

Oracle® Retail Merchandising and WMS Cloud

Implementation Guide



G42860-01
September 2025



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, software documentation, data (as defined in the Federal Acquisition Regulation), 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 embedded, installed, or activated on delivered hardware, and modifications of such programs) and Oracle computer documentation or other Oracle data delivered to or accessed by U.S. Government end users are "commercial computer software," "commercial computer software documentation," or "limited rights data" pursuant to the applicable Federal Acquisition Regulation and agency-specific supplemental regulations. As such, the use, reproduction, duplication, release, display, disclosure, modification, preparation of derivative works, and/or adaptation of i) Oracle programs (including any operating system, integrated software, any programs embedded, installed, or activated on delivered hardware, and modifications of such programs), ii) Oracle computer documentation and/or iii) other Oracle data, is subject to the rights and limitations specified in the license contained in the applicable contract. The terms governing the U.S. Government's use of Oracle cloud services are defined by the applicable contract for such services. 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®, Java, MySQL, and NetSuite are registered trademarks of Oracle and/or its affiliates. Other names may be trademarks of their respective owners.

Intel and Intel Inside 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, Epyc, and the AMD 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.

Contents

Send Us Your Comments

Preface

| | |
|---|----|
| Audience | i |
| Customer Support | i |
| Improved Process for Oracle Retail Documentation Corrections | i |
| Oracle Retail Documentation on the Oracle Help Center (docs.oracle.com) | ii |
| Conventions | ii |

1 Introduction

2 Integration Architecture

| | |
|--------------------------|---|
| Domain Value Maps (DVMs) | 2 |
|--------------------------|---|

3 Integration Error Handling

| | |
|--------------------------------|---|
| Universal Service Mapper (USM) | 2 |
|--------------------------------|---|

4 Data Conversion

| | |
|--------------------------------|---|
| Manual Conversion | 1 |
| Order Types | 1 |
| Stock Orders | 1 |
| Customer Orders | 1 |
| Customer Orders (Store Pickup) | 2 |
| Return to Vendor Request | 2 |
| Reason Codes | 2 |
| Customer Order Returns | 2 |
| Receiver Unit Adjustments | 3 |
| Lock Codes | 3 |
| Convert Master Data | 5 |

| | |
|----------------------------------|----|
| Stores, Partners, and Warehouses | 5 |
| Stores | 6 |
| External Finishers | 6 |
| Warehouses | 7 |
| Vendors | 7 |
| Items | 8 |
| Item Header | 8 |
| Item Barcodes | 9 |
| Item Pre-packs | 10 |
| Item Supplier Pack Size | 10 |
| Item Dimensions | 10 |
| Convert Transactional Data | 11 |
| Purchase Orders | 12 |
| Inventory | 13 |

5 Integration Configuration

| | |
|-------------------------------|----|
| Schedule Jobs | 1 |
| Enabled Auto Send | 1 |
| Configure Outbound Interfaces | 1 |
| Inventory History Export | 2 |
| Outbound Loads Export | 4 |
| Outbound Manifest | 6 |
| Other Configurations | 8 |
| Sequence Counters | 8 |
| BOL Number | 8 |
| Inbound Shipment Number | 8 |
| Load and Parcel Manifests | 9 |
| IB Shipment Types | 9 |
| Company Configurations | 9 |
| Decimal Support | 9 |
| Facility Configurations | 11 |
| RF Configuration | 13 |
| Create LPN | 13 |
| Receive LPN | 14 |
| Configure Integration Layer | 15 |
| Configure USM | 15 |
| Configure Static DVMs | 15 |
| Configure RIB | 15 |
| Configure Merchandising | 16 |
| Configure OMS | 16 |

A Appendix - Solution Overviews

| | |
|---|-----|
| Oracle Warehouse Management Cloud | A-1 |
| Retail Integration | A-1 |
| Merchandising | A-1 |
| Retail Store Inventory and Operations Cloud Service | A-1 |
| Retail Order Broker Cloud Service | A-1 |

B Appendix - Outbound and Inbound Interfaces

| | |
|--|-----|
| Outbound from WMS Cloud | B-1 |
| ASN Outbound | B-1 |
| WMS Cloud Integrations | B-1 |
| ASN Out (ASNOutCre) | B-1 |
| Inventory Adjustments | B-2 |
| WMS Cloud Integrations | B-2 |
| Inventory Adjustment (InvAdjustCre) | B-2 |
| Return to Vendor Shipments | B-2 |
| WMS Cloud Integrations | B-2 |
| RTV (RTVCre) | B-2 |
| Receiving | B-3 |
| WMS Cloud Integrations | B-3 |
| Receipt (ReceiptMod) | B-3 |
| Stock Order Status | B-3 |
| WMS Integrations | B-3 |
| SO Status (SOStatusCre) | B-3 |
| Inbound to WMS Cloud | B-4 |
| Store | B-4 |
| WMS Cloud Integrations | B-4 |
| Create and Modify (StoreCre) | B-4 |
| Delete (StoreDel) | B-4 |
| Vendor | B-5 |
| WMS Cloud Integrations | B-5 |
| Vendor Create (VendorCre) and Modify (VendorFulRep) | B-5 |
| Vendor Delete (VendorDel) | B-5 |
| External Finisher | B-5 |
| WMS Cloud Integrations | B-5 |
| Partner Create (PartnerCre PartnerDtlCre) and Modify (PartnerDtlMod, PartnerMod) | B-6 |
| Partner Delete (PartnerDel) | B-6 |
| Partner Details Delete (PartnerDtlDel) | B-6 |
| Items | B-6 |
| WMS Cloud Integrations | B-6 |

| | |
|---|-----|
| Item Create (ItemCre, ItemUpcCre, ItemBomCre) and Modify (ItemUpcMod, ItemBomMod, ItemFulRep) | B-7 |
| ASN Inbound | B-7 |
| WMS Cloud Integrations | B-7 |
| Create (ASNInCre or Modify (ASNInMod) | B-7 |
| Delete (ASNInDel) | B-7 |
| Order | B-7 |
| WMS Cloud Integrations | B-8 |
| PO Create (POCre) and Modify (POFulRep) | B-8 |
| PO Delete (PODel) | B-8 |
| Stock Order | B-8 |
| WMS Cloud Integrations | B-8 |
| Stock Order Create (SOCre) or Modify (SOFulRep) | B-9 |
| Stock Order Header Delete (SOHdrDel) | B-9 |
| RTV Requests | B-9 |
| WMS Cloud Integrations | B-9 |
| Create RTV Request (RTVReqCre) or Modify (RTVReqFulRep) | B-9 |
| Delete RTV Request (RTVReqDel) | B-9 |

C Appendix - Activity Codes

| | |
|------------------------|-----|
| Activity Code Mappings | C-1 |
|------------------------|-----|

D Appendix - Known Issues

E Appendix - Other Resources

| | |
|--------------------------------|-----|
| WMS Cloud | E-1 |
| Oracle Retail Integration | E-1 |
| Oracle Retail Merchandising | E-1 |
| Other Documentation References | E-1 |

Send Us Your Comments

Oracle Retail and WMS Cloud 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 Online Documentation available on the Oracle Technology Network Web site. 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 <http://www.oracle.com>.

Preface

This guide provides detailed information that is important when integrating between Oracle Warehouse Management Cloud and Oracle Retail solutions.

Audience

This guide is intended for administrators.

This guide describes the administration tasks for Oracle Retail Integration Cloud Services.

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

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.

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 Help Center (docs.oracle.com)

Oracle Retail product documentation is also available on the following Web site:

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

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

Conventions

The following text conventions are used in this document:

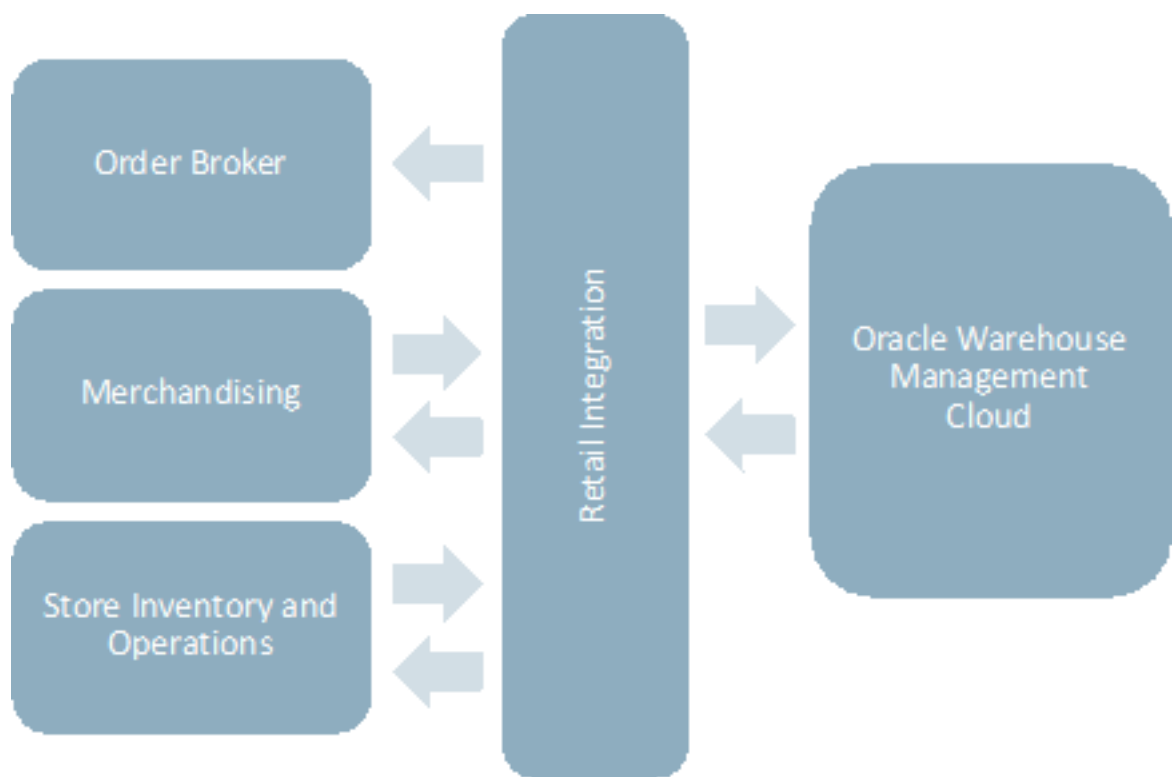
| Convention | Meaning |
|------------------------|--|
| boldface | Boldface type indicates graphical user interface elements associated with an action, or terms defined in text or the glossary. |
| <i>italic</i> | Italic type indicates book titles, emphasis, or placeholder variables for which you supply particular values. |
| <code>monospace</code> | Monospace type indicates commands within a paragraph, URLs, code in examples, text that appears on the screen, or text that you enter. |

1

Introduction

This document will provide an overview of the integration between Oracle Warehouse Management Cloud and Oracle Retail solutions. The Oracle Retail solutions included in this integration are primarily Oracle Retail Integration¹, and Oracle Retail Merchandising². However, there are integration points with Oracle Retail Store Inventory and Operations Cloud Service (SIOCS)³ and Oracle Retail Order Broker (OROB) as well.

Figure 1-1 Integration Points



The integration between these solutions has been built using pre-existing integrations and using mapping tools that are part of the Retail Integration component known as RIB (Retail Integration Bus). Messages from Merchandising and SIOCS are published to the RIB and then transformed into the format needed by WMS Cloud. The mapping tool in the integration layer is responsible for calling the WMS Cloud REST APIs to deliver the information.

Similarly, when WMS Cloud sends information to the Oracle Retail solutions, the WMS Cloud outbound interface is configured to work with an exposed SOAP API in the integration layer which transforms the data into the format needed by Merchandising and places it on the queue to be picked up by the appropriate RIB subscriber using a REST API.

¹ This refers to both the cloud service and on-premise versions of the retail integration solutions.

² This integration is supported in on premise and cloud versions of Merchandising (RMS and RMFCS).

³ This integration is also supported in SIM on premise for version 16.0.2.x.

Merchandising and WMS Cloud Integrations

Table 1-1 Merchandising and WMS Cloud Integrations

| From Merchandising to WMS Cloud | From WMS Cloud to Merchandising |
|--|---------------------------------|
| Items | Outbound ASNs |
| Stores | Inventory Adjustments |
| Partners | RTV Shipments |
| Suppliers | Stock Order Status Updates |
| Purchase orders | Outbound ASNs |
| Stock Orders (Transfers and Allocations) | |
| RTV Requests | |

SIOCS and WMS Cloud Integrations

Table 1-2 SIOCS and WMS Cloud Integrations

| From SIOCS to WMS Cloud | From WMS Cloud to SIOCS |
|-------------------------|-------------------------|
| Outbound ASNs | Outbound ASNs |

OROB and WMS Cloud Integrations

Table 1-3 OROB and WMS Cloud Integrations

| From OROB to WMS Cloud | From WMS Cloud to OROB |
|-----------------------------|----------------------------|
| Fulfillment Order Revisions | Stock Order Status Updates |
| | Receipts |

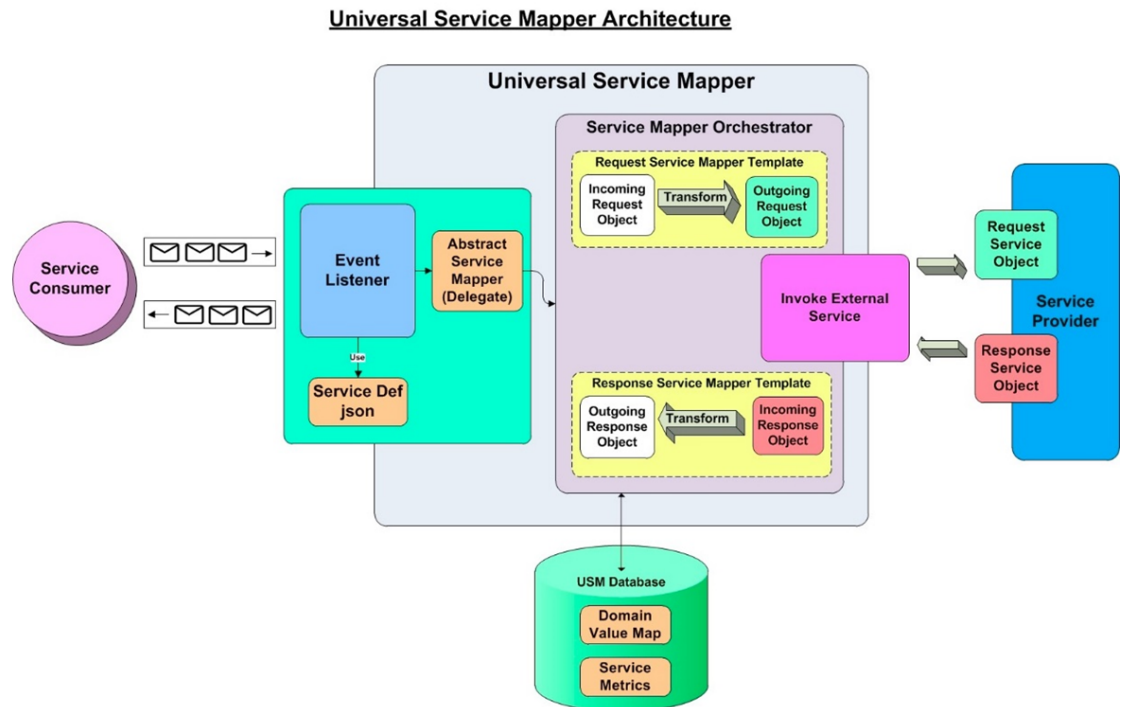
WMS Cloud also subscribes to inbound ASNs that are received from suppliers with shipment details for purchase orders. These are also brokered through RIB, but the source for these transactions is the supplier.

