFCS 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 Merchandising 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 Merchandising .zip 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 Merchandising files, and is staged for store deployment with a deployment ID greater than the deployment ID of the Merchandising files.

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

1. Set up sales tax rules. To set up a simple rate based tax rule, use existing record types TAX_LOCATION, TAX_AUTHORITY, TAX_GROUP, TAX_GROUP_RULE, and TAX_RATE_RULE to populate tax tables tax_tax_loc, tax_tax_authority, tax_tax_group, tax_tax_group_rule, and tax_tax_rate_rule. For more details on tax rule configuration, see the TAXING section in the *Oracle Retail Xstore Point of Service Host Interface Guide* available on My Oracle Support.
2. Set up retail store and tax location mapping in table tax_rtl_loc_tax_mapping using existing record type TAX_RETAIL_LOCATION_MAPPING. For more details on this record type, see the TAXING section in the *Oracle Retail Xstore Point of Service Host Interface Guide* available on My Oracle Support.
3. ITEM_TAX_GROUP is used to update the item record in the itm_item_options table with sales tax group ID. This .mnt file has to be imported after the RMFCS 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 RMFCS. No inventory information is integrated between Xstore and RMFCS, 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 RMFCS. 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 RMFCS and Xstore is not supported.

Serialized Inventory

RMFCS 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 RMFCS, the last location to be processed by the integration code sets the item level serialized flag in Xstore.

Note: RMFCS 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 Sales Audit.

Recognition of a Sale

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

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 RMFCS and RPCS 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 RMFCS and RPCS)
- Manufacturer's Recommended Retail (from RMFCS)

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

Sales Audit

This section describes sale audit considerations.

Register-level Balancing

Xstore workstation and Sales Audit register are equivalent concepts; however Sales Audit does not have an entity equivalent to the Xstore till, which means that Xstore cannot be configured for till-level balancing when integrated with Sales Audit. When integrated with Xstore, Sales Audit should be configured with a balancing level of Register and Xstore will always send 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 Sales Audit only supports one. Therefore only the transaction level sales associate is exported to Sales Audit.

Tender Type

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

Coupons

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

Promotion

The OCDS_SUBTASK_DETAILS table contains metadata that controls what data is requested from OCDS by the Oracle Retail Xstore Suite. Since the flow of promotion data into and out of OCDS is not currently supported, the "PROMOTION" subtask must not be enabled in the database.

Important: The flow of Promotion data into and out of OCDS is not currently supported.

To prevent Xstore Office from requesting promotion data from OCDS, delete all records from the OCDS_SUBTASK_DETAILS table in the Xadmin schema where SUBTASK_ID = 'PROMOTION'.

RTLog Generator On-Premises

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:

- rtlog-generator-config-resa-cs.zip
- rtlog-generator-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.

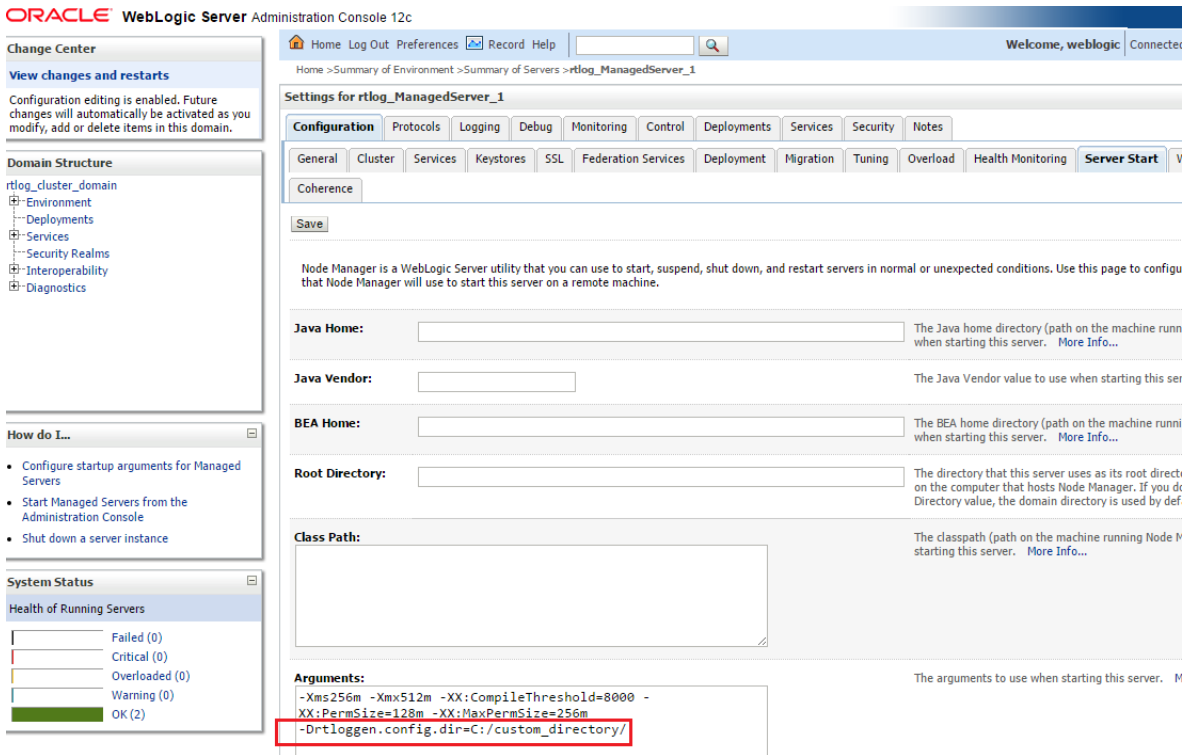
To integrate with ReSA on premise, rtlog-gen-config-resa-onprem.zip should be used to externally configure and extend the RTLog Generator's functionality.

To set up the external configuration features:

1. Extract the configuration file's content into the `C:\<rtlog-generator-config>` directory if installing on Microsoft Windows or `/usr/local/<rtlog-generator-config>` on Linux OS. These directories are the default locations where the RTLog Generator application will look for the configuration files. These default locations can be overridden/changed by using one of the following ways:
 - Pass a JVM argument to the server startup script and bounce the server:
`-Drtloggen.config.dir=C:/<custom_directory>/`

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

Figure 6-1 Administration Console Configuration Page



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

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

Figure 6-2 Example of context-param Field Update

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

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

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

- `rtlogconfig.properties`:

This file contains 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="P01" RecordValue="P01" />
  <VALUE_MAPPING sourceValue="P02" RecordValue="P02" />
  <VALUE_MAPPING sourceValue="P03" RecordValue="P03" />
  <VALUE_MAPPING sourceValue="P04" RecordValue="P04" />
  <VALUE_MAPPING sourceValue="P05" RecordValue="P05" />
  <VALUE_MAPPING sourceValue="SAMPLE" RecordValue="SAMPLE_VALUE" />
  </VALUE_MAPPINGS>
</MAP>

```

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>

