

Oracle® Retail Merchandising Foundation Cloud Service

Operations Guide Volume 2 - Message Publication and Subscription Designs



Release 19.3.000

F44402-12

January 2024

The Oracle logo, consisting of a solid red square with the word "ORACLE" in white, uppercase, sans-serif font centered within it.

ORACLE®

F44402-12

Copyright © 2024, Oracle and/or its affiliates.

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	xxxviii
Documentation Accessibility	xxxviii
Related Documents	xxxviii
Customer Support	xxxix
Improved Process for Oracle Retail Documentation Corrections	xxxix
Oracle Retail Documentation on the Oracle Help Center (docs.oracle.com)	xl
Conventions	xl

1 Introduction

2 RIB Publication Designs

Allocation Publication API	2-1
Functional Area	2-1
Business Overview	2-1
New Allocations	2-2
New Allocation Details	2-5
Updated Allocations	2-5
Deleted Allocations	2-6
Error Handling	2-7
Message XSD	2-7
ASN Outbound Publication API	2-8
Functional Area	2-8
Business Overview	2-8
On-line Shipping/Receiving	2-8
Franchise Order Shipment and Return	2-9
Advance Shipment Notification	2-9
Error Handling	2-13

Message XSD	2-13
Available Inventory for Store and Warehouse Publication API	2-14
Functional Area	2-14
Business Overview	2-14
Item-Location Level Available to Sell Message	2-14
Error Handling	2-15
Message XSD	2-15
Banner and Channel Publication API	2-15
Functional Area	2-15
Business Overview	2-15
New Banners	2-16
Updated Banners	2-16
Deleted Banners	2-17
New Channels	2-17
Updated Channels	2-17
Deleted Channels	2-17
Error Handling	2-18
Message XSD	2-18
Brand Publication API	2-18
Functional Area	2-18
Business Overview	2-18
New Brand	2-18
Updated Brand	2-19
Deleted Brand	2-19
Error Handling	2-19
Message XSD	2-19
Code and Diff Type Publication API	2-19
Functional Area	2-19
Business Overview	2-20
New Code Types	2-20
Updated Code Types	2-20
Deleted Code Types	2-20
New Codes	2-21
Updated Codes	2-21
Deleted Codes	2-21
New Diff Types	2-22
Updated Diff Types	2-23
Deleted Diff Types	2-23
Error Handling	2-23
Message XSD	2-24
Company Closed Publication API	2-24

Functional Area	2-24
Business Overview	2-24
New Closings	2-24
Updated Closings	2-25
Deleted Closings	2-25
New Closing Exceptions	2-25
Updated Closing Exceptions	2-25
Deleted Closing Exceptions	2-26
Error Handling	2-26
Message XSD	2-26
Country Publication API	2-26
Functional Area	2-26
Business Overview	2-27
New Country	2-27
Updated Country	2-27
Deleted Country	2-27
Error Handling	2-28
Message XSD	2-28
Customer Order Fulfillment Confirmation Publication API	2-28
Functional Area	2-28
Business Overview	2-28
Error Handling	2-29
Message XSD	2-30
Delivery Slot Publication API	2-30
Functional Area	2-30
Business Overview	2-30
Create Delivery Slot	2-30
Update Delivery Slot	2-30
Delete Delivery Slot	2-31
Error Handling	2-31
Message XSD	2-31
Differentiator Group Publication API	2-31
Functional Area	2-31
Business Overview	2-31
New Diff Group Header	2-31
New Diff Group Detail	2-32
Updated Diff Group Header	2-32
Updated Diff Group Detail	2-32
Deleted Diff Group	2-32
Deleted Diff Group Detail	2-33
Error Handling	2-33

Message XSD	2-33
Differentiator Publication API	2-33
Functional Area	2-34
Business Overview	2-34
New Differentiator IDs	2-34
Updated Differentiator IDs	2-34
Deleted Differentiator IDs	2-34
Error Handling	2-35
Message XSD	2-35
Item Location Publication API	2-35
Functional Area	2-35
Business Overview	2-35
New Item Location Relationship	2-36
Item/Location Create and Update	2-36
Item/Physical Location Create and Update	2-36
Item/Virtual Location Create and Update	2-37
Modified Item Location Relationship	2-41
Deleted Item Location Relationship	2-41
Item/Location Delete	2-41
Item/Physical Location Delete	2-42
Item/Virtual Location Delete	2-42
Item Replenishment Attributes	2-42
Item/Location Virtual Location Replenishment Details	2-42
Flex Attributes	2-43
Flex Attributes	2-43
Error Handling	2-43
Message XSD	2-44
Item Publication API	2-44
Functional Area	2-44
Business Overview	2-44
Deposit Items	2-44
Catch-Weight Items	2-45
Item Transformation	2-46
Item and Item Component Descriptions	2-47
New Item Message Processes	2-48
Basic Item Message	2-48
New Item Message Publication	2-48
Subordinate Data and XML Tags	2-48
Modify and Delete Messages	2-66
Modify Messages	2-66
Delete Messages	2-66