2

Integration Architecture

This chapter explains the different components that are used to make the whole integration work seamlessly.

Figure 2-1 Universal Service Mapper Architecture



The component that is used to do the mapping needed for this integration is referred to as the universal service mapper (USM). It has three major components:

- Event Listener
- Service Mapper Orchestrator
- External Service Invocation and Service Provider

Event Listener

The Event Listener is a service which is open to receiving data from the application that is connected to it. For the integration covered in this document, the two applications are a RIB component called RIB-LGF and WMS Cloud. The applications have the following URL pattern set in their target for USM: `http://<host>:<port>`.

When the application sends data, the Event Listener internally calls the Abstract Service Mapper, which determines the family, message type and the operations from the message received by referring to the Service Definition JSON files. These files store the crucial data required for the communication of the participating applications, including the host URLs of the source and destination applications along with usernames and passwords, if applicable.

Service Map Orchestrator

The Abstract Service Mapper in turn calls the Service Map Orchestrator which decides what data is to be populated in the mapper templates. The orchestrator does the mapping, field by field, from the source application to destination application.

The actual mapping logic is contained in orchestration files. All fields are mapped using a one-to-one mapping. Fields not required by the destination application are simply dropped. If a field is not present in the source data, then it is mapped using a pre-determined default value or left as null if allowed.

Certain key-value pairs are also managed in the integration layer in order to maintain context between the applications. See [Domain Value Maps \(DVMs\)](#) for more information on this configuration.

Service Provider and External Services

The Service Map Orchestrator component calls the services hosted by the service providers (RIB-LGF or WMS Cloud) after the mapping operations are completed. The service provider consumes these REST service calls via USM. USM holds the information necessary for it to call these services in a file.

Templates

Templates files hold the actual mapping information and use an xml format. There are three different types of templates used:

- Request Templates – used when the source application sends a message that has to be mapped to the destination application format.
- Response Templates – the result of the mapping that has been performed on the source application.
- Failure Templates – contain error codes and messages due to issues incurred in the mapping process, such as missing data or unexpected server events.

These templates are visible in the USM user interface to help manage errors that may occur in the integration.

Domain Value Maps (DVMs)

Domain Value Maps are tables that are present in the integration layer and contain defaults and cross references used in the mapping between applications and assist in transforming the messages from one format to the other.

There are two types of DVMs - static and dynamic. Static DVMs are used for foundational or seed data that doesn't change on a regular basis. Dynamic DVMs are built during runtime and are usually associated with transactional data, such as purchase orders.

For more information on all these concepts and the overall architecture, see the *Oracle Retail Integration Cloud Service Universal Service Mapper User Guide*.

3

Integration Error Handling

WMS Cloud

WMS Cloud provides visibilities to errors that may occur in the integration in two screens - Input Interfaces and Output Interfaces. The Inbound Interfaces screen will provide visibility to errors that occur in the integration from Merchandising or SIOCS applications into WMS Cloud.

For data inbound to WMS Cloud, the APIs will respond back with appropriate error messages in the response if the message fails validation. These error messages can also be seen in the "Stage Record" tab of "Input Interfaces" screen as shown below which contains both failed and successful messages.

Figure 3-1 Input Interfaces

Input Interfaces

Purchase Order

Upload Files

Run Interface

Files

Stage Record

Reprocess selected records

Reprocess ALL failed

Clear Stage Records

| File group | Header group | PO Nbr | Facility | Company | Action Code | Status | Err Str | Order Date |
|----------------|----------------|---------------|----------|----------|-------------|--------|--|------------|
| 123_2019031... | 23_20190314... | PO03142019... | DC1 | NS_TEST1 | UPDATE | Failed | Sequence Number Mismatch for Item alternate code Item 11 | 2019-03-14 |
| 123_2019031... | 23_20190314... | PO03142019... | DC1 | NS_TEST1 | UPDATE | Failed | Cannot create multiple details with the same sequence number but different items. (Seq nbr: 1, items: Item ... | 2019-03-14 |
| 123_2019031... | 23_20190314... | PO03142019... | DC1 | NS_TEST1 | UPDATE | Failed | Order Quantity cannot be less than Shipped/Received Qty.(Ord_qty is 10.00000, shipped qty is 15.00000, ... | 2019-03-14 |
| 123_2019031... | 23_20190314... | PO03142019... | DC1 | NS_TEST1 | CREATE | Failed | Cannot change the PO as the pre-existing one is already linked to a shipment | 2019-03-14 |
| 123_2019031... | 23_20190314... | PO03142019... | DC1 | NS_TEST1 | CREATE | Failed | Cannot change the PO as the pre-existing one is already linked to a shipment | 2019-03-14 |

If there are issues with the data sent in the interface, appropriate changes need to be made to the data and data needs to be resent. However, if the issue is that dependent data is missing, for example, a purchase order is sent for an item that doesn't exist in WMS Cloud, then once the data is corrected the record can be reprocessed from the Input Interfaces view by selecting the appropriate record and clicking the "Reprocess selected records" button.

For data sent from WMS Cloud to Merchandising, SIOCS, and OROB, the status of the messages sent from WMS cloud can be seen in the Output Interfaces view. Like the Input Interfaces view, both successful and failed messages are visible in this screen.

Figure 3-2 Output Interfaces

| Output Interfaces | | | | | |
|-------------------|--------------|--------------------------|--------------------------------|-------------------------|-----------------------|
| Interface Logs | | Resend | Download | Output Interface Target | |
| Facility Code | Company Code | Interface Type | Filename | Status | Create Timestamp |
| DC1 | NS_TEST1 | Inventory History Export | IHTHGNS_TEST1DC10000003325.xml | Failed | 01/17/2019 8:44:54 PM |
| DC1 | NS_TEST1 | Inventory History Export | IHTHGNS_TEST1DC10000016853.xml | Processed | 02/22/2019 2:33:33 AM |
| DC1 | NS_TEST1 | Inventory History Export | IHTHGNS_TEST1DC10000016938.xml | Processed | 02/22/2019 3:58:31 AM |
| DC1 | NS_TEST1 | Inventory History Export | IHTHGNS_TEST1DC10000016943.xml | Processed | 02/22/2019 4:03:31 AM |

If messages appear with Failed status, then the details of the errors can be viewed by clicking on the Interface Logs button, as seen below. Depending on the error, this view will also allow the message to be resent by clicking on the Resend button.

Figure 3-3 Output Interfaces – Interface Logs

| ← Output Interfaces → Interface Logs | | | |
|--------------------------------------|---|--|-----------------------|
| Sequence Nbr | Connection Info | Message | Create Timestamp |
| 4 | 1 - REST Web Service: oic.nsuser@https://p1.logfireapp... | Could not POST to URL: 403 - FORBIDDEN | 01/17/2019 8:44:55 PM |
| 3 | 1 - REST Web Service: oic.nsuser@https://p1.logfireapp... | Response content: <!DOCTYPE html> <html lang="e... | 01/17/2019 8:44:55 PM |
| 2 | 1 - REST Web Service: oic.nsuser@https://p1.logfireapp... | POST - https://p1.logfireapps.com/ns_test2/index/ | 01/17/2019 8:44:55 PM |
| 1 | 1 - REST Web Service: oic.nsuser@https://p1.logfireapp... | Transmission Processing Started | 01/17/2019 8:44:55 PM |

Universal Service Mapper (USM)

The Universal Service Mapper (USM) can also be used to view any errors that may occur during the transfer of data.

The landing page for this mapper is the Home tab which provides the overall status of the system. It also gives the number of available mappings, total service activity count and the number of successful and failed activities.

Figure 3-4 USM Home

ORACLE® Universal Service Mapper

Welcome, usmadmin

Thu Sep 05 05:54 EDT 2019

Select Project: LogFireIntegration

Home

Mapping Designer

Test Drivers

Monitoring

Admin

Import/Export

Configurations

System Logs

System Summary

Available Mappings16

Service Activity Count88

System Health1

Successful Activity56

Failed Activity32

USM Metrics for today

Search activities by service mapper name

</

The Monitoring tab gives information on the success or failure of a particular flow, including the start and end time of the source operation and the status of the mapping operation. Based on that, you will be able to see the data before and after the mapping for the highlighted flow in the table to help with error tracking.

Note

USM message logging is optimized to improve performance. Larger messages will be truncated to 1 MB and then saved to the database. These truncated messages show up as “before mapping” and “after mapping” on the UI.

Figure 3-5 USM Monitoring

Oracle Universal Service Mapper

Welcome, usmadmin

Select Project: LogFireIntegration ▼

Home | Mapping Designer | Test Drivers | Monitoring | Admin | Import/Export | Configurations | System Logs

USM Metrics

Search Criteria: Service Mapper Name All Select Date from 05/13/2019 02:06 AM to 09/09/2019 03:02 AM Showing 1 to 10 of 88 records

| Activity Id | Start Time | End Time | Mapper Service Name | Source Operation | Status |
|-------------|-----------------------|-----------------------|--|------------------|--------|
| 121 | Sep 04 04:20 EDT 2019 | Sep 04 04:20 EDT 2019 | LogFireToRRecept_ServiceMappingOrchestration.smo | HttpPost | ✓ |
| 106 | Aug 28 02:44 EDT 2019 | Aug 28 02:44 EDT 2019 | RibToLogFireVendor_ServiceMappingOrchestration.smo | VendorCre | ✓ |
| 105 | Aug 28 02:30 EDT 2019 | Aug 28 02:30 EDT 2019 | RibToLogFireVendor_ServiceMappingOrchestration.smo | VendorDel | ✓ |
| 104 | Aug 28 02:28 EDT 2019 | Aug 28 02:28 EDT 2019 | RibToLogFireVendor_ServiceMappingOrchestration.smo | VendorDel | ✗ |
| 103 | Aug 28 02:18 EDT 2019 | Aug 28 02:18 EDT 2019 | RibToLogFireVendor_ServiceMappingOrchestration.smo | VendorCre | ✓ |
| 102 | Aug 28 02:13 EDT 2019 | Aug 28 02:13 EDT 2019 | RibToLogFireVendor_ServiceMappingOrchestration.smo | VendorCre | ✓ |
| 101 | Aug 28 02:04 EDT 2019 | Aug 28 02:04 EDT 2019 | LogFireToRibInAdjust_ServiceMappingOrchestration.smo | HttpPost | ✓ |

Response and Request Data for Activity id: 121

Request Data Before Mapping

```
<?xml version='1.0'?>
<LogData><Header><DocumentVersion>5.0.1</DocumentVersion>
<OriginSystem>LogFire-OriginsSystem</OriginSystem>
<ParentCompanyCode><Entity>Inventory_History</Entity>
<TimeStamp>2019-08-26T04:20:20</TimeStamp>
<MessageId>HHTHONCO_E862245288500023062501</MessageId>
<LogInventoryHeadlines>=inventory_history_vgroup_rtbl4641</Group_Id>
<seq_nbr>1</seq_nbr>
<FacilityCode>622452885-facility_code</FacilityCode>
```

Request Data After Mapping

```
<ApplicationMessages xmlns="http://www.oracle.com/integration/rib/ApplicationMessages/v1">
  xsi:schemaLocation="http://www.oracle.com/integration/rib/ApplicationMessages/v1 schema.xsd" xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">
```

Additionally, the System Logs tab gives detailed information at an individual flow level. The status of each message can be viewed by clicking on the respective flow and viewing the corresponding logs.