```

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

Deployment

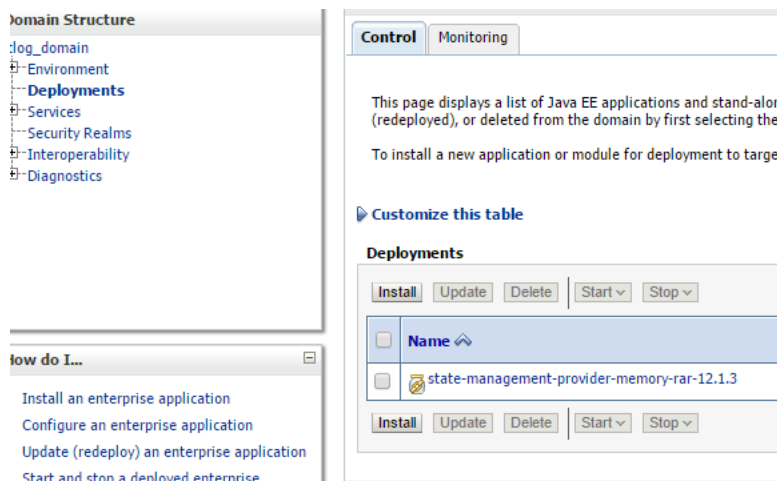
If you are deploying in a cluster, first set up a WebLogic cluster. For more information, see "[WebLogic Cluster Setup](#)".

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

To deploy the RTLog Generator application:

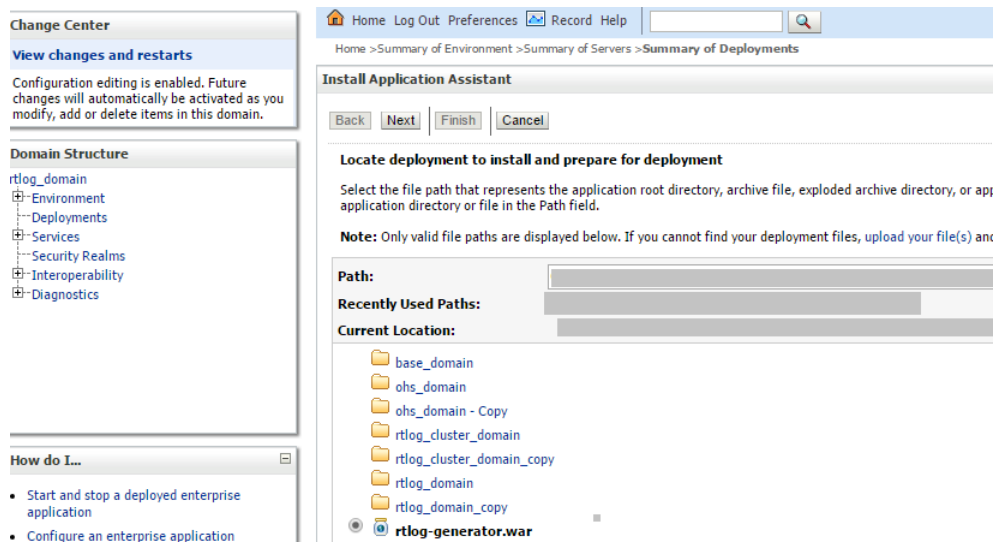
1. Log in to the WebLogic 12 Server Administration Console (http://<hostName>:<port>/console).
2. Click the Deployment link from the left navigation menu.
3. Click Install.

Figure 6–7 Administration Console Control Page



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

Figure 6–8 Administration Console Install Application Assistant Page



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

Figure 6–9 Administration Console Summary of Deployments

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

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:

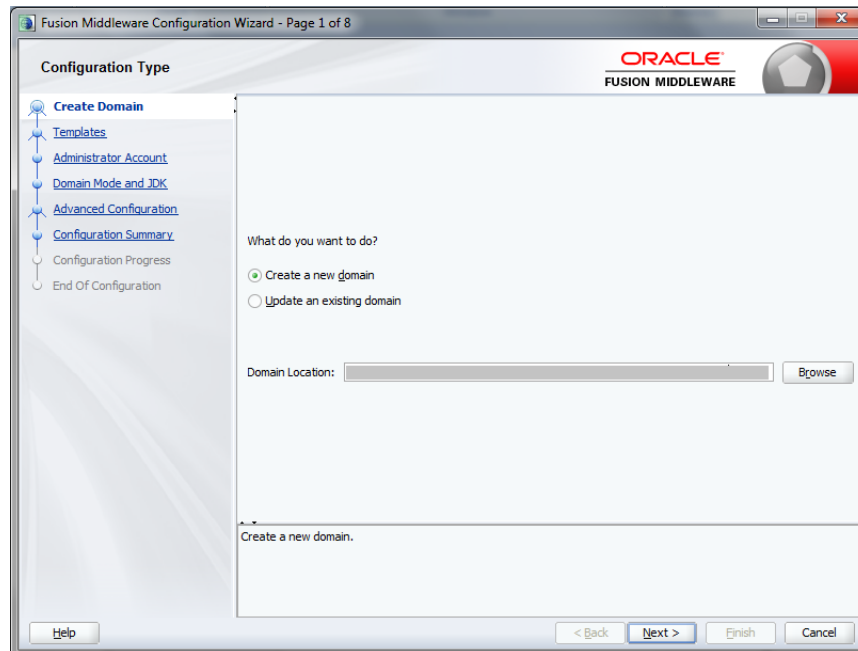
- To deploy on a cluster, see "[Deployment of the RTLog Generator Application on a Cluster](#)".
- To enable security for the RTLog Generator application, see "[Security Configuration](#)". When deploying in a non-clustered environment, continue at this section.

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.

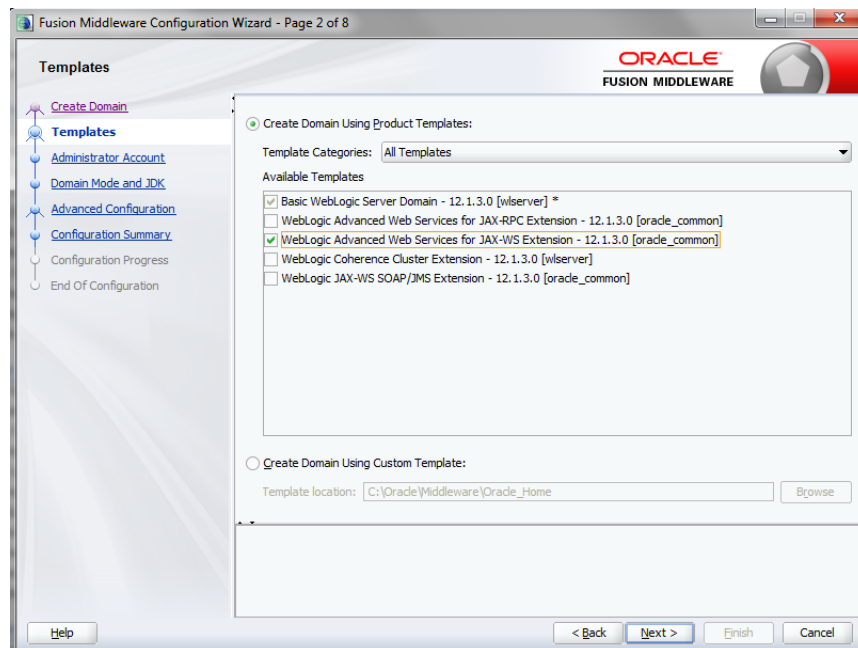
To set up the cluster to use RTLog Generator:

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

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

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

Fusion Middleware Configuration Wizard - Page 3 of 8

Administrator Account

ORACLE
FUSION MIDDLEWARE

[Create Domain](#)
[Templates](#)
Administrator Account
[Domain Mode and JDK](#)
[Advanced Configuration](#)
[Configuration Summary](#)
 Configuration Progress
 End Of Configuration

Name

Password

Confirm Password

Help < Back Next > Finish Cancel

- 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

Fusion Middleware Configuration Wizard - Page 4 of 8

Domain Mode and JDK

ORACLE
FUSION MIDDLEWARE

[Create Domain](#)
[Templates](#)
[Administrator Account](#)
Domain Mode and JDK
[Advanced Configuration](#)
[Configuration Summary](#)
 Configuration Progress
 End Of Configuration

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.