Flex Attributes	2-69
Full Message Updates	2-70
Error Handling	2-70
Message XSD	2-70
Merchandise Hierarchy Publication API	2-71
Functional Area	2-71
Business Overview	2-72
New Division	2-72
Updated Division	2-72
Deleted Division	2-72
New Group	2-73
Updated Group	2-73
Deleted Group	2-73
New Department	2-73
Updated Department	2-74
Deleted Department	2-75
New Class	2-75
Updated Class	2-75
Deleted Class	2-76
New Subclass	2-76
Updated Subclass	2-76
Deleted Subclass	2-77
Custom Flex Attributes	2-77
Error Handling	2-78
Message XSD	2-78
Order Publication API	2-78
Functional Area	2-79
Business Overview	2-79
New Purchase Orders	2-79
Updated Purchase Orders	2-85
Full Message Updates	2-86
Deleted Purchase Orders	2-86
Flex Attributes	2-86
Error Handling	2-87
Message XSD	2-87
Organizational Hierarchy Publication API	2-88
Functional Area	2-88
Business Overview	2-88
New Chains/Areas/Regions/Districts	2-88
Updated Chains/Areas/Regions/Districts	2-89
Deleted Chains/Areas/Regions/Districts	2-89

Error Handling	2-89
Message XSD	2-90
Partner Publication API	2-90
Functional Area	2-90
Business Overview	2-90
Partner Creation	2-90
Partner Header Modification	2-93
Partner Detail Creation	2-93
Partner Detail Modification	2-93
Deleting Partners	2-93
Deleting Partner Details	2-94
Flex Attributes	2-94
Error Handling	2-94
Message XSD	2-95
Receiver Unit Adjustment Publication API	2-95
Functional Area	2-95
Business Overview	2-95
Creating Receiver Unit Adjustments	2-95
Error Handling	2-96
Message XSD	2-96
RTV Request Publication API	2-97
Functional Area	2-97
Business Overview	2-97
Create RTV Request	2-97
Update RTV Request	2-99
Delete RTV Request	2-99
Flex Attributes	2-99
RTV Request Full Publish	2-100
Error Handling	2-100
Message XSD	2-100
Season and Phase Publication API	2-100
Functional Area	2-100
Business Overview	2-101
New Seasons	2-101
Updated Seasons	2-101
New Phases	2-101
Updated Phases	2-102
Deleted Seasons	2-102
Deleted Phases	2-102
Error Handling	2-102
Message XSD	2-102

Store Publication API	2-103
Functional Area	2-103
Business Overview	2-103
Create Stores	2-103
Modify Stores	2-103
Create Store Addresses	2-103
Modify Store Addresses	2-103
Create Store Hours	2-104
Modify Store Hours	2-104
Flex Attributes	2-108
Delete Stores	2-109
Error Handling	2-109
Message XSD	2-109
Transfer Publication API	2-110
Functional Area	2-110
Business Overview	2-110
New Transfer	2-110
Updated Transfer	2-110
Flex Attributes	2-118
Deleted Transfer	2-118
Transfer Unapprove	2-119
Error Handling	2-119
Message XSD	2-119
UDA Publication API	2-120
Functional Area	2-120
Business Overview	2-120
New UDAs	2-120
Updated UDAs	2-121
Deleted UDAs	2-121
New UDA Details	2-121
Updated UDAs	2-122
Deleted UDAs	2-122
Error Handling	2-122
Message XSD	2-122
Vendor Publication API	2-123
Functional Area	2-123
Business Overview	2-123
Vendor Creation	2-123
Vendor Header Modification	2-123
Vendor Address Creation	2-128
Vendor Address Modification	2-128