Figure 3-6 USM System Logs

ORACLE

Universal Service Mapper

Welcome, useradmin
Thu Sep 05 05:56 EDT 2019

Select Project: LogFireIntegration

Home

Mapping Designer

Test Drivers

Monitoring

Admin

Imports/Export

Configurations

System Logs

USM Log Files

Select Log Location: Default

| File Name | Size (in KB) | Last Modified |
|-----------------------------------|--------------|------------------------------|
| usm-default.log | 280.29 | Thu Sep 05 05:56:48 EDT 2019 |
| LogFireToRdbReceiver-system.log | 62.04 | Wed Sep 04 04:20:56 EDT 2019 |
| RdbToLogFireVendor-system.log | 612.49 | Wed Aug 28 02:44:15 EDT 2019 |
| LogFireToRdbmAdjust-system.log | 22.61 | Wed Aug 28 02:04:03 EDT 2019 |
| RdbToLogFireStockOrder-system.log | 387.54 | Tue Aug 27 05:17:35 EDT 2019 |
| RdbToLogFireOrder-system.log | 373.63 | Tue Aug 27 02:51:25 EDT 2019 |
| RdbToLogFireStores-system.log | 92.33 | Tue Aug 27 02:36:44 EDT 2019 |
| RdbToLogFireRTVReq-system.log | 46.21 | Tue Aug 27 02:36:09 EDT 2019 |
| RdbToLogFireASIN-system.log | 61.13 | Tue Aug 27 02:35:44 EDT 2019 |

File Content

```

2019-09-04T04:20:56,154 [[ACTIVE]] ExecuteThread: '32' for queue: 'weblogic.kernel.Default (self-tuning)'] DEBUG GeneratedMethodAccessor417 - .....
2019-09-04T04:20:56,160 [[ACTIVE]] ExecuteThread: '32' for queue: 'weblogic.kernel.Default (self-tuning)'] DEBUG GeneratedMethodAccessor417 - .....@BIZInt.....
2019-09-04T04:20:56,162 [[ACTIVE]] ExecuteThread: '32' for queue: 'weblogic.kernel.Default (self-tuning)'] DEBUG GeneratedMethodAccessor417 - .....
2019-09-04T04:20:56,165 [[ACTIVE]] ExecuteThread: '32' for queue: 'weblogic.kernel.Default (self-tuning)'] DEBUG GeneratedMethodAccessor417 - log6j2.xml loaded from zip://scratch:/u1/domain/u1_rgt/RTOdomain/servers/usm-server/tmp/_u1_usr-usm-web-var/0160h/var/NB-INT/_u1_cis_gen.jar/log6j2.xml
2019-09-04T04:20:56,165 [[ACTIVE]] ExecuteThread: '32' for queue: 'weblogic.kernel.Default (self-tuning)'] DEBUG GeneratedMethodAccessor417 - Class loaded from /u1/rghu_rtg/usm_data
2019-09-04T04:20:56,162 [[ACTIVE]] ExecuteThread: '32' for queue: 'weblogic.kernel.Default (self-tuning)'] INFO GeneratedMethodAccessors38 - Reloading external_env_info..json as it has not been reloaded for a while.
2019-09-04T04:20:56,163 [[ACTIVE]] ExecuteThread: '32' for queue: 'weblogic.kernel.Default (self-tuning)'] DEBUG GeneratedMethodAccessor417 - envfiledir = /u1/rghu_rtg/usm_data/service_mappers
2019-09-04T04:20:56,165 [[ACTIVE]] ExecuteThread: '32' for queue: 'weblogic.kernel.Default (self-tuning)'] DEBUG GeneratedMethodAccessor417 - External env map item name(https.proxyhost)
2019-09-04T04:20:56,165 [[ACTIVE]] ExecuteThread: '32' for queue: 'weblogic.kernel.Default (self-tuning)'] DEBUG GeneratedMethodAccessor417 - External env map item value(https.proxy.us.oracle.com)
2019-09-04T04:20:56,165 [[ACTIVE]] ExecuteThread: '32' for queue: 'weblogic.kernel.Default (self-tuning)'] DEBUG GeneratedMethodAccessor417 - External env map item name(https.proxyHost)

```


4

Data Conversion

This chapter lists the setup and conversion of data to be done before any messages are sent between solutions. This includes conversion of master data, like stores, warehouses, and items, as well as conversion of any open transactional data and initialization of inventory levels.

Manual Conversion

For a few foundational data elements, it is recommended that you just manually set up the data in WMS Cloud to match what you have in your Merchandising implementation. This includes the configuration of order types and reason codes.

Order Types

The **Order type** field is used to define the set of transactions that will be mapped to Merchandising, SIOCS, and OROB. These need to be manually set up within WMS Cloud and cannot be pre-configured. One record needs to be created in WMS Cloud for each of the order types that is listed below. This is case sensitive and only these order types are supported in the integration.

Table 4-1 Order Types

| Oracle Retail Transaction | WMS Cloud | |
|----------------------------------|------------|--------------------------|
| | Order Type | Description |
| Customer Orders | B2C | Sales Orders |
| Stock Orders and Purchase Orders | B2B | Stock Orders |
| RTV Request | RTV | Return to Vendor Request |

The flags for configuration of the order types can be set however works best for your business, with some exceptions described below.

For more information on how to create order types and how these flags are used in WMS Cloud, see the Oracle Warehouse Management Cloud documentation.

Stock Orders

The **Partial allocation** flag must not be checked when integrating the stock order status for stock allocation or deallocation with Merchandising.

Customer Orders

The **Single Order on Multiple Loads** must be set to **Not Allowed** when integrating with OROB, because it can only handle a single tracking number for each fulfillment order.

Customer Orders (Store Pickup)

For B2C orders that don't need to be directly shipped to the customer but instead will be picked up from the store, USM will send only the destination store ID to WMS Cloud and not the address details of the customer. Also, the **Consumer Direct** field (set to Y in this case) that is required for SIOCS while processing the message will be present in the ASNOut message for this workflow.

Return to Vendor Request

To support multiple inventory status codes in the stock return request from Merchandising, as well as to send RTV shipment with the inventory from a locked status to Merchandising, the following configurations must be made for an RTV order type:

- **Eligible Lock Code for Allocation** - must be set to a comma-delimited list of all lock codes that could have RTVs created against them. In Merchandising, these lock codes are mapped to the inventory status codes within an inventory status. For more details on the lock code and inventory status code mapping, see the "[Lock Codes](#)" section below.
- **Wave Flag** - this flag must be unchecked. RTV orders will be allocated and picked using the "RF Direct Allocation" or "RF Pick and Allocate" modules in WMS.
- **Allocate during Pick** - this flag must be checked to allow allocation when the order is picked for return.

Reason Codes

Reason codes in WMS Cloud are similar to the inventory adjustment reason codes used in Merchandising. If you already have inventory adjustments reasons configured in Merchandising, then these same codes should be setup in WMS Cloud, as appropriate. If you are adding new codes for your WMS Cloud implementation, then you will need to add them in both solutions.

Note

Merchandising requires inventory adjustment reason codes to be numeric and less than or equal to four digits in length.

Additionally, you will need to set up a reason code in WMS Cloud with the ID of **RMA**, which should be used for customer order returns processing in the warehouse.

Customer Order Returns

For customer order returns processed into the warehouse, WMS Cloud does not update Merchandising directly, as Merchandising is expecting an update through returns processing (**OMS > Sales Audit > Merchandising**). So, the return should be mapped only to OMS from WMS. To prevent the return from flowing through and updating Merchandising, a **reason code RMA** must be created in WMS. This is what is used in the integration to filter these transactions from updating Merchandising. This reason code must be used when processing inventory adjustments in WMS related to a customer order return. To better support this process, it is recommended that you create a distinct RF screen for customer orders using the Createln option and as part of that setup, set **RMA** as the default **reason-code** with the **mode** set to either **prompt for reason code OR null**.

Note

Customer order returns cannot use Receiving in WMS when integrated with Merchandising.

Receiver Unit Adjustments

If you want receipt adjustments made during pre-verification to be communicated to Merchandising as receipt adjustments, rather than just an inventory adjustment, then you should configure a reason code in WMS Cloud named `RUA`. This will be a signal to the integration of this adjustment to map it to the Merchandising as a receipt adjustment rather than an inventory adjustment. To better support this process, it is recommended that you create a distinct RF screen for receipt adjustment using the **ModifyIn** option and, as part of that setup, set `RUA` as the default reason code with the mode set to either `prompt for reason code` OR `null`.

Note

The `RUA` reason code should only be used for the purpose described above. Usage of that reason code for other adjustments will raise errors since Merchandising requires that reason codes for inventory adjustments be numeric.

Lock Codes

Lock codes in WMS Cloud are similar to inventory status codes in Merchandising. These need to be coordinated between the two solutions in order to properly map locked inventory in WMS to Merchandising. There are two components of this in WMS Cloud and two components of this in Merchandising that should be configured. First, in WMS Cloud, when creating the lock code, you will also configure an ERP bucket that this lock code should map to. This is because in some cases two lock codes can be assigned for an item in WMS Cloud, but in Merchandising it will always only have one unavailable status at a time. You can also define a priority for the lock code. If used, this will determine which is used to update Merchandising in the case that two are added or removed. Both the lock code and ERP bucket values must exist in Merchandising. See "[Appendix - Activity Codes](#)" for details on how they are translated in the integration.

| | |
|---|-------------------------------------|
| Lock Code * | DM |
| Description | Damaged |
| Allocatable | <input type="checkbox"/> |
| Unlock on locate to Reserve | <input type="checkbox"/> |
| Treat as Attribute | |
| Allow Multiple Lock Codes | <input type="checkbox"/> |
| Generate Adjustments Instead of Lock Code Records | <input type="checkbox"/> |
| Allow Loading | <input type="checkbox"/> |
| Allow Manifesting | <input type="checkbox"/> |
| Prevent Putaway | <input type="checkbox"/> |
| Copy to Transfer Shipment flag | <input checked="" type="checkbox"/> |
| Prevent Direct Consume | <input type="checkbox"/> |
| Inventory attribute transfer code | |
| ERP Bucket | TRBL |
| Lock Code Priority | 1 |

Both the lock code and ERP bucket value will also need to be created as Inventory Status Codes setup in Merchandising. Then, if you wish this code to update an unavailable inventory bucket in Merchandising, you must also map the status code to an inventory status in Merchandising. This allows you to track fewer unavailable statuses in Merchandising than in WMS, if desired. Any inventory status codes that are not mapped to a status in Merchandising will update stock on hand, but not the non-sellable quantity in Merchandising. These codes are used in processing integrations for receipts of POs and stock orders, inventory adjustments and RTVs.

The example below shows how this might be configured:

Table 4-2 WMS Cloud

| Lock Code | ERP Bucket |
|----------------|------------|
| DM - Damaged | TRBL |
| IR - In Repair | TRBL |

Table 4-3 Merchandising

| Inventory Status Code | Inventory Status |
|-----------------------|------------------|
| DM - Damaged | 1 - Damaged |
| IR - In Repair | 1 - Damaged |
| TRBL - Trouble | 1 - Damaged |

In this case, both the lock codes in WMS Cloud are mapped to the same unavailable bucket in Merchandising.

Note

Allocatable lock codes, which allows different classifications of available inventory, are not supported in the WMS Cloud and Merchandising integration.

Convert Master Data

This section describes how to initialize master data in WMS Cloud based from Merchandising data. Master data includes stores, warehouses, external finishers, supplier sites and items from Merchandising. Prior to running the integration to load data into Cloud WMS, the key master data from Merchandising must be loaded into WMS Cloud, to ensure all subscribers and publishers can execute successfully.

Other Notes

- The WMS Managed flag for each of the below described entities in WMS Cloud should be set to N (false).
- Where necessary, truncate Merchandising values to ensure that they meet any size limitations of WMS Cloud.
- In addition to the data elements called out below, you will need to validate that the codes used for language, states and countries match between Merchandising/SIOCS and WMS Cloud as part of the initial configuration.
- WMS Cloud is a multi-tenant application and requires a company code¹ to be sent in all inbound integrations, including those recommended for data conversion below. The code to be used when loading data can be found in the Companies screen in WMS Cloud for your implementation.

Note

This code will be assigned to your company on provisioning of WMS Cloud. It is not the same ID created as the top level of the organizational and merchandise hierarchies in Merchandising.

- Use the action code CREATE for all conversion data loading, when required on the WMS Cloud inbound file.
- To extract data from your Merchandising database to support the conversion requirements outlined below, it is recommended that you use one of the following methods:
 - Replicated DAS schema
 - Application Express (APEX) Data Viewer
 - BI Publisher

Stores, Partners, and Warehouses

First, stores, partners, and warehouses must be loaded. For stores and partners, these can be setup manually in the WMS Cloud Facilities screen, or loaded from data extracted from

¹ This code will be assigned to your company on provisioning of WMS Cloud. It is not the same ID created as the top level of the organizational and merchandise hierarchies in Merchandising.

Merchandising. It is recommended that data be loaded using the Store and Site interfaces to minimize data entry errors.

Stores

Extract the attributes for stores into a .csv file, based on the attributes in the Store API. The table below outlines how the Merchandising attributes map to the WMS Cloud attributes.

Table 4-4 Stores Mappings

| WMS Cloud | Merchandising |
|-----------|----------------------|
| code | store.store |
| name | store.store_name |
| address_1 | addr.add_1 |
| address_2 | addr.add_2 |
| city | addr.city |
| locality | addr.county |
| state | addr.state |
| country | addr.country_id |
| zip | addr.post |
| contact | store.store_mgr_name |
| phone_nbr | store.phone_number |
| email | store.email |
| lang | store.lang |

Other Notes

- Use the primary business address (type 01) as the address for this import.

External Finishers

Extract attributes for external finishers into a .csv file, based on the attributes in the Site API. The table below outlines how the Merchandising attributes map to the WMS Cloud attributes.

Table 4-5 External Finisher Mappings

| WMS Cloud | Merchandising |
|-----------|----------------------|
| code | partner.partner_id |
| name | partner.partner_desc |
| address_1 | addr.add_1 |
| address_2 | addr.add_2 |
| city | addr.city |
| locality | addr.county |
| state | addr.state |
| country | addr.country_id |
| zip | addr.post |
| contact | addr.contact_name |

Table 4-5 (Cont.) External Finisher Mappings

| WMS Cloud | Merchandising |
|-----------|--------------------|
| phone_nbr | addr.contact_phone |
| email | addr.contact_email |
| lang | partner.lang |

Other Notes

- Only partners of type 'E' (External Finisher) from Merchandising should be loaded into WMS Cloud.
- Use the primary business address (type 01) as the address for this import.

Warehouses

WMS Cloud does not have an inbound integration for warehouses, so they must be created within the solution manually through the Facilities screens. These warehouse IDs in WMS Cloud should match the physical warehouse IDs in Merchandising. For more details on how to create facilities, refer to the Oracle Warehouse Management Cloud documentation.