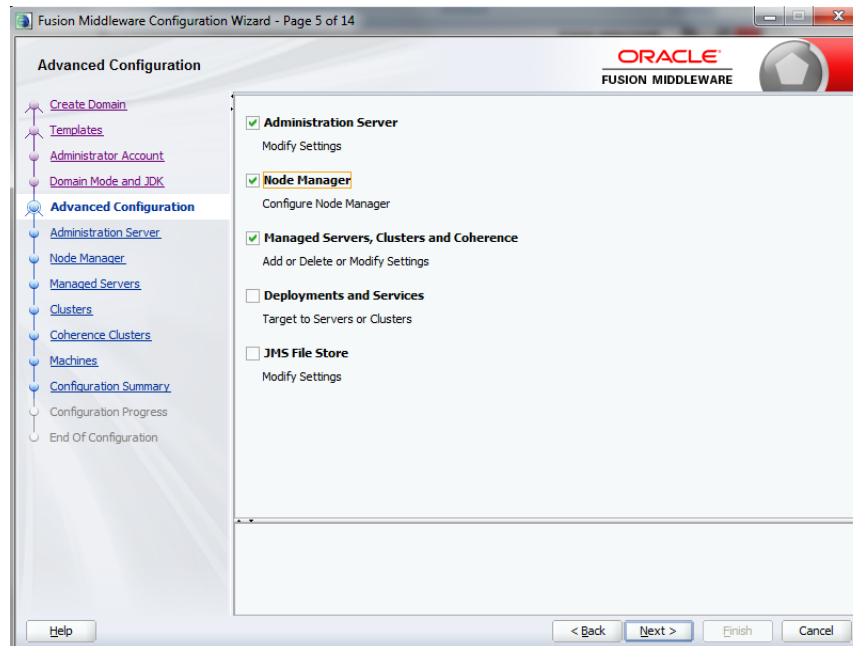
JDK

Oracle HotSpot:

Other JDK Location: Browse

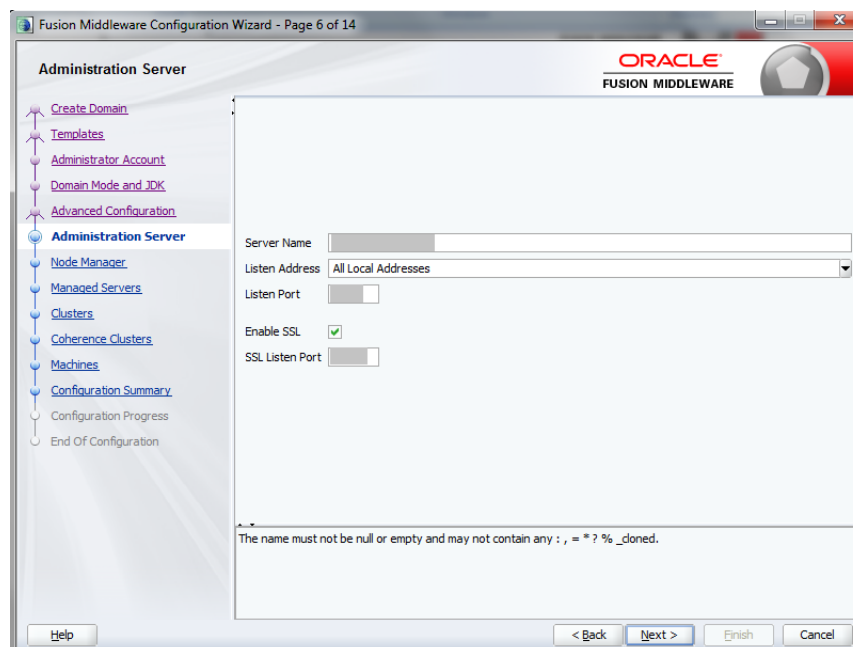
Help < Back Next > Finish Cancel

- 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

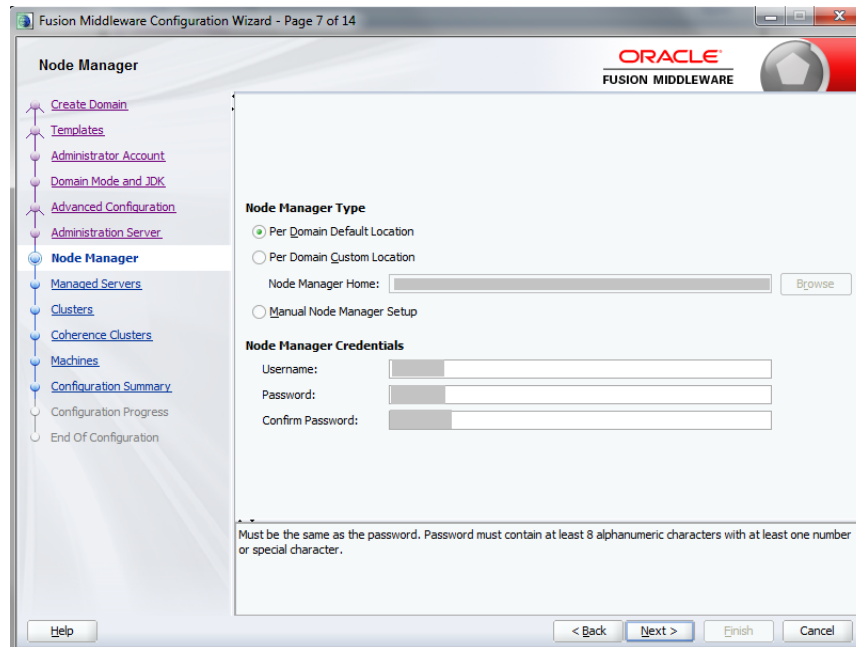
7. On the Administration Server page, enter the values to configure the administration server. The administrator server controls all the managed servers that are part of the cluster.

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**.

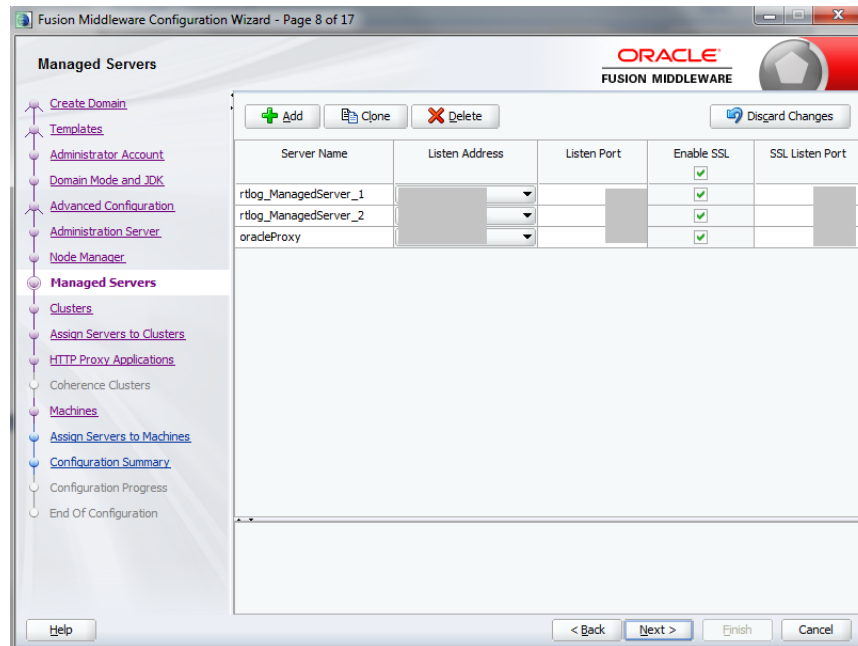
Figure 6–15 Configuration Wizard Administration Server Page

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**.

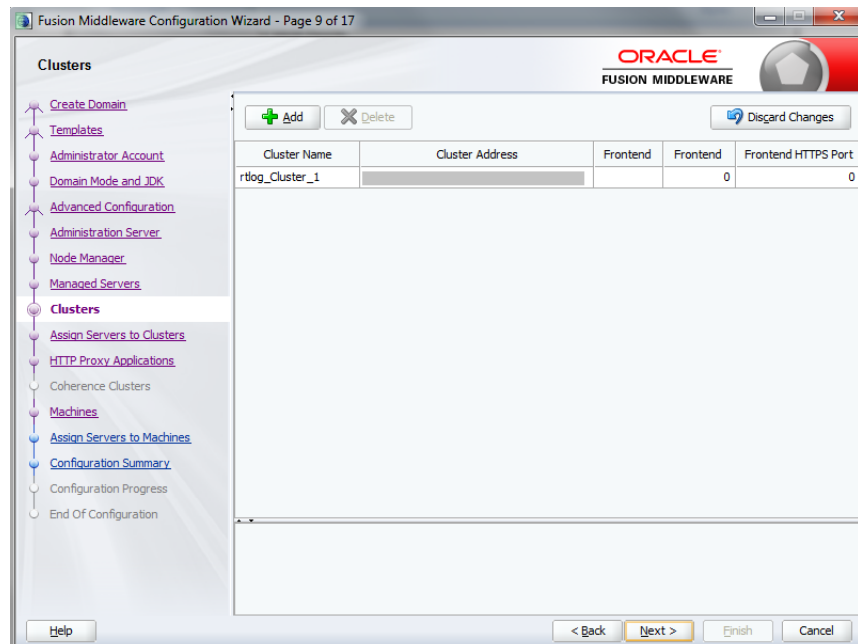
Figure 6–16 Configuration Wizard Node Manager Page



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

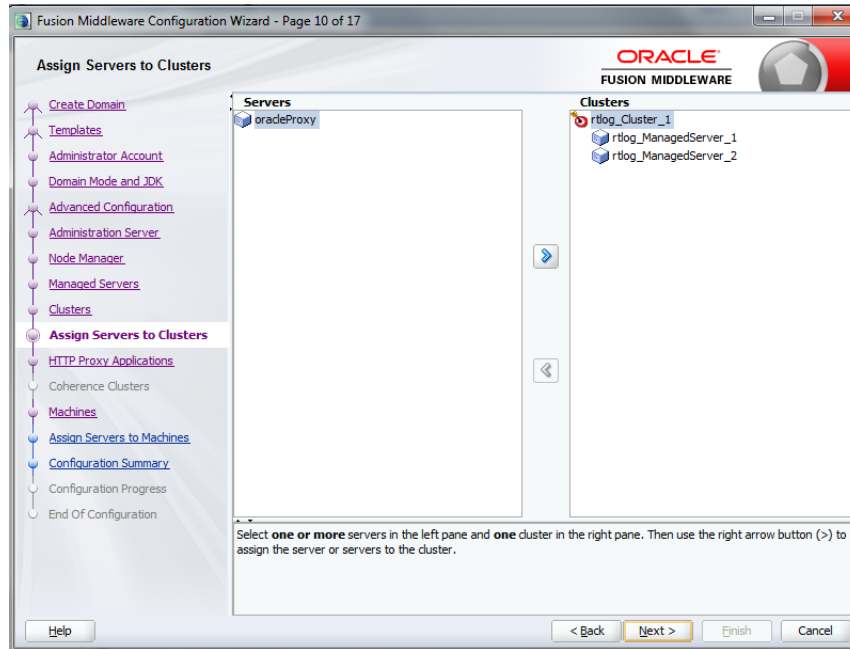
- On the Clusters page, add and configure the cluster. Enter the cluster name followed by the cluster address, that is, IP address1:port1, IP address2:port2, so on. Click Next.

Figure 6–18 Configuration Wizard Clusters Page

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

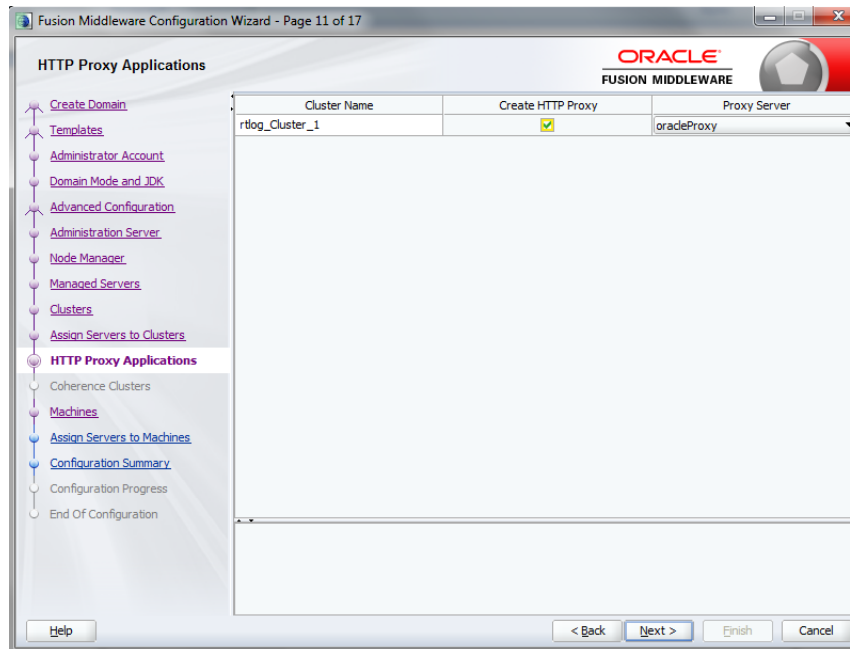
Note: Do not include the Oracle Proxy as part of the cluster.