Note

If you add any new physical warehouses in Merchandising post implementation of this integration, those facilities will also need to be created manually in WMS Cloud using similar rules to that described above.

Vendors

Extract attributes for supplier sites into a .csv file, based on the attributes in the Vendor API. The table below outlines how the Merchandising attributes map to the WMS Cloud attributes.

Table 4-6 Vendors Mappings

| WMS Cloud | Merchandising |
|-------------|--------------------|
| vendor_code | sups.supplier |
| name | sups.sup_name |
| address_1 | addr.add_1 |
| address_2 | addr.add_2 |
| address_3 | addr.add_3 |
| city | addr.city |
| state | addr.state |
| country | addr.country_id |
| zip | addr.post |
| contact | addr.contact_name |
| phone_nbr | addr.contact_phone |

Table 4-6 (Cont.) Vendors Mappings

| WMS Cloud | Merchandising |
|-----------|--------------------|
| email | addr.contact_email |

Other Notes

- Only supplier sites should be loaded into WMS Cloud.
- Use the primary business address (type 01) as the address for this import.

Items

To initialize items in WMS Cloud, it is recommended that they be loaded from data exported from Merchandising to create a .csv file with the attributes required. Then, use the APIs provided by WMS Cloud for loading. Only approved items should be loaded into WMS Cloud.

Item Header

Use the Item API to load transaction level items from Merchandising. The table below outlines how the Merchandising attributes map to the WMS Cloud attributes.

Table 4-7 Item Header Mappings

| WMS Cloud | Merchandising |
|-------------------------------|--|
| part_a or item_alternate_code | item_master.item |
| part_a | item_master.item_parent |
| part_b | item_master.diff1 ':' diff_ids.diff_1_description - truncated to fit the field size in WMS Cloud |
| part_c | item_master.diff2 ':' diff_ids.diff_2_description - truncated to fit the field size in WMS Cloud |
| part_d | item_master.diff3 ':' diff_ids.diff_3_description - truncated to fit the field size in WMS Cloud |
| part_e | item_master.diff4 ':' diff_ids.diff_4_description - truncated to fit the field size in WMS Cloud |
| hierarchy1_code | item_master.dept |
| hierarchy1_description | deps.dept_name |
| hierarchy2_code | item_master.class |
| hierarchy2_description | class.class_name |
| hierarchy3_code | item_master.subclass |
| hierarchy3_description | subclass.sub_name |
| description | item_master.item_desc |
| short_descr | item_master.short_desc |
| shipping_temperature_instr | item_master.handling_temp |
| catch_weight_method | (see below) |
| retail_price | (see below) |
| brand_code | item_master.brand_name |

Table 4-7 (Cont.) Item Header Mappings

| WMS Cloud | Merchandising |
|----------------------|---------------------------|
| invn_attr_a_tracking | item_master.inventory_ind |
| description2 | image: (see below) |

Other Notes

- Only approved, transaction-level items should be loaded into WMS Cloud using this API.
- For items without a parent or with a parent but not using diffs in Merchandising, part_a should be set equal to the item ID when loaded into WMS Cloud.
- Merchandising will not send length, width, height, weight, and cube information to WMS Cloud. These fields will be protected in WMS Cloud. This is due to differences in UOM related fields between the two solutions.
- Values not listed above are not available in Merchandising so should be left null in the import to WMS Cloud.
- Item Facility records do not need to be loaded. They will be created automatically in WMS Cloud.
- If an item has a defined catch_weight_type in Merchandising, then use Y for the catch_weight_method field. Otherwise, set to N.
- For retail price, the integration will send the regular retail price for the item in the base zone converted to the primary currency and standard unit of measure. Similar logic can be used for the item conversion.
- Hierarchy levels 4 and 5 should be left null in WMS Cloud.
- The primary item image for an item in Merchandising will be mapped to the WMS Cloud description2 field for items in the integration, if defined. For conversion, it is recommended you do the same, concatenating the image_addr and the image_name for the primary image defined for your item.

Item Barcodes

Use the Item Barcode API to load reference items from Merchandising to WMS Cloud. The table below outlines how the Merchandising attributes map to the WMS Cloud attributes.

Table 4-8 Item Barcodes Mappings

| WMS Cloud | Merchandising |
|----------------|-------------------------|
| vendor_barcode | item_master.item |
| item_barcode | item_master.item_parent |

Other Notes

- Only approved reference items (where tran_level+1 = item_level) from Merchandising for transaction items that meet the criteria of the item import above should be mapped in this API.
- WMS Cloud requires a mandatory barcode for each item loaded into it, and in cases where Merchandising does not have this value, the item can be duplicated as the barcode.

Item Pre-packs

Use the Item Pre-packs API to load the components of a pack item from Merchandising to WMS Cloud. The table below outlines how the Merchandising attributes map to the WMS Cloud attributes.

Table 4-9 Item Pre-packs Mappings

| WMS Cloud | Merchandising |
|------------------|---------------------------------|
| parent_item_code | packitem_breakout.pack_no |
| seq_no | packitem_breakout.seq_no |
| child_item_code | packitem_breakout.item |
| child_units | packitem_breakout.pack_item_qty |

Other Notes

- Only component details for approved, complex pack items should be loaded into WMS Cloud using this API.
- Simple packs are treated as regular items in terms of stocking and inventory movement within WMS Cloud, hence they will be sent in the integration as part of the item feed.

Item Supplier Pack Size

Use the Item Supplier Country API to load the pack size details for an item from Merchandising to WMS Cloud. The table below outlines how the Merchandising attributes map to the WMS Cloud attributes.

Table 4-10 Item Supplier Pack Size Mappings

| WMS Cloud | Merchandising |
|------------------|-----------------|
| std_pack_qty | inner_pack_size |
| std_case_qty | supp_pack_size |
| lpns_per_tier | ti |
| tiers_per_pallet | hi |

Other Notes

- This is an optional data point; the item supplier pack size details are not interfaced in the default setup of this integration.
- Only pack sizes for the primary supplier and country are loaded into WMS Cloud using this API.

Item Dimensions

Use the Item Dimension API to load the dimension details for an item from Merchandising to WMS Cloud. The table below outlines how the Merchandising attributes map to the WMS Cloud attributes.

Table 4-11 Item Dimensions Mappings

| WMS Cloud | Merchandising |
|-----------|--|
| length | Based on item_supp_country_dim.dim_object, For dim object as each, item_supp_country_dim. unit_length For dim object as inner, item_supp_country_dim. std_pack_length For dim object as case, item_supp_country_dim. std_case_length |
| width | Based on item_supp_country_dim.dim_object, For dim object as each, item_supp_country_dim. unit_width For dim object as inner, item_supp_country_dim. std_pack_width For dim object as case, item_supp_country_dim. std_case_width |
| height | Based on item_supp_country_dim.dim_object, For dim object as each, item_supp_country_dim. unit_height For dim object as inner, item_supp_country_dim. std_pack_height For dim object as case, item_supp_country_dim. std_case_height |
| weight | Based on item_supp_country_dim.dim_object, For dim object as each, item_supp_country_dim. unit_weight For dim object as inner, item_supp_country_dim. std_pack_weight For dim object as case, item_supp_country_dim. std_case_weight |

Other Notes

- This is an optional data point; the item dimension details are not interfaced in the default setup of this integration.
- Only dimensions for the primary supplier and country are loaded into WMS Cloud using this API.

Convert Transactional Data

It is highly recommended that you close all open shipments between the facilities that will be using WMS Cloud and your stores so that there are no open transfers or allocations for the facility, including any customer orders from the warehouse. This means that no stock orders would need to be converted into WMS Cloud from Merchandising. However, it is expected that some open purchase orders may exist that need to be converted. Additionally, there will be a need to initialize the inventory values in WMS Cloud and synch this with Merchandising. The sections below outline the recommended approach for this.

Purchase Orders

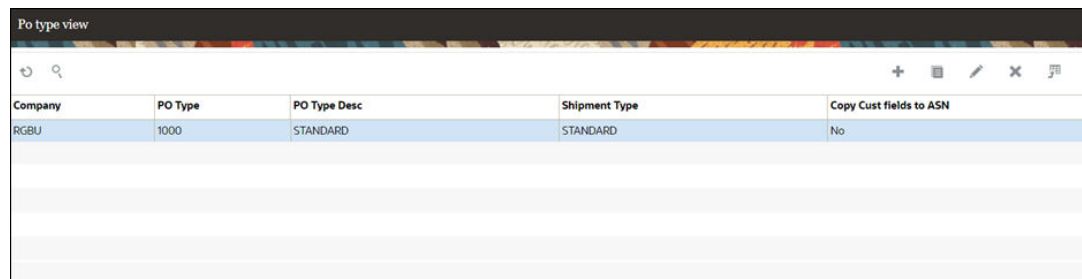
To initialize purchase orders in WMS Cloud, it is recommended that they be loaded from data exported from Merchandising using the Data Viewer or BI Publisher to create a .csv file with the attributes needed. Then, use the APIs provided by WMS Cloud for loading. Only approved purchase orders that have open quantity still to be received should be loaded into WMS Cloud.

Table 4-12 Purchase Orders Mappings

| WMS Cloud | Merchandising |
|----------------------|--|
| facility_code | wh.physical_wh |
| po_nbr | ordhead.order_no |
| dept_code | ordhead.dept |
| vendor_code | ordhead.supplier |
| cancel_date | ordhead.not_after_date |
| cust_field_1 | ordhead.currency_code |
| ord_date | use conversion date |
| delivery_date | ordhead.not_before_date |
| ship_date | ordhead.earliest_ship_date |
| item_alternate_code | ordloc.item |
| facility_code | wh.physical_wh |
| ord_qty | ordloc.qty_ordered (see also note below) |
| uom | item_master.standard_uom |
| unit_cost | ordloc.unit_cost |
| pre_pack_total_units | ordsku.supp_pack_size |
| seq_nbr | (see note below) |

Other Notes

- All open, approved POs that are shipping to a facility that will be managed by WMS Cloud.
- Only include open orders in the converted data. Open orders are those in approved status in Merchandising that have non-received order quantity.
- All virtual warehouse order quantity will need to be summed to the physical warehouse level for this conversion. Virtual warehouses will not be visible in WMS Cloud.
- Within WMS Cloud, the following setup needs to be done manually within the **Po type view** UI as shown below:



| Company | PO Type | PO Type Desc | Shipment Type | Copy Cust fields to ASN |
|---------|---------|--------------|---------------|-------------------------|
| RGBU | 1000 | STANDARD | STANDARD | No |
| | | | | |
| | | | | |
| | | | | |
| | | | | |

This is required at the WMS end while creating a record at their end based on the incoming PO information from the Merchandising end.

- The sequence number required by WMS Cloud is similar to a PO line number and is used for all purchase orders loaded into WMS Cloud. This will need to be generated for the converted orders. The generated sequence number must match the way that the integration layer will generate sequence numbers for purchase orders going forward so that any updates to converted POs will not encounter issues. This is done using a java function hashCode() for the item to generate a unique sequence number.
 - `Math.abs(poDetail.item.toString()).hashCode()`
- Vendor ASNs do not need to be converted.

Inventory


In order to initialize inventory levels in WMS Cloud and ensure that they match the values in Merchandising, it is recommended that a cycle count be created and the results of the count be mapped to a similar count setup in Merchandising. WMS Cloud doesn't support importing a cycle count from an external system, so the recommended approach is as follows²:

Note

After go-live, this is the recommended approach for future cycle counting in the warehouse as well.




1. Schedule a cycle count in WMS Cloud.
2. Schedule a corresponding stock count in Merchandising for the same date/facility including all departments.
3. Export the results of the count from WMS Cloud by scheduling an Inventory Summary export (see below for more information on scheduling jobs).

Figure 4-1 WMS Scheduled Jobs


ORACLE WMS 18C

☐ Input Interfaces
 ☒ **Scheduled Jobs**
☐ Order Types
 ☐ Facility Parameters
 ☐ Output Interfaces

Scheduled Jobs







 Enable/Disable

| Job Number | Job Type | Facility | Company | Schedule Name |
|------------|----------------------------|----------|---------|-------------------|
| J000000025 | Generate Inventory Summary | DC_ES | CO_ES | Inventory Summary |

4. Configure the output interface.

² After go-live, this is also the recommended approach for future cycle counting in the warehouse as well.

Figure 4-2 WMS Output Interface Configuration

| Output Interface Configuration | | | |
|---|--------------|--------------------------|------------------|
|    | | | |
| Facility Code | Company Code | Interface Type | Interface Format |
| DC_ES | CO_ES | Inventory Summary Export | XML |

5. Convert the format exported above into the format used by the Merchandising import for WMS stock counts³, which is used to take a file from a warehouse management system and convert it into the format needed by the Stock Count Upload process. Make sure the date used in the extracted file matches the count created in step [2](#).

Note

This import is called LIFSTKUP_JOB in the job scheduler. It is also part of the STOCKCOUNTUPLOAD_PROCESS and STOCKCOUNT_UPLOAD_PROCESS_ADHOC. See the *Oracle Retail Merchandising Foundation Cloud Service Operations Guide Volume 1* for more details.

6. Run the appropriate processes and take the appropriate actions in Merchandising to upload the results and complete the count.

This should set the inventory levels in WMS Cloud and ensure that they match those in Merchandising.

Other Notes

- Only the total quantity and/or total weight are used by Merchandising in the input file for the import. The other quantity and weight fields are ignored.

³ This import is called LIFSTKUP_JOB in the job scheduler. It is also part of the STOCKCOUNTUPLOAD_PROCESS and STOCKCOUNT_UPLOAD_PROCESS_ADHOC. See the *Oracle Retail Merchandising Foundation Cloud Service Operations Guide Volume 1* for more details.

5

Integration Configuration

This chapter covers configuration that needs to be done in order to extract WMS transaction records and set up the outbound interfaces.