Figure 6–19 Configuration Wizard Assign Servers to Clusters Page

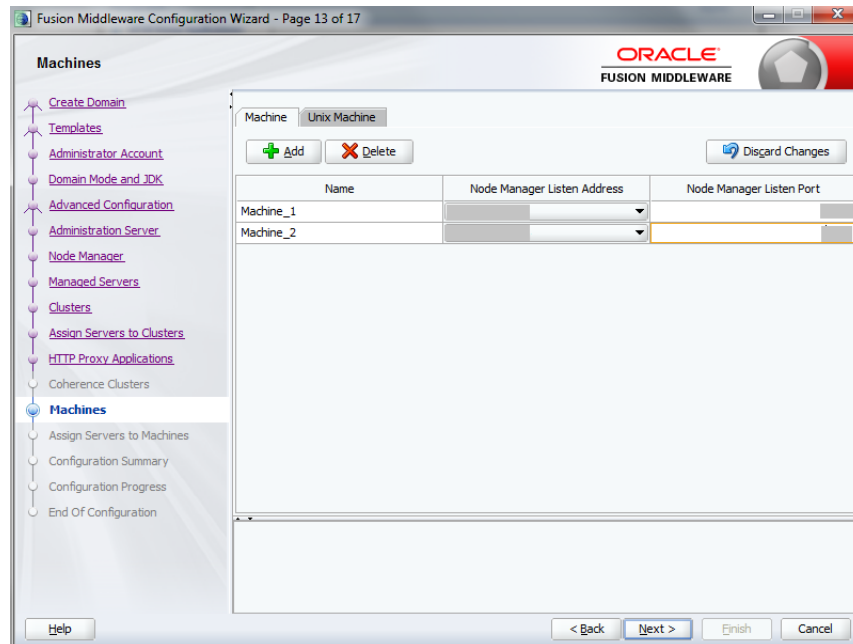


- On the HTTP Proxy Applications page, select Create HTTP Proxy and then select the server from the drop-down list. By default, it should have already been selected. Click Next.

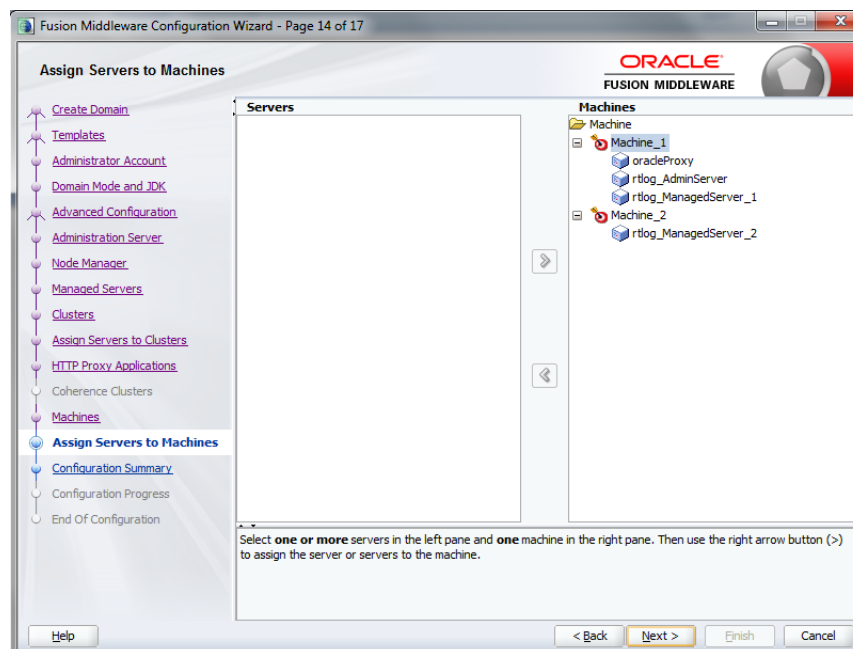
Figure 6–20 Configuration Wizard HTTP Proxy Applications Page



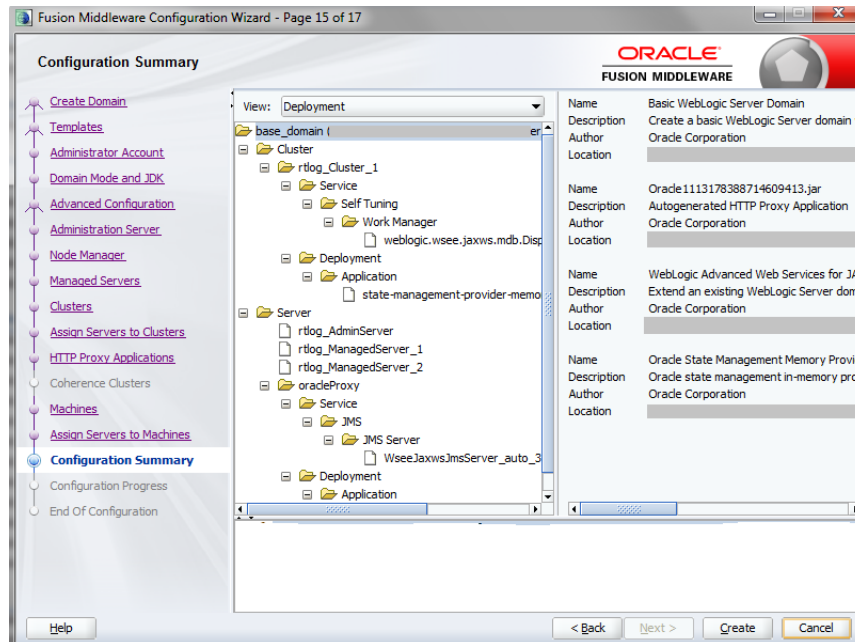
- On the Machines page, add and configure each machine. To add Machine_1 and Machine_2, click **Add** and enter the respective IP addresses. This configuration is for setting up the Node managers on both the machines. Since these node managers are physically separated, you can select the same host. Click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

Figure 6–21 Configuration Wizard Machines Page

- 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

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

Figure 6–23 Configuration Wizard Configuration Summary Page

To complete the configuration of the cluster:

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

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

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

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

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

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

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

5. Pack the created domain:

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

```
<WL_HOME>\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')
```

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

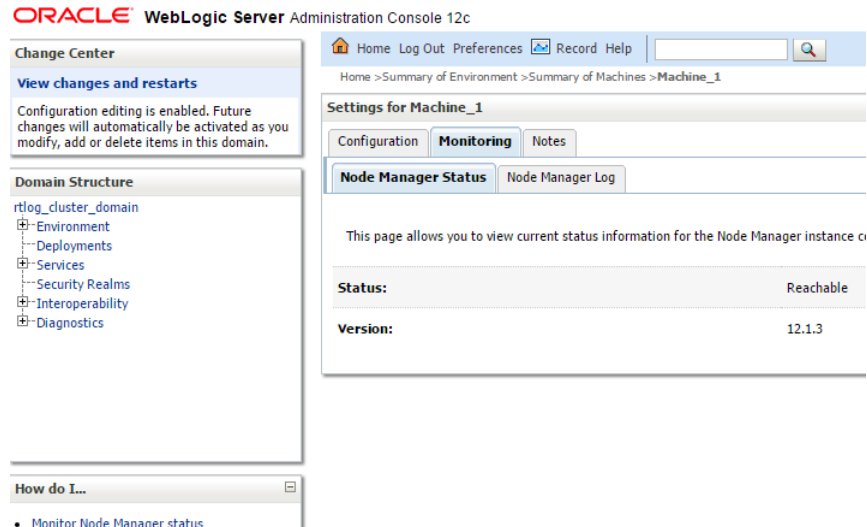
```
nmEnroll ('<WL_HOME>/user_projects/domains/<rtlog_cluster_
domain>','<WL_HOME>/user_projects/domains/<rtlog_cluster_
domain>/nodemanager')
```

- 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.

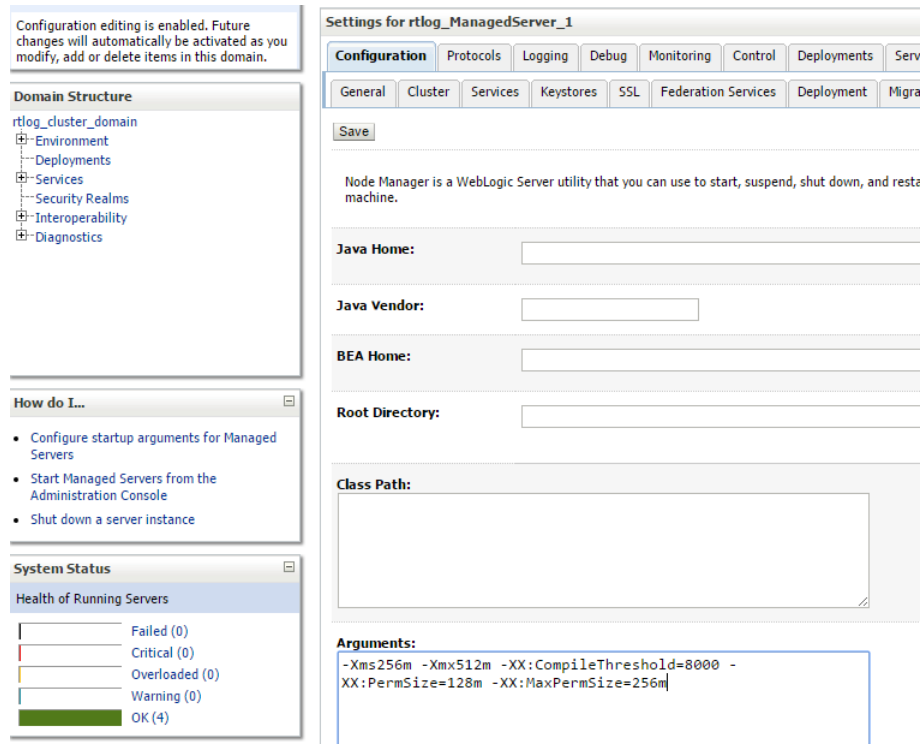
Figure 6–24 Administration Console Settings Page



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



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	
rtlog_AdminServer(admin)	Configured		Machine_1	RUNNING	OK	
rtlog_ManagedServer_1	Configured	rtlog_Cluster_1	Machine_1	RUNNING	OK	
rtlog_ManagedServer_2	Configured	rtlog_Cluster_1	Machine_2	RUNNING	OK	

Deployment of the RTLog Generator Application on a Cluster

To deploy the application:

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

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

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

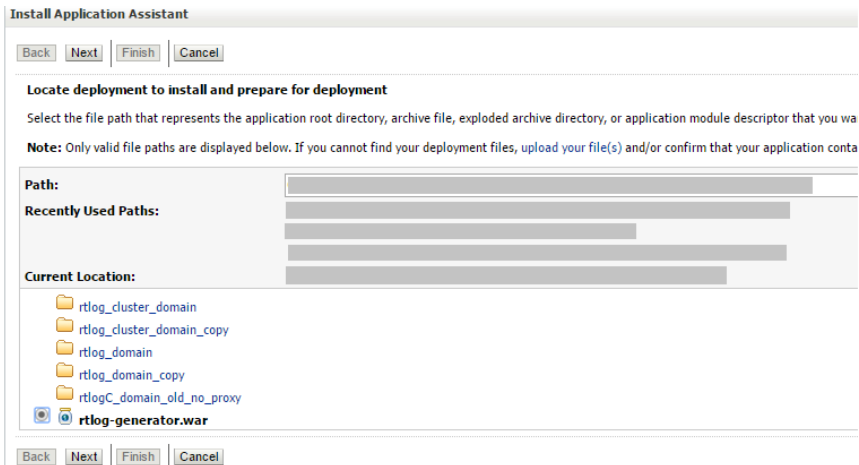
2. Navigate to the Administration Console home page and click Deployments in the left navigation menu. [Figure 6–27](#) shows an example of the page before deploying the RTLog Generator application.

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

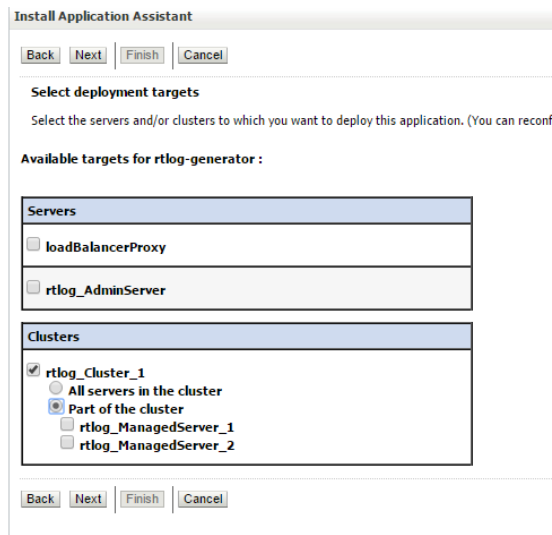
3. Click **Install**. The Install Application Assistant page appears. Select the path to the RTLog Generator WAR directory. Select the rtlog-generator.war option. Click **Next**.

Figure 6–28 Administration Console Install Application Assistant Page



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

Figure 6–29 Install Application Assistant Select Deployment Targets Page



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

Control Monitoring

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, or 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.

[Customize this table](#)

Deployments

Install Update Delete Start Stop

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

Install Update Delete Start Stop

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

ORACLE WebLogic Server Administration Console 12c

Home Log Out Preferences Record Help Welcome,

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

Messages

✓ All changes have been activated. No restarts are necessary.
 ✓ Settings updated successfully.