Schedule Jobs

Create a schedule job for Inventory History that will extract the relevant information during scheduled intervals from WMS Cloud. This needs to be set up for Receipts, RTVs, Stock Orders, Inventory Adjustments, and Outbound ASNs. You can create one schedule for all of these entities, or separate schedules by functional area. Segregating has the advantage of setting different frequency on each scheduled job which is set using the Every and Period fields. The example below shows how the Generate Inventory History extract in WMS Cloud could be setup in three different schedules based on the activity code that correlates to a Merchandising entity.

Table 5-1 Inventory History Extract in WMS Cloud

| Job Type | Activity Code | Schedule Name | Schedule Type | Every | Period |
|------------------------------------|---------------|---------------|---------------|--|--------|
| Generate Inventory History Extract | 1 | Receipts | Interval | Set these fields to values between 1 and 5 minutes, depending on the volume of transactions that are being processed in your warehouse. The more transactions being processed, the shorter the interval should be. | |
| Generate Inventory History Extract | 10 | SOSstatus | Interval | | |
| Generate Inventory History Extract | 19 | InvAdj | Interval | | |

Enabled Auto Send

The main way to extract WMS transaction records is through scheduled jobs at pre-determined intervals. You can also automatically send inventory history records in a real-time manner as and when they are created. They are sent through the Output Interface setup just like other outgoing messages configured through scheduled jobs. To enable Auto-Send of Inventory History for an activity code, choose **enable_auto_send** from the Inventory History Activity Parameter screen. If you are using this option for integration, ensure all activity codes generated in the same group in WMS are integrated using the **enable_auto_send** option. This ensures that these integration messages are interfaced in the required order, making consistent updates to Merchandising.

Configure Outbound Interfaces

Outbound interface configurations must be created in WMS Cloud for each facility, one for each of the following exports:

- Inventory History
- Outbound Loads
- Outbound Manifest

Inventory History Export

The Inventory History export is used to communicate updates to receipts, for purchase orders, transfers and allocations, stock order status updates, and inventory adjustments. To configure this for export, set the interface format to XML on the Output Interface Configuration for Inventory History Export.

Note

Although these exports require configuration in WMS, these are not customizable integrations. Only the activity codes described in this section are supported in the base integration.

Next, create three targets for Inventory History Export, using the appropriate host and port for your implementation in the URL definition. The target creation can be done in the WMS Cloud application under the Output Interface Configuration tab.

Figure 5-1 Output Interface Configuration

Sequence Nbr: 1

MHE System:

Interface Protocol: REST Web Service

URL: http://<host>:<port>/usm/EventL

FTP Host:

FTP Port: 0

Target Directory:

Username:

Password:

Encrypt File: ☐

Encrypt Recipient:

Send Acknowledgement Email: ☐

Max Retries: 3

Retry Delay (Seconds): 10

Send Failure Email: ☒

Custom Integration: ☐

Buttons: Save, Cancel, Reset

Target 1: Receipts

URL: http://<host>:<port>/usm/EventListener.do?app=rib-lgf&family=Receiving&type=ReceiptMod

Output Interface Target Criteria:

- Column name = Activity Code
- SQL operator = '='

- Column value = '1|4'

Target 2: SOSStatus

URL: http://<host>:<port>/usm/EventListener.do?app=rib-lgf&family=SOSStatus&type=SOSStatusCre

Output Interface Target Criteria:

- Column name = Activity Code
- SQL operator = IN
- Column value = '10|11|12|27|31|32|54|85'

Target 3: InvAdj

URL: http://<host>:<port>/usm/EventListener.do?app=rib-lgf&family=InvAdjust&type=InvAdjustCre

Output Interface Target Criteria:

- Column name = Activity Code
- SQL operator = IN
- Column value = '2|4|17|19|22|23|24|25|29|30|49|50|53|65'
 - See "[Appendix - Activity Codes](#)" for more details on these activity codes.

Outbound Loads Export

The Outbound Loads export is used to map shipments that are shipped via TL/LTL from the warehouse to Merchandising (RTVs and stock order shipments), SIOCS (stock order shipments to stores), and OROB (stock order shipments for customer orders), as opposed to via a parcel service such as UPS or FedEx¹. To configure this for export, set the interface format to XML on the Output Interface Configuration for Outbound Loads Export.

Note

This determination is made based on the presence or absence of a parcel service in the Ship Via field for an order.

Next, create two targets for Outbound Loads Export, using the appropriate host and port for your implementation in the URL definition. The target creation can be done in the WMS Cloud application under the Output Interface Configuration tab.

¹ This determination is made based on the presence or absence of a parcel service in the Ship Via field for an order.

Figure 5-2 Output Interface Configuration

Sequence Nbr: 1

MHE System:

Interface Protocol: REST Web Service

URL: http://<host>:<port>/usm/EventL

FTP Host:

FTP Port: 0

Target Directory:

Username:

Password:

Encrypt File: ☐

Encrypt Recipient:

Send Acknowledgement Email: ☐

Max Retries: 3

Retry Delay (Seconds): 10

Send Failure Email: ☒

Custom Integration: ☐

Buttons: Save, Cancel, Reset

Target 1: RTV

URL: http://<host>:<port>/usm/EventListener.do?app=rib-lgf&family=RTV&type=RTVCre?interfaceType=ObLoad

Output Interface Target Criteria:

- Column name = Order type
- SQL operator = IN

- Column value = RTV

Target 2: ASNOut

URL: http://<host>:<port>/usm/EventListener.do?app=rib-lgf&family=ASNOut&type=ASNOutMod?interfaceType=ObLoad

Output Interface Target Criteria:

- Column name = Order Type
- SQL operator = IN
- Column value = B2B | B2C

Target 3: ASNOut

URL: http://<host>:<port> /usm/EventListener.do?app=rib-lgf&family=ASNOut&type=ASNOutCre&interfaceType=ObIpnShipping

Output Interface Target Criteria:

- Column name = Order Type
- SQL operator = IN
- Column value = B2B

Outbound Manifest

The Outbound Manifest export is also used to map shipments to Merchandising (RTVs and stock order shipments), SIOCS (stock order shipments to stores), and OROB (stock order shipments for customer orders). But unlike the Outbound Load Export, it is done via a parcel service². To configure this for export, set the interface format to XML on the Output Interface Configuration for Outbound Manifest.

Note

This determination is made based on the presence or absence of a parcel service in the Ship Via field for an order.

² This determination is made based on the presence or absence of a parcel service in the Ship Via field for an order.

Figure 5-3 Output Interface Configuration

DC_ES - CO_ES Administrator

Sequence Nbr 1

MHE System

Interface Protocol REST Web Service

URL http://<host>:<port>/usm/EventL

FTP Host

FTP Port 0

Target Directory

Username

Password

Encrypt File

Encrypt Recipient

Send Acknowledgement Email

Max Retries 3

Retry Delay (Seconds) 10

Send Failure Email ☒

Custom Integration ☐

Save Cancel Reset

Next, create two targets for Outbound Manifest, using the appropriate host and port for your implementation in the URL definition. The target creation can be done in the WMS Cloud application under the Output Interface Configuration tab.

Target 1: RTV

URL: http://<host>:<port>/usm/EventListener.do?app=rib-lgf&family=RTV&type=RTVCre?interfaceType=ObManifest

Output Interface Target Criteria:

- Column name = Order type
- SQL operator = IN
- Column value = RTV

Target 2: ASNOut

URL: http://<host>:<port>/usm/EventListener.do?app=rib-lgf&family=ASNOut&type=ASNOutMod?interfaceType=ObManifest

Output Interface Target Criteria:

- Column name = Order type
- SQL operator = IN
- Column value = B2B | B2C

For ALL of the targets listed above, include the following:

- Interface Protocol = REST Web Service
- Username = <username for your XXX account>
- Password = <password for your XXX account>

Note

If the account password expires or is changed, it needs to be changed here as well.

Other Configurations

A few other configurations need to be considered in WMS Cloud that are used in the integrations with Merchandising, SIOCS, and OROB.

Sequence Counters

You will need to configure the sequence length for specific counters in the Sequence Counters tab in WMS Cloud to ensure the generated sequence numbers do not exceed the length supported by the integration to Merchandising. The following are the counter codes that need to be configured:

BOL Number

The sequence count used for generating BOL numbers is used in the integration of stock order shipments from the warehouse and mapped to the BOL number in Merchandising. It must be configured such that the combined prefix and sequence number does not exceed 17 characters in length that are allowed by the integration.

Inbound Shipment Number

The sequence counter used for generating shipment numbers is used in the integration of DC to DC shipments. It must be configured such that the combined prefix and sequence number do not exceed 30 characters.

Load and Parcel Manifests

The sequence counter used for generating load and parcel manifest numbers is mapped to the RTV external reference number in Merchandising. It is also used for mapping to the ASN number for outbound shipments from the warehouse. For both of these, the sequence must be configured such that the combined prefix and sequence number do not exceed 14 characters.

IB Shipment Types

When mapping inbound ASNs to WMS Cloud, the integration will map to a specific shipment type if the ASN does not contain cartons - this could be a vendor ASN or an ASN from another warehouse or store. This shipment type will be determined, on receiving the ASN, whether or not carton details should be mapped to Merchandising. This is needed because WMS Cloud will create an LPN if one doesn't exist, and if Merchandising also received the inbound ASN from an external system that did not contain carton IDs, it won't be able to understand the carton details from WMS Cloud. To support this, you will need to configure the following shipment type:

- **NO CARTON** - the other required fields in WMS Cloud to values that make sense for your business. Only the shipment type itself is used in the integration.

Company Configurations

Decimal Support

Merchandising supports up to 4 decimal places for quantity and currency values. Hence, it must be ensured that WMS Cloud is also configured to support that many decimal places as well. This is done using the **Companies** view in WMS Cloud by clicking on the **Decimal Settings** option. You will need to configure both the Max Qty Decimal Precision and the Max Weight Volume Dimension Decimal Precision.

The screenshot shows the 'Companies' view in WMS Cloud. At the top, there are tabs: 'Company Security Config', 'Decimal settings', and 'Activate / Deactivate Company'. Below these is a table with columns: 'Company Code', 'Company Name', 'Type', 'Address 1', 'Address 2', 'Address 3', 'Locality', 'City', 'State', 'ZIP', 'Country', 'Phone Nbr', and 'Edit'. The first row shows 'RBU' as the company code, 'RETAIL GLO...' as the company name, and 'Regular' as the type. A 'Decimal settings' dialog box is open in the foreground. It has a 'Password' field with a masked password '*****'. Below it are two fields: 'Max Qty Decimal Precision' and 'max weight volume Dimension decimal precision', both set to '4'. At the bottom of the dialog are 'Submit' and 'Cancel' buttons.

Next, decimal tracking has to be enabled on the items that would be handled in decimal quantities. In the **Items** view, select the item and click on the edit icon. In the edit pane, check the **Handle Decimal Qty** check box.

Items

Print Label Item Pre-Pack Item Material Hazard

| Company | Code | External Style | Style | Part b | Part c | Part d | Part e | Part f | Alternate Item Codes | Barcode | Item Description |
|---------|----------------|----------------|----------------|------------|--------|--------|--------|--------|----------------------|----------------|------------------|
| RGBU | CEREAL&ITE... | | CEREAL&ITE... | BLUE BERRY | SMALL | | | | 107550248 | 40000075002... | Ramya_Rei |
| RGBU | CEREAL&AIT... | | CEREAL&AIT... | APPLE | SMALL | | | | 107550230 | 400000750019 | Ramya_Rei |
| RGBU | 107650126 | | 107650126 | | | | | | 107650151 | 107650151 | Regular Item |
| RGBU | CEREAL&ITEM... | | CEREAL&ITEM... | APPLE | LARGE | | | | 107650134 | CEREAL&ITE... | Cereal APP |
| RGBU | RAMYA&TES... | | RAMYA&TES... | APPLE | LARGE | | | | 107650118 | 400000700045 | Ramya_Rei |
| RGBU | RAMYA&TES... | | RAMYA&TES... | APPLE | LARGE | | | | 107650100 | 400000700038 | Ramya_Rei |
| RGBU | 107550184-B... | | 107550184 | BLUE BERRY | LARGE | | | | 107550205 | 107550205 | Ramya_Rei |
| RGBU | 107550176 | | 107550176 | | | | | | 107550176 | 107550176 | Ramya_Rei |
| RGBU | TEST2&STYL... | | TEST2&STYLE | PLAIN | SMALL | | | | 400000700021 | 400000700021 | Ramya_Rei |
| RGBU | TEST2&STYL... | | TEST2&STYLE | HONEY | SMALL | | | | 400000700014 | 400000700014 | Ramya_Rei |
| RGBU | 107600129 | | 107600129 | | | | | | 107600129 | 107600129 | Regular Item |
| RGBU | 107550133 | | 107550133 | | | | | | 107550133 | 107550133 | Regular Item |
| RGBU | 107600102 | | 107600102 | | | | | | 107600102 | 107600102 | Regular Item |
| RGBU | 107550117 | | 107550117 | | | | | | 107550117 | 107550117 | Regular Item |
| RGBU | 107600081 | | 107600081 | | | | | | 107600081 | 107600081 | Regular Item |
| RGBU | 107550096 | | 107550096 | | | | | | 107550096 | 107550096 | Regular Item |
| RGBU | 107550088 | | 107550088 | | | | | | 107550088 | 107550088 | Auto Item |
| RGBU | 107500081 | | 107500081 | | | | | | 107500081 | 107500081 | PS REG ITI |
| RGBU | 107400207 | | 107400207 | | | | | | 107400207 | 107400207 | Regular Item |
| RGBU | 107400186 | | 107400186 | | | | | | 107400186 | 107400186 | Regular Item |
| RGBU | 107450175 | | 107450175 | | | | | | 107450175 | 107450175 | Regular Item |
| RGBU | 107450159 | | 107450159 | | | | | | 107450159 | 107450159 | Regular Item |
| RGBU | 107450132 | | 107450132 | | | | | | 107450132 | 107450132 | Regular Item |

Hazardous ☐

Pre-Pack Code

Host Aware Item Flag ☒

Is Parent ☐

Dummy SKU Flag ☐

Pack With Wave ☐

Cubiscan Mod Timestamp

Conversion Factor 1

Track Inventory Attr A Not Required

Track Inventory Attr B Not Required

Track Inventory Attr C Not Required

Track Inventory Attr D Not Required

Track Inventory Attr E Not Required

Track Inventory Attr F Not Required

Track Inventory Attr G Not Required

Packing Tolerance Percent 0

Item Description 2

Item Description 3

Image

Handle Decimal Qty ☐

Save Cancel Reset

Enforce Ship Dates

There are parameters at the company level in WMS Cloud³ that determine whether a message will be displayed to the user during loading if the dates sent for an outbound shipment (transfer, allocation, or RTV) are earlier than the start ship date or later than the stop ship date when compared to the current date. If this is not set, then these dates will be for reference only. It is recommended that for implementations with Merchandising that these be set to No.

Company Parameters

| Company | Parameter Key | Parameter Value |
|---------|---|-----------------|
| RGBU | CYCLE_COUNT_REASON_CODE | |
| RGBU | DEFAULT_ERP_BUCKET_FOR_PACKED_INVENTORY | PAK |
| RGBU | DEFAULT_ERP_BUCKET_FOR_RECEIVED_INVENTORY | ATS |
| RGBU | DEFAULT_TRANSFER_IBSHIPMENT_TYPE | |
| RGBU | DFLT_BATCH_EXP_ADJ_REASON_CODE | DM |
| RGBU | DISABLE_LEGACY_CUSTOM_LABEL_LOOKUP_FOR | |
| RGBU | ENFORCE_START_SHIP_DATE | no |
| RGBU | ENFORCE_STOP_SHIP_DATE | no |
| RGBU | ESPL_REPORT_USES_CUSTOM_UOM | No |
| RGBU | FILES_TO_GENERATE_AT_LPN_PACKED | OLI |
| RGBU | FILES_TO_GENERATE_AT_ORDER_PACKED | |

Packing Routing Mode

To support the Ship Via configurations described below, the company level parameter PACKING_ROUTING_MODE should be set to mode_0.

³ ENFORCE_STOP_SHIP_DATE and ENFORCE_START_SHIP_DATE

| Company Parameters | | |
|--------------------|---|-----------------|
| Company | Parameter Key | Parameter Value |
| RGBU | OUTPUT_LGF_FINFO_ORDER_VERIFICATION | |
| RGBU | OUTPUT_LGF_FINFO_PALLET_SHIPPING_INFO | lgf |
| RGBU | OUTPUT_LGF_FINFO_TO_MHE_DISTRIBUTION_INFO | |
| RGBU | OUTPUT_LGF_FINFO_TO_MHE_IBLPN_INFO | |
| RGBU | OUTPUT_LGF_FINFO_WAVE_PICK_INFO | |
| RGBU | PACKING_ROUTING_MODE | MODE_0 |
| RGBU | PACKS_OF_ONE_UNIT_OK | no |
| RGBU | PACK_LPN_OBLPN_LOCATE_PROMPT | no |
| RGBU | PARCEL-DIMENSIONS-UOM | IN |
| RGBU | PARCEL-WEIGHT-UOM | LB |
| RGBU | PARENT_FACILITIES_VISIBLE_TO_CLIENT_COMPANIES | no |

Default Inventory Status

For inventory adjustments, you will also need to configure the value ATS into the DEFAULT_ERP_BUCKET_FOR_RECEIVED_INVENTORY parameter in the Company Parameters page, as shown in the image below. This is used to differentiate inventory adjustments that are moving inventory out of stock completely in the warehouse (which is sent to Merchandising as a null disposition) and those that would move inventory from a locked status to available status in the warehouse, or vice versa.

| Company | Parameter Key | Parameter Value |
|---------|---|-----------------|
| RGBU | GEN_WMS_CSV_WITH_BOM | no |
| RGBU | HONOR_LOCK_CODE_PRIORITY | yes |
| RGBU | DEFAULT_ERP_BUCKET_FOR_PACKED_INVENTORY | PAK |
| RGBU | DEFAULT_ERP_BUCKET_FOR_RECEIVED_INVENTORY | ATS |
| RGBU | CONSIDER_ITEM_STD_UOM_QUANTITY_FLG | yes |
| RGBU | FILES_TO_GENERATE_AT_LPN_PACKED | OLI |
| RGBU | FILES_TO_GENERATE_AT_ORDER_PACKED | |
| RGBU | WRITE_RF_END_OF_SCREEN_MARKER | no |
| RGBU | OUTPUT_LGF_FINFO_PALLET_SHIPPING_INFO | lgf |
| RGBU | OUTPUT_LGF_FINFO_OBLPN_SHIPPING_INFO | lgf18c |
| RGBU | OUTPUT_LGF_FINFO_TO_MHE_IBLPN_INFO | |
| RGBU | OUTPUT_LGF_FINFO_TO_MHE_DISTRIBUTION_INFO | |

Ship Output Merge Details

The company parameter SHIPOUTPUT_MERGE_DETAILS should be set to Yes when integrating with Merchandising. This parameter is to ensure that if an outbound shipment (e.g. allocation to the store) has the same item being picked from multiple locations in the warehouse that the details are merged together on a single line item in the outbound integration, rather than one line per warehouse location.

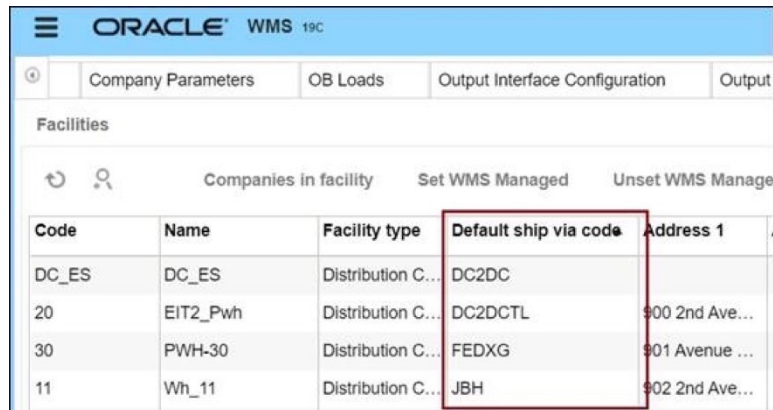
Facility Configurations

Ship Via Configuration

The Ship Via code is used by WMS Cloud to determine the carrier that should be used for shipping and is required for all order types. For B2C orders, this will be communicated from Merchandising based on the information sent from OMS. However, for other order types, B2B and RTV, Merchandising doesn't communicate a preferred carrier. For B2B orders, it is

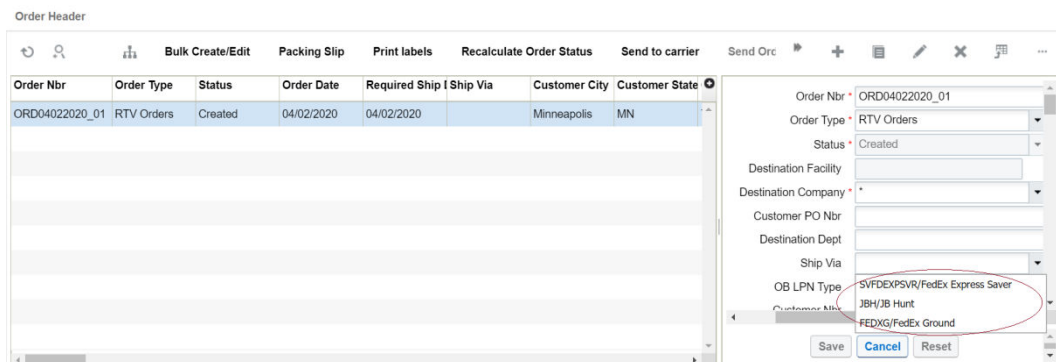
recommended that you set a default at the facility level to prevent having the user from having to enter a value manually for each shipment. This can be done by doing the following:

- At the company level, configure the Packing Routing Mode, as described above.
- At the facility level, select the default ship via code to use for B2B orders. For B2B orders, the Ship Via code must not be longer than 4 characters in order to correctly map to Merchandising.



| Code | Name | Facility type | Default ship via code | Address 1 |
|-------|----------|-------------------|-----------------------|----------------|
| DC_ES | DC_ES | Distribution C... | DC2DC | |
| 20 | EIT2_Pwh | Distribution C... | DC2DCTL | 900 2nd Ave... |
| 30 | PWH-30 | Distribution C... | FEDXG | 901 Avenue ... |
| 11 | Wh_11 | Distribution C... | JBH | 902 2nd Ave... |

For the RTV order type, this will have to be manually selected for each order.



| Order Nbr | Order Type | Status | Order Date | Required Ship I | Ship Via | Customer City | Customer State |
|----------------|------------|---------|------------|-----------------|----------|---------------|----------------|
| ORD04022020_01 | RTV Orders | Created | 04/02/2020 | 04/02/2020 | | Minneapolis | MN |

Create Facility Transfer Shipment for All

This parameter is used by WMS Cloud to enable automatic creation of an ASN in the destination facility. However, the shipment created doesn't contain the order number, which means it cannot be used to communicate the order number as expected by Merchandising on receipt at the destination facility. Instead, when integrated with Merchandising, the inbound shipment will be created by the integration layer based on the outbound ASN from the shipping facility, including the order number, so this parameter should be set to N.

| Facility | Program Key | Parameter Key | Parameter Value |
|----------|---------------|---|-----------------|
| DC1 | FACILITY_PARM | CREATE_FACILITY_TRANSFER_SHIPMENT_FOR_ALL | no |

RF Configuration




WMS Cloud allows for configuration of certain RF screens to support defaulting and influence system behavior. The following configurations are strongly suggested when integrating with Oracle Retail solutions.

Create LPN

Create {lpn} (or Createlpn) is an RF option in WMS Cloud that can be used to add to inventory or move inventory out of an active location. Usually, two different screens are used to handle these two functions.

If you plan to use the Createlpn RF option to move inventory out of an **active** location by setting the **inventory-source** to **Active**, then it is mandatory that you configure it to have the **mode** set to **No reason code prompt** in order to properly integrate with Merchandising.

In the example below, you can see that two modules have been created, one to add inventory and one to move inventory out of active, called Split from Active.

| Screens | | | | |
|---|-------------|-----------------------|-------------------|-------------------|
| <div>   Generate Screens Add to Current Group Menu</div> | | | | |
| Products | Screen Type | Module | Screen Name ▲ | Nbr of Parameters |
| wms | RF-Text | RF-Text: Create {lpn} | Create LPN | 15 |
| wms | RF-Text | RF-Text: Create {lpn} | Split from Active | 15 |

Then, in the screen below, the inventory-source and mode have been configured as required to Active and No reason code prompt, respectively.

| Screens Screen Parameters | | | | |
|---------------------------|----------------------------|----------------|-----------------|-----------------------|
| Screen | Module Parameter | Parameter Type | Parameter Value | Module parm choice |
| Split from Active | extra-item-property-disp | Selection | | |
| Split from Active | default-lockcode | Text | PP | |
| Split from Active | crt-cont-ihl-existing-lpn | Selection | | |
| Split from Active | prompt-batch-expiry | Selection | | |
| Split from Active | expired-invn-lockcode | Text | | |
| Split from Active | inventory-source | Selection | | Active |
| Split from Active | xdock-ordtype | Text | | |
| Split from Active | prompt-locln | Selection | | Prompt location |
| Split from Active | dest-facility-code | Text | | |
| Split from Active | mode | Selection | | No reason code prompt |
| Split from Active | create-lpn-mode | Selection | | |
| Split from Active | xdock-iblpn | Selection | | |
| Split from Active | lpn-as-physical-pallet | Selection | | |
| Split from Active | reason-code | Text | | |
| Split from Active | infer-prepack-parent-it... | Selection | | |

Receive LPN

Depending on the receiving module you are using, Receive {lpn} Shipment or Receive {lpn} Load, you should configure the parameter **prevent-entire-po-to-asn-conversion-flg** to be set to **Yes**. This flag prevents WMS Cloud from converting the entire purchase order into an ASN in situations where an ASN is not received from the supplier, which can limit updates to the PO that can be made in Merchandising during receiving.

| Screen | Module Parameter | Parameter Type | Parameter Value | Module parm choice |
|------------|---|----------------|-----------------|--------------------|
| Receive PO | extra-item-property-disp | Selection | | |
| Receive PO | flowthrough-alloc-mode | Selection | | |
| Receive PO | flowthrough-alloc-prty | Selection | | |
| Receive PO | mhe-system-code | Text | | |
| Receive PO | generate-distro-message | Selection | | |
| Receive PO | lpn-prompt | Selection | | |
| Receive PO | prompt-location | Selection | | |
| Receive PO | print-labels | Selection | | |
| Receive PO | prompt-dock-door | Selection | | |
| Receive PO | prompt-trailer | Selection | | |
| Receive PO | confirm-uom-qty | Selection | | |
| Receive PO | prompt-po-seq-line | Selection | | |
| Receive PO | prevent-entire-po-to-asn-conversion-flg | Selection | | |

Module Parameter: prevent-entire-po-to-asn-convers

Parameter Value:

Module parm choice: **Yes**

Buttons: Save, Cancel, Reset

Time Zone: US/Eastern

Rows Per Page: 25 / 2 Pages

Configure Integration Layer

Configure USM

In USM, URLs will need to be configured for your environment to allow the RIB to communicate with WMS Cloud. This configuration will be done by the Oracle Cloud Operations team, but you may be asked to help provide the URLs for this configuration. These are the two URLs that need to be configured:

- `LogFire_Host_Url_Key` – this is your link to the WMS Cloud application; configure this appropriately to allow USM to connect to the WMS Cloud application.
- `RibLgf_host_UrlKey` – this is the link to the RIB application for this integration; set this field in USM to enable it to connect with the RIB

Configure Static DVMs

Once the links to the end applications are configured, the static DVMs have to be configured with values from your specific implementation for company code and facility codes. These are mandatory values. The DVMs that are to be edited are:

- `CompanyCode_dvm.LogFireIntegration` – this is where the company code is to be set. The company code to use here can be found in the WMS Cloud Companies screen. The entry to be made in the DVM is "CompanyName" in the name column and your company code in the value column. Company code is case sensitive.
- `FacilityCode_dvm.LogFireIntegration` – this is where the facility codes are to be set - one record for each physical warehouse that you configured in WMS Cloud in the Warehouse conversion section. The entry to be made in the DVM is the warehouse ID in the FacilityId column, facility type in the FacilityType column and facility time zone in the FacilityTimeZone column. This should match exactly how you have configured your facilities in WMS Cloud.