Settings for rtlog_cluster_domain

Configuration Monitoring Control Security Web Service Security Notes

General JTA JPA EJBs **Web Applications** Logging Log Filters

Save

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 ri "a*" is a compact syntax for indicating all previous releases, role-name = "a*" was t the realm. This parameter is one require 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 t backward compatibility flag. Until versio

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

Figure 6–32 WebLogic Plugin Enabled Parameter



Security Configuration

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

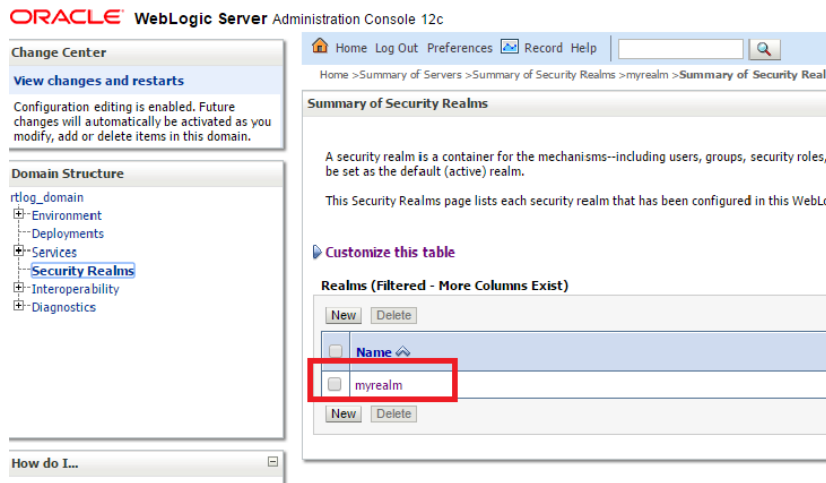
- Container level security: Basic HTTP authentication by setting up the security realm in WebLogic. To configure this security, see "[Container Level Security](#)".
- Transport level security: SOAP requests are sent over the secured protocol (HTTPS) by configuring the keystore/truststore in the WebLogic 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



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

ORACLE WebLogic Server Administration Console 12c

Home Log Out Preferences Record Help

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

Create a New Group

OK Cancel

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:

How would you like to describe the new Group?

Description:

Please choose a provider for the group.

Provider:

OK Cancel

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

ORACLE WebLogic Server Administration Console 12c

Home Log Out Preferences Record Help

Home > Summary of Servers > Summary of Security Realms > myrealm > Summary of Security Realms > myrealm > Users and Groups > rtloguser > Users and Groups :

Create a New User

OK Cancel

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:

How would you like to describe the new User?

Description:

Please choose a provider for the user.

Provider:

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

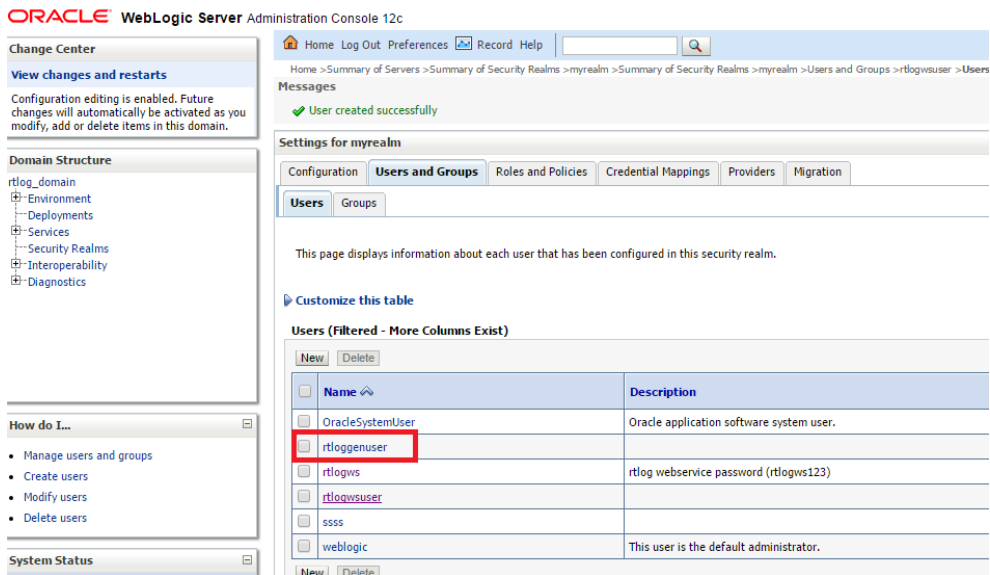
* Password:

* Confirm Password:

OK Cancel

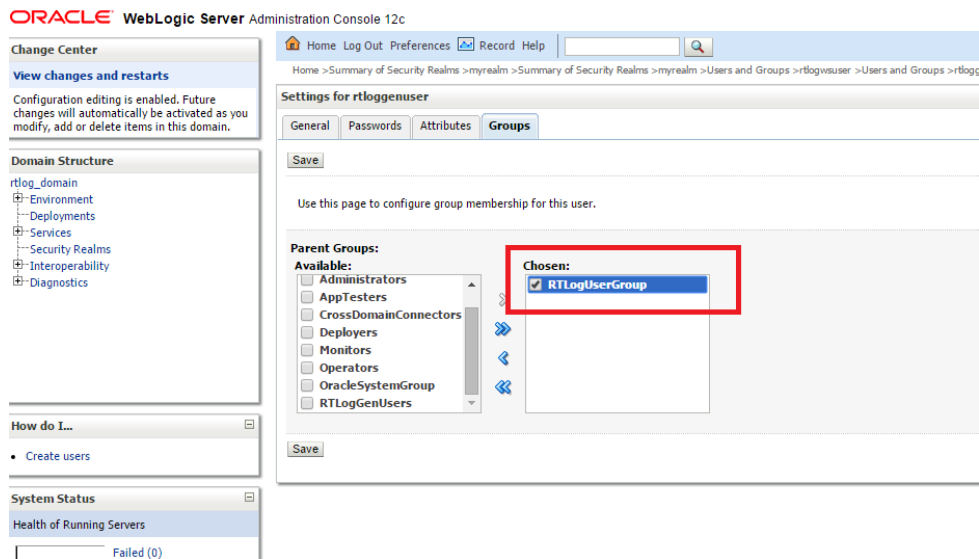
6. In the list of users, click the newly created user.

Figure 6–36 Users Page



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

Figure 6–37 User Settings Page



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, to use the same truststore file.

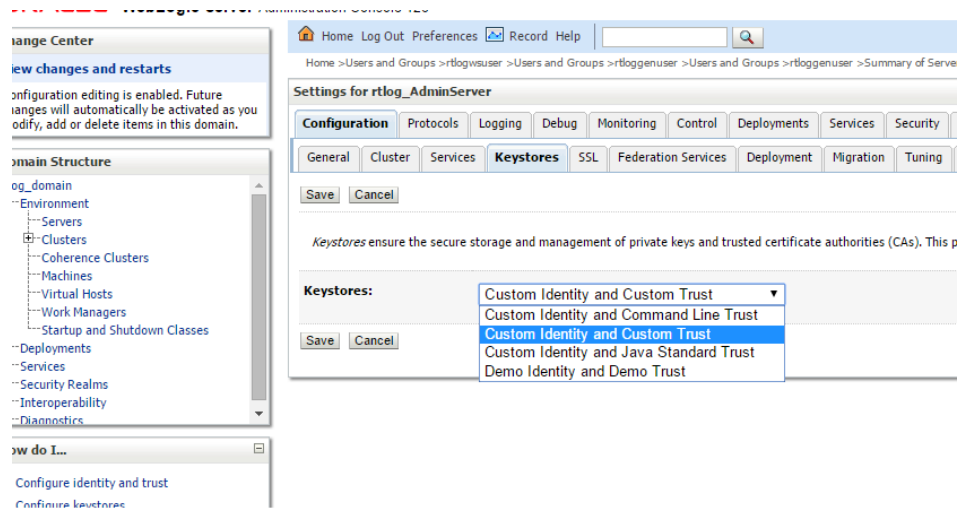
3. Log in to the WebLogic console. Click Environment and then the Servers link from the left navigations menu.