Note

There is also a static DVM for country code, however this is not required for the integration so does not need a configuration.

More details about the configuration of the USM application can be found in the *Oracle Retail Integration Cloud Service Universal Service Mapper User Guide*, a link for which is provided in "[Appendix - Activity Codes](#)".

Configure RIB

When being sent from Merchandising to WMS Cloud, data passes through an application called RIB-TAFR (Transformation Addressing Filtering Routing). As part of the standard RIB configuration, the warehouse facility IDs need to be configured in the RIB in order for the TAFR logic to filter and route messages. This is done by the Oracle Cloud Operations team. Once you have your warehouse facilities set up in WMS Cloud, you will need to provide these IDs to the Oracle Cloud Operations team for them to do this configuration. More details can be found in the *Oracle Retail Integration Bus Operations Guide*.

Figure 5-4 RIB Warehouse Facility IDs

rib-tafr: System Options Page Refreshed Mon Apr 22 2019 12:17:49 GMT+0530 (India Standard Time)

Home Adapter Manager Log Manager RIB Logs Manage Configurations **RIB ServiceMonitor**

System Options

Rib Properties

| | | |
|----------------------------|--------------------------------|-------------------------------------|
| facility_id.PROD.1 | <input type="text" value="1"/> | ↗ ↖ |
| facility_id.PROD.121232 | <input type="text" value="1"/> | ↗ ↖ |
| facility_id.PROD.2 | <input type="text" value="1"/> | ↗ ↖ |
| facility_id.PROD.3 | <input type="text" value="1"/> | ↗ ↖ |
| facility_id.PROD.622452885 | <input type="text" value="1"/> | ↗ ↖ |
| facility_id.PROD.716182531 | <input type="text" value="1"/> | ↗ ↖ |
| facility_id.PROD.725746721 | <input type="text" value="1"/> | ↗ ↖ |
| facility_id.PROD.725746731 | <input type="text" value="1"/> | ↗ ↖ |
| facility_id.PROD.882899294 | <input type="text" value="1"/> | ↗ ↖ |

Configure Merchandising

The standard RIB messages used for WMS Cloud integration with Merchandising are generally the same as those that are used for integration with other solutions, such as SIOCS. However, for certain integrations, due to the way that WMS Cloud requires data to be sent in a different manner for modified data. In the standard RIB messages used for SIOCS, only the data elements that have changed are sent. For example, if an item description is updated, only the item ID and description that changed are included in the published modification. However, for WMS Cloud, the full item details need to be resent in the case of a modification or delete. To support this, the Publish Full Objects system option in Merchandising must be checked (Y) in order for the modified data to correctly update WMS Cloud.

Finally, the Retain Customer Information flag must also be checked (Y) in order for customer details to be passed to WMS Cloud for customer order fulfillment.

Configure OMS

OMS is not directly integrated with WMS Cloud to send customer orders; instead customer order details are integrated through Merchandising, including attributes like carrier codes, necessitating coordination of codes used between solutions. In particular, the shipping carrier codes (Ship Via codes in OROMS), which identify the shipping company to be used for the customer order, need to be setup to match the Ship Via codes in WMS Cloud.

For more details on how customer orders integrate to WMS Cloud from OROMS, OROB, and Merchandising, see the *Merchandising and SIM Integration with OMS and OB* white paper found on My Oracle Support under [ID 2088235.1](#).

Note

There is no special configuration required in SIOCS in order for it to receive messages from WMS Cloud. That is all managed through RICS.

A

Appendix - Solution Overviews

Oracle Warehouse Management Cloud

Oracle Warehouse Management Cloud (WMS Cloud) is the industry's leading cloud-based inventory and warehouse management system. With innovative product features, mobile solutions and an easy-to-use interface, Oracle's WMS Cloud combines the speed and savings of the cloud with the best-of-breed warehouse management and fulfillment capabilities.

Retail Integration

The Oracle Retail Integration Cloud Service (RICS) and its on-premise counterpart are comprised of a suite of products that are cloud deployable for integration to Oracle Retail applications and on-premises or hybrid configurations of third-party applications. The integration products supported are the Oracle Retail Integration Bus (RIB), the Oracle Retail Service Backbone (RSB), and Oracle Retail Bulk Data Integration (BDI). It also contains a component called the Universal Service Mapper (USM), which allows for the definition, mapping, configuration, and deployment required to maintain a seamless integration in heterogeneous environments.

Merchandising

The Oracle Retail Merchandising Foundation Cloud Service (RMFCS) and Oracle Retail Merchandising System (RMS) are the market-leading merchandising solution for retailers of all sizes. These solutions enable many of the best retailers in the world to better manage, control, and perform crucial day-to-day merchandising activities with ease. From new product introduction, to financial inventory valuation, it provides retailers with a complete end-to-end solution.

Retail Store Inventory and Operations Cloud Service

Oracle Retail Store Inventory and Operations Cloud Service (SIOCS) provides retailers with a comprehensive collection of automated and simple, predictable and repeatable processes supporting accurate store inventory with enterprise-wide visibility necessary to support supply chain and commerce-anywhere decisions.

Retail Order Broker Cloud Service

Oracle Retail Order Broker Cloud Service is a cloud service application that enables a retailer's order anywhere, fulfill from anywhere, to anywhere strategy with its view of inventory across the enterprise and its rules-based shopping engine, ensuring margin optimization and reduced order fulfillment costs.

B

Appendix - Outbound and Inbound Interfaces

Outbound from WMS Cloud

The following lists all the outbound interfaces from WMS Cloud available in USM, for integration into Merchandising or SIOCS, along with their interface names, template files, service definition files, orchestration files, supported message types, and payloads.

Note

These interfaces have the exact same Failure and Response templates. Namely:

- LogFireToRib_DefaultAllAction_ResponseDataMapping.template
- LogFireToRib_DefaultAllAction_FailureResponseDataMapping.template

You will be able to see the details available in these templates in the USM screens to help with resolving any issues in the integration. See also the [Integration Errors](#).

ASN Outbound

This integration will be used for shipments from the warehouse to a store or to another warehouse. It also will be used for customer orders that are shipped to the customer directly.

WMS Cloud Integrations

Service Definition

- LogFireToRibASNOutObLoad_ServiceMappingDef.json
- LogFireToRibASNOutObManifest_ServiceMappingDef.json

Orchestration File

- LogFireToRibASNOutObLoad_ServiceMappingOrchestration.smo
- LogFireToRibASNOutObManifest_ServiceMappingOrchestration.smo

ASN Out (ASNOutCre)

RIB Payload

- ASNOutDesc

Template Files (Request)

- LogFireToRibASNOutObLoad_HttpPost_RequestResponseDataMapping.template

- LogFireToRibASNOutObManifest_HttpPost_RequestResponseDataMapping.template

Inventory Adjustments

Any inventory adjustments that will increment or decrement inventory in the warehouse will be sent via this integration point. This includes removing inventory from stock completely or moving it to and from an available status. See "[Appendix - Activity Codes](#)" for more information about mapping inventory adjustments.

WMS Cloud Integrations

Service Definition

- LogFireToRibInvAdjust_ServiceMappingDef.json

Orchestration File

- LogFireToRibInvAdjust_ServiceMappingOrchestration.smo

Inventory Adjustment (InvAdjustCre)

RIB Payload

- InvAdjustDesc

Template Files (Request)

- LogFireToRibInvAdjust_HttpPost_RequestDataMapping.template

Return to Vendor Shipments

This integration will be used when shipping inventory back to a supplier from the warehouse. RTV shipments support returns from both available and unavailable inventory buckets. When shipping out an RTV order from a non-sellable bucket, the inventory disposition codes in the shipment will be based on lock codes in WMS.

WMS Cloud Integrations

Service Definition

- LogFireToRibRTV_ServiceMappingDef.json

Orchestration File

- LogFireToRibRTV_ServiceMappingOrchestration.smo

RTV (RTVCre)

RIB Payload

- RTVDesc

Template Files (Request)

- LogFireToRibRTV_HttpPost_RequestDataMapping.template

Receiving

This integration will be used for receipts of purchase orders, transfers, or allocations into the warehouse. Both container level and detailed receipts will flow through this message to update Merchandising.

WMS Cloud Integrations

Service Definition

- LogFireToRibReceipt_ServiceMappingDef.json

Orchestration File

- LogFireToRibReceipt_ServiceMappingOrchestration.smo

Receipt (ReceiptMod)

RIB Payload

- ReceiptDesc

Template Files (Request)

- LogFireToRibReceipt_HttpPost_RequestDataMapping.template

Stock Order Status

This integration will be used to communicate updates for stock orders - which can be either a transfer or allocation. Status updates are based on certain activity codes in the Inventory History table in WMS Cloud. See "[Appendix - Activity Codes](#)" for more details.

WMS Integrations

Service Definition

- LogFireToRibSOStatus_ServiceMappingDef.json

Orchestration File

- LogFireToRibSOStatus_ServiceMappingOrchestration.smo

SO Status (SOStatusCre)

RIB Payload

- SOStatusDesc

Template Files (Request)

- LogFireToRibSOStatus_HttpPost_RequestDataMapping.template

Inbound to WMS Cloud

The following lists the inbound interfaces to interfaces used for integration into WMS Cloud from Merchandising or SIOCS, along with their interface names, template files, service definition files, orchestration files, supported message types, and payload name. Due to the manner in which WMS Cloud consumes data for the following interface points, a full payload is used for sending updates from Merchandising, rather than just the data that has changed: items, purchase orders, stock orders, vendors, and RTV requests, using the "FullRep" message type. In the case of purchase orders, any order line that has already been over-received is not published as a part of this.

The following list of Interfaces have the exact same Failure and Response templates.

- RibToLogFire_DefaultAllAction_FailureResponseDataMapping.template
- RibToLogFire_DefaultAllAction_ResponseDataMapping.template

Store

This integration will send new and updated store information from Merchandising to WMS Cloud. If a store is deleted in Merchandising, that will also be communicated to WMS Cloud.

WMS Cloud Integrations

Service Definition

- RibToLogFireStores_ServiceMappingDef.json

Orchestration File

- RibToLogFireStores_ServiceMappingOrchestration.smo

Create and Modify (StoreCre)

RIB Payload

- StoreDesc

Template Files (Request)

- RibToLogFireStores_StoreCre_RequestDataMapping.template

Delete (StoreDel)

RIB Payload

- StoreRef

Template Files (Request)

- RibToLogFireStores_StoreDel_RequestDataMapping.template

Vendor

This integration will send new and updated supplier site information from Merchandising to WMS Cloud. If a supplier site is deleted, that will also be communicated to WMS Cloud.

WMS Cloud Integrations

Service Definition

- RibToLogFireVendor_ServiceMappingDef.json

Orchestration File

- RibToLogFireVendor_ServiceMappingOrchestration.smo

Vendor Create (VendorCre) and Modify (VendorFulRep)

RIB Payload

- VendorDesc

Template Files (Request)

- RibToLogFireVendor_VendorCre_RequestDataMapping.template

Vendor Delete (VendorDel)

RIB Payload

- VendorRef

Template Files (Request)

- RibToLogFireVendor_VendorDel_RequestDataMapping.template

External Finisher

This integration will send new and updated external finisher information, including addresses, from Merchandising to WMS Cloud. If an external finisher or its address is deleted, that will also be communicated to WMS Cloud.

WMS Cloud Integrations

Service Definition

- RibToLogFirePartner_ServiceMappingDef.json

Orchestration File

- RibToLogFirePartner_ServiceMappingOrchestration.smo

Partner Create (PartnerCre PartnerDtlCre) and Modify (PartnerDtlMod, PartnerMod)

RIB Payload

- PartnerDesc

Template Files (Request)

- RibToLogFirePartner_PartnerCre_RequestDataMapping.template

Partner Delete (PartnerDel)

RIB Payload

- PartnerRef

Template Files (Request)

- RibToLogFirePartner_PartnerDel_RequestDataMapping.template

Partner Details Delete (PartnerDtlDel)

RIB Payload

- PartnerRef

Template Files (Request)

- RibToLogFirePartner_PartnerDtlDel_RequestDataMapping.template

Items

This integration will send new or updated items to WMS Cloud from Merchandising, including parent items, transaction level items, and reference items, as applicable. Only approved items are sent in this integration. For pack items, the component details for the pack are also integrated. See the [Items](#) section for key assumptions about items that are integrated to WMS Cloud.

Note

Item Delete is not currently supported in this integration.

WMS Cloud Integrations

Service Definition

- RibToLogFireItem_ServiceMappingDef.json

Orchestration File

- RibToLogFireItem_ServiceMappingOrchestration.smo

Item Create (ItemCre, ItemUpcCre, ItemBomCre) and Modify (ItemUpcMod, ItemBomMod, ItemFulRep)

RIB Payload

- ItemDesc

Template Files (Request)

- RibToLogFireItem_ItemCre_RequestResponseDataMapping.template
- RibToLogFireItem_ItemUpcCre_RequestResponseDataMapping.template
- RibToLogFireItem_ItemBomCre_RequestResponseDataMapping.template

ASN Inbound

This integration is used to map shipments inbound to a warehouse, which could be from a supplier or a store. Modifications to shipments or deletions of a shipment can also occur based on adjustments sent from a store or supplier.

WMS Cloud Integrations

Service Definition

- RibToLogFireASNIn_ServiceMappingDef.json

Orchestration File

- RibToLogFireASNIn_ServiceMappingOrchestration.smo

Create (ASNInCre or Modify (ASNInMod)

RIB Payload

- ASNInDesc

Template Files (Request)

- RibToLogFireASNIn_ASNInCre_RequestDataMapping.template

Delete (ASNInDel)

RIB Payload

- ASNInRef

Template Files (Request)

- RibToLogFireASNIn_ASNInDel_RequestDataMapping.template

Order

This interface is used to send new and updated purchase orders from Merchandising to WMS Cloud. In case an order is being republished using the `POFulRep` message based on certain

modifications, the order lines that have already been over-received prior to this will not be published again by Merchandising. Only approved orders with a warehouse as one of the destination locations will be sent to WMS Cloud. For more details about PO mapping rules, see the [Purchase Orders](#) section.

WMS Cloud Integrations

Service Definition

- RibToLogFireOrder_ServiceMappingDef.json

Orchestration File

- RibToLogFireOrder_ServiceMappingOrchestration.smo

PO Create (POCre) and Modify (POFulRep)

RIB Payload

- PODesc

Template Files (Request)

- RibToLogFireOrder_POCre_RequestDataMapping.template

PO Delete (PODel)

RIB Payload

- PORef

Template Files (Request)

- RibToLogFireOrder_PODel_RequestDataMapping.template

Stock Order

Stock orders can be either transfers or allocations, including customer order related transfers. These orders are sent from Merchandising to WMS Cloud when the warehouse is either the shipping location or the receiving location on the stock order. Only approved stock orders will be integrated as part of this flow. For stock orders involving a destination warehouse, WMS Cloud requires a purchase order at the receiving location in order to successfully receive the units that are present in the stock order. In order to handle this, a dummy PO with the same item-location combinations as the stock order is internally created and sent along with the Stock Order to WMS Cloud. This PO would not have any vendor details since it is not an actual order originating from the vendor.

WMS Cloud Integrations

Service Definition

- RibToLogFireStockOrder_ServiceMappingDef.json

Orchestration File

- RibToLogFireStockOrder_ServiceMappingOrchestration.smo

Stock Order Create (SOCre) or Modify (SOFulRep)

RIB Payload

- SODesc

Template Files (Request)

- RibToLogFireStockOrder_SOCre_RequestDataMapping.template

Stock Order Header Delete (SOHdrDel)

RIB Payload

- SORef

Template Files (Request)

- RibToLogFireStockOrder_SOHdrDel_RequestDataMapping.template

RTV Requests

This integration will send return to vendor transactions initiated in Merchandising that are requested to be shipped from a warehouse. Merchandising sends the inventory disposition code in the RTV request if the inventory to be returned is from an unavailable bucket. If the RTV line item is to be returned from the multiple inventory status codes, then Merchandising sends these codes in the form of a comma-separated list.

WMS Cloud Integrations

Service Definition

- RibToLogFireRTVReq_ServiceMappingDef.json

Orchestration File

- RibToLogFireRTVReq_ServiceMappingOrchestration.smo

Create RTV Request (RTVReqCre) or Modify (RTVReqFulRep)

RIB Payload

- RTVReqDesc
- RTVReqRef

Template Files (Request)

- RibToLogFireRTVReq_RTVReqCre_RequestDataMapping.template

Delete RTV Request (RTVReqDel)

RIB Payload

- RTVReqDesc

- RTVReqRef

Template Files (Request)

- RibToLogFireRTVReq_RTVReqDel_RequestDataMapping.template

C

Appendix - Activity Codes

Activity Code Mappings

The tables below provide a summary of how certain inventory history codes from WMS Cloud are mapped to Merchandising. In some cases, activity codes are only conditionally updating Merchandising, based on various factors in processing in WMS Cloud.

Note

For all mappings below, in cases where more than one purchase order number is present in an activity code record (comma-delimited list) from WMS and is to be used in processing in the integration with Merchandising, the first purchase order number listed in the record will be used.

Table C-1 Receipts (PO and Stock Order)

| WMS Cloud | | Merchandising | | Comments |
|---|----------|---------------|-----------------|--|
| Lock Code | Qty Used | Update SOH? | Update Unavail? | |
| Activity Code 1 - Container Received | | | | |
| No | Adj Qty | Yes | No | Receipt into available inventory |
| Current/ Previous ERP Buckets | Adj Qty | Yes | Yes | Receipt into unavailable inventory |
| Activity Code 4 - Inventory Adjusted pre-verification | | | | |
| Current/ Previous ERP Buckets | Adj Qty | Yes | Yes | If the reason code is RUA and there is a PO number in the transaction, this will be mapped as a receiver unit adjustment. Otherwise, it will be mapped as an inventory adjustment. |

Table C-2 Stock Order Status

| WMS Cloud | | Merchandising | | Comments |
|--|----------|---------------|-----------------|--|
| Lock Code | Qty Used | Update SOH? | Update Unavail? | |
| Activity Code 10 - Container Detail Packed | | | | |
| | Orig Qty | No | No | Increases distro quantity on transfer/allocation only. No impact to inventory in Merchandising. If a customer order, no update to Merchandising, only Order Broker (or 3rd party OMS). |

Table C-2 (Cont.) Stock Order Status

| WMS Cloud | | Merchandising | | Comments |
|---|----------|---------------|-----------------|--|
| Lock Code | Qty Used | Update SOH? | Update Unavail? | |
| Activity Code 11 - Container Packed | | | | |
| | Orig Qty | No | No | Increases distro quantity on transfer/allocation only. No impact to inventory in Merchandising. If a customer order, no update to Merchandising, only Order Broker (or 3rd party OMS). |
| Activity Code 12 - Order Detail Shorted | | | | |
| | Adj Qty | No | No | Increases cancel quantity on transfer/allocation only. No impact to inventory in Merchandising. If a customer order, no update to Merchandising, only Order Broker (or OMS). Note: if the activity code has a ref value 2 = 91, this is ignored by the integration; this is used for internal WMS processing only. |
| Activity Code 27 - Order Cancellation | | | | |
| | Adj Qty | No | No | Increases cancel quantity on transfer/allocation only. No impact to inventory in Merchandising. If a customer order, no update to Merchandising, only Order Broker (or OMS). Note: if the activity code has a ref value 2 = 91, this is ignored by the integration; this is used for internal WMS processing only. |
| Activity Code 31 - OB Container Modified | | | | |
| | Adj Qty | No | No | Decreases distro quantity on transfer/allocation only. No impact to inventory in Merchandising. If a customer order, Order Broker will move the order line back to Accepted status based on this status update. |
| Activity Code 32 - OB Container Cancelled | | | | |
| | Adj Qty | No | No | Decreases distro quantity on transfer/allocation only. No impact to inventory in Merchandising. If a customer order, Order Broker will move the order line back to Accepted status based on this status update. |
| Activity Code 54 - Container Detail Picked | | | | |
| | Orig Qty | No | No | Increases distro quantity on transfer/allocation only. No impact to inventory in Merchandising. If a customer order, no update to Merchandising, only Order Broker (or OMS). |
| Activity Code 85 - Order Detail Status Change | | | | |
| | Adj Qty | No | No | Increases selected quantity on transfer/allocation only. No impact to inventory in Merchandising. If a customer order, no update to Merchandising, only Order Broker (or 3rd party OMS). |

Table C-3 Inventory Adjustments

| WMS Cloud | | Merchandising | | Comments |
|--|----------|---------------|-----------------|---|
| Lock Code | Qty Used | Update SOH? | Update Unavail? | |
| Activity Code 2 - Container Consumed | | | | |
| | Adj Qty | Yes | | |
| Activity Code 4 - Inventory Adjusted pre-verification | | | | |
| Current/ Previous ERP Buckets | Adj Qty | Yes | Yes | If the reason code is RUA and there is a PO number in the transaction, this will be mapped as a receiver unit adjustment (see Receipts above). Otherwise, it will be mapped as an inventory adjustment. |
| Activity Code 17 - Inventory Adjusted post-verification | | | | |
| Current/ Previous ERP Buckets | Adj Qty | Yes | | |
| Activity Code 19 - Inventory Adjustment Cycle Count Active | | | | |
| | Adj Qty | Yes | | |
| Activity Code 22 - Lock Container - Before ASN Verification | | | | |
| Current/ Previous ERP Buckets | Orig Qty | | Increase | |
| Activity Code 23 - Lock Container - After ASN Verification | | | | |
| Current/ Previous ERP Buckets | Orig Qty | | Increase | |
| Activity Code 24 - Unlock Container - Before ASN Verification | | | | |
| Current/ Previous ERP Buckets | Orig Qty | | Decrease | |
| Activity Code 25 - Unlock Container - After ASN Verification | | | | |
| Current/ Previous ERP Buckets | Orig Qty | | Decrease | |
| Activity Code 29 - Create Allocatable Container | | | | |
| | Adj Qty | Increase | | Only processed if the Inventory History reason code is not RMA and the order number is null. If there is a value in the order number, then this will be mapped to the "Table C-2" message (see above). |
| Activity Code 30 - Create Unallocatable Container | | | | |

Table C-3 (Cont.) Inventory Adjustments

| WMS Cloud | | Merchandising | | Comments |
|---|----------|---------------|-----------------|---|
| Lock Code | Qty Used | Update SOH? | Update Unavail? | |
| Current/ Previous ERP Buckets | Adj Qty | Increase | Increase | Only processed if the order number is null. If there is a value in the order number, then this will be mapped to the " Table C-2 " message (see above). |
| Activity Code 49 - Lock Active | | | | |
| Current/ Previous ERP Buckets | Orig Qty | | Yes | |
| Activity Code 50 - Unlock Active | | | | |
| Current/ Previous ERP Buckets | Orig Qty | | Yes | |
| Activity Code 53 - Cycle Count - Reserve SKU Counted | | | | |
| Current/ Previous ERP Buckets | Adj Qty | Yes | | |
| Activity Code 65 - Lock OBLPN | | | | |
| Current/ Previous ERP Buckets | Orig Qty | Yes | | |

D

Appendix - Known Issues

The following section covers features and workflows that are currently not supported in both the solutions, the integration flow, or both of these.

Table D-1 Known Issues

| Issue | Notes |
|--------------------------------|--|
| Merchandising update on waving | <p>When WMS Cloud starts working on a stock order (transfer or allocation) by waving, the activity code used to support this update does not have quantity information available to update Merchandising and, therefore, the SO status message to Merchandising assumes the entire stock order is allocated when a stock order is waved in WMS Cloud.</p> <p>If you don't wish to use this feature, it is recommended that you do not configure the activity code in WMS Cloud.</p> |
| Item Delete | When an item is deleted in Merchandising, it is not communicated to WMS Cloud. Such items must be manually deleted from WMS when deleted from Merchandising. |
| Stock Order Delete | When a stock order is closed in Merchandising, the details are not communicated to WMS Cloud. It is expected that these orders would be separately purged at the WMS end based on their system configuration. |
| Transfers with Finishing | Integration of finishing details is not supported between Merchandising and WMS Cloud. If you have internal finishers in your warehouses where you need to send finishing information to the warehouse, this integration will need to be field enabled in order to support this process. |
| Customer Order Returns | <p>Integration between Oracle Retail Order Management Cloud Service (OROMS) and WMS Cloud for customer order returns to a warehouse is not currently supported in the integration. Customers requiring this functionality can look at these two options:</p> <ul style="list-style-type: none"> Processing the return in both WMS Cloud and OROMS manually. In WMS Cloud, it would be processed as an adjustment using reason code RMA (which is not mapped to Merchandising). Then, when the return is processed in OROMS, it will be communicated to Merchandising (including the warehouse details) to update inventory.¹ Field enable the integration between WMS Cloud and OROMS using standard APIs for each product. |
| Cycle Counts | Cycle counts between WMS and Merchandising require manual communication of the items to be counted and the date for the count, based on the cycle count request created in Merchandising. After the count, the results should be extracted into a file to be sent to Merchandising in the format specified in the Conversion of WMS Stock Count Results File (lifestkup) integration. See the <i>Merchandising Operations Guide</i> for more details. |
| Allocatable Lock Codes | Allocatable lock codes, which allows different classifications of available inventory, are not supported in the WMS Cloud and Merchandising integration. This functionality should not be configured in WMS Cloud. |
| Ship Via for RTVs | The ship via code on an order is used by WMS Cloud to determine the carrier that should be used for shipping and is required for all shipments by WMS Cloud. For B2C or B2B order types, this is either provided in the integration (B2C) or able to be derived (B2B). However, for RTVs this must be manually selected on the order. |

Table D-1 (Cont.) Known Issues

| Issue | Notes |
|--------------------------------|---|
| Transaction Item with a Parent | <p>Transaction-level items setup in Merchandising with a parent, where the diffs at the parent level are all Diff IDs, rather than one or more diff groups cannot be supported when integrating with WMS Cloud. This is due to the way in which WMS Cloud handles the item code by concatenating the item parts 'a-f' together in a way that must be unique. See the Convert Master Data section for information on how items are mapped to the WMS Cloud parts.</p> <p>Items with a parent that does not use diff groups or diffs is supported and will be treated like an item without a parent in WMS Cloud.</p> |

¹ See also the Merchandising and SIM Integration with OMS and OB white paper at My Oracle Support ID 2088235.1.

E

Appendix - Other Resources

WMS Cloud

For other documentation on WMS Cloud including links to user documentation, parcel carrier integration, and integration details, see document ID 2422433.2 on My Oracle Support and reference the Documentation tab.

Oracle Retail Integration

For other documentation on retail integration, including links to integration guides, USM and RIB documentation, and updates on the most recent releases, see the link to the official product documentation page here: <https://docs.oracle.com/en/industries/retail/index.html>.

Oracle Retail Merchandising

For access to the Merchandising operations and implementation guides, as well as all other documentation, see the link to the official product documentation page here: <https://docs.oracle.com/en/industries/retail/index.html>.

Other Documentation References

For details on customer order integration between all the solutions covered in this document, see the white paper available on My Oracle Support document ID 2088235.1 titled Merchandising and SIM Integration with OMS and OB.