Figure 6–39 Administration Console Servers Page

The screenshot shows the Oracle WebLogic Server Administration Console. The left-hand navigation pane is expanded to 'Environment' > 'Servers'. The main content area is titled 'Summary of Servers' and includes a 'Configuration' tab. Below the title, there is a table of servers. The table has a header row with 'Name' and a dropdown arrow, and a row for 'rtlog_AdminServer(admin)'. Below the table are 'New', 'Clone', and 'Delete' buttons. The console also shows a 'Domain Structure' tree on the left and a 'Change Center' at the top left.

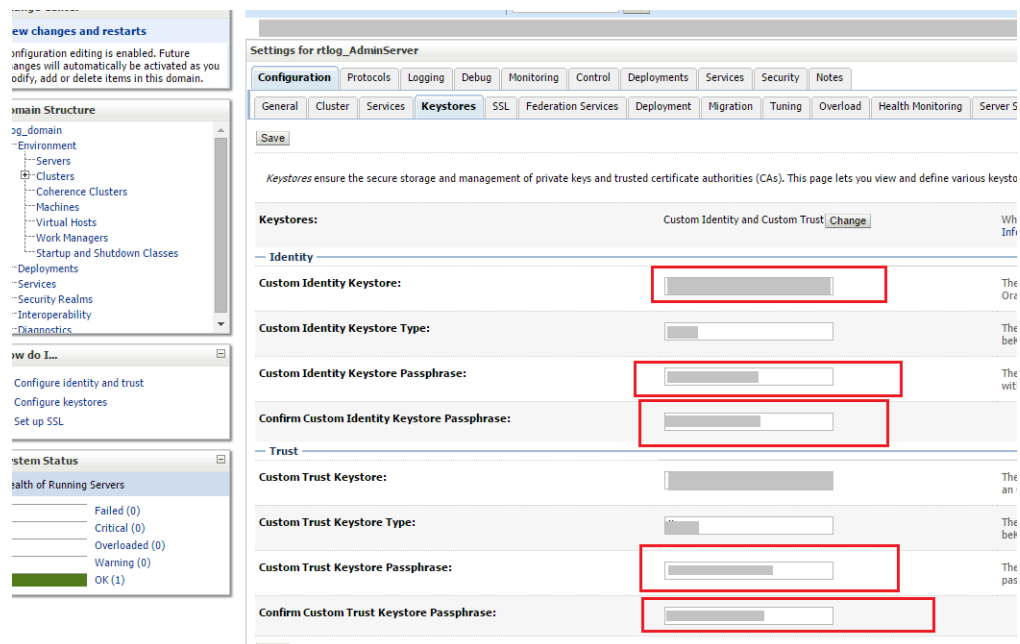
4. Click **Change**. Select Custom Identity and Custom Trust. Click **Save**.

Figure 6–40 Keystores Settings



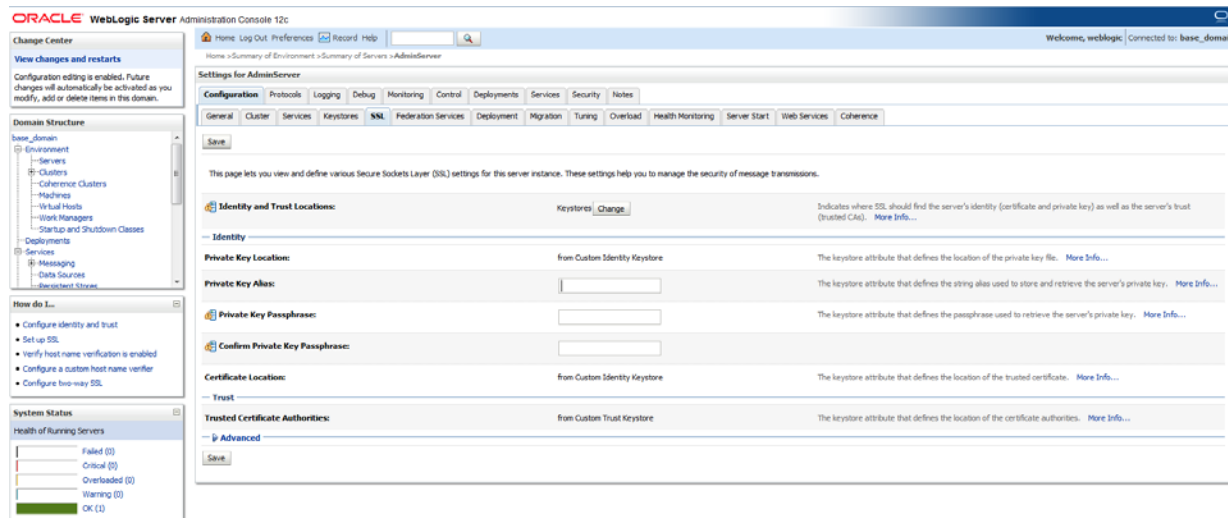
5. Click the linked name for the Administration Server. The page containing the settings for the Administration Server appears. Select the Keystores tab.

Figure 6–41 Settings for the Administration Server



6. Enter the path to keystore.jks, including the file name, and enter the custom Identity Keystore passphrase you created for the keystore. Repeat this for 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



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:

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

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


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

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

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

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

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

- Configure `endpointAddress` of the `ReSA_Broadcaster_jaxws` bean.

RTLog Generator Cloud

This chapter describes the RTLog Generator on cloud.

RTLog Generator Cloud

This chapter describes how to 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 on-premises or with RMFCS 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 [Configuration](#) section in [Chapter 6, "RTLog Generator On-Premises"](#) and the *Retail Xstore - RTLog Generator Extension Guidelines (Doc ID 2174095.1)* on <https://support.oracle.com>.

Integration

This section describes the RTLog Generator Cloud integration.

Updating Mapping Configuration

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

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

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 = DIR/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 <code>RTLogMappingConfig.xml</code> file. If the customer has not uploaded a customized configuration xml file yet, provides a copy of the default mapping configuration XML file that is provided with the deployment.
PUT	OAuth2	application/json	Customer submits the updated <code>RTLogMappingConfig.xml</code> 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 <code>RTLogMappingConfig.xml</code> file active yet, HTTP 204 status is returned. The default <code>RTLogMappingConfig.xml</code> that is part of the deployment will resume being the active mapping configuration.

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

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

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

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

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

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

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

RMFCS on Cloud:

- SFTP credentials to connect to the RMFCS 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.

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 7-3 Request IDCS OAuth2 Token - OAuth2 Token Request

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

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

Example 7-4 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 7-5 Provide OAuth2 Token - Provide OAuth2 Token for REST Services

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

Appendix: 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
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