Vendor Detail Deletion	2-129
Flex Attributes	2-130
Full Message Updates	2-130
Error Handling	2-131
Message XSD	2-131
Warehouse Publication API	2-131
Functional Area	2-131
Business Overview	2-131
Warehouse Creation	2-131
Warehouse Modification	2-132
Warehouse Addresses	2-135
Deleting Warehouses	2-136
Flex Attributes	2-137
Error Handling	2-137
Message XSD	2-137
Work Orders In Publication API	2-138
Functional Area	2-138
Business Overview	2-138
New Work Order	2-138
Updated Work Order	2-139
Deleted Work Order	2-139
Error Handling	2-140
Message XSD	2-140
Work Orders Out Publication API	2-140
Functional Area	2-140
Business Overview	2-141
New Work Order	2-141
Deleted Work Order	2-143
Error Handling	2-144
Message XSD	2-144

3 RIB Subscription Designs

Allocation Subscription API	3-1
Functional Area	3-1
Business Overview	3-1
Allocation Creation	3-1
Allocation Modification	3-1
Allocation Details	3-2
Deleting Allocations	3-3
Error Handling	3-3

Message XSD	3-4
Appointment Subscription API	3-4
Functional Area	3-4
Business Overview	3-4
New Appointments	3-4
New Appointment Details	3-4
Updated Appointments	3-4
Updated Appointment Details	3-5
Delete Appointments	3-5
Delete Detail Level Appointments	3-5
Error Handling	3-6
Message XSD	3-6
ASN Outbound Subscription API	3-6
Functional Area	3-7
Business Overview	3-7
Other Notes	3-7
Shipment Message Details	3-8
ASNOUT Item Details	3-11
Error Handling	3-12
Message XSD	3-13
Cost Change Subscription API	3-13
Functional Area	3-13
Business Overview	3-13
Custom Flex Attributes	3-15
Error Handling	3-15
Message XSD	3-15
Currency Exchange Rate Subscription API	3-16
Functional Area	3-16
Business Overview	3-16
Error Handling	3-16
Message XSD	3-17
Customer Order Fulfillment Subscription API	3-17
Functional Area	3-17
Business Overview	3-17
Other Notes	3-18
Error Handling	3-25
Message XSD	3-25
Differentiator Group Subscription API	3-26
Functional Area	3-26
Business Overview	3-26
Creating Diff Groups	3-26

Updating Diff Groups	3-27
Deleting Diff Groups	3-27
Error Handling	3-28
Message XSD	3-28
Differentiator Subscription API	3-28
Functional Area	3-28
Business Overview	3-28
Creating Differentiators	3-29
Updating Differentiators	3-29
Deleting Differentiators	3-29
Error Handling	3-30
Message XSD	3-30
DSD Deals Subscription API	3-30
Functional Area	3-30
Business Overview	3-30
Error Handling	3-32
Message XSD	3-32
DSD Receipt Subscription API	3-32
Functional Area	3-32
Business Overview	3-32
Error Handling	3-34
Message XSD	3-34
Freight Term Subscription API	3-35
Functional Area	3-35
Business Overview	3-35
Creating/Updating Freight Terms	3-35
Error Handling	3-35
Message XSD	3-36
General Ledger Chart of Accounts Subscription API	3-36
Functional Area	3-36
Business Overview	3-36
Create and Update Chart of Accounts	3-36
Error Handling	3-38
Message XSD	3-38
Inbound ASN Subscription API	3-38
Functional Area	3-38
Business Overview	3-39
Creating/Updating ASN	3-39
Delete ASN	3-42
Error Handling	3-43
Message XSD	3-43

Inventory Adjustment Subscription API	3-43
Functional Area	3-43
Business Overview	3-43
Inventory Adjustment Transaction Codes	3-44
Other Notes:	3-44
Error Handling	3-46
Message XSD	3-46
Inventory Request Subscription API	3-46
Functional Area	3-46
Business Overview	3-46
Creating Inventory Requests	3-47
Error Handling	3-48
Message XSD	3-48
Item Location Subscription API	3-48
Functional Area	3-48
Business Overview	3-48
Creating Locations	3-48
Updating Item Location	3-48
Flex Attributes	3-52
Error Handling	3-53
Message XSD	3-53
Item Reclassification Subscription API	3-53
Functional Area	3-53
Business Overview	3-53
New/Updated Reclassifications	3-54
Deleting Reclassifications	3-55
Error Handling	3-57
Message XSD	3-57
Item Subscription API	3-57
Functional Area	3-57
Business Overview	3-57
Creating Items	3-58
Updating Items	3-64
Deleting Items	3-64
Creating an Item Supplier	3-65
Updating an Item Supplier	3-66
Deleting an Item Supplier	3-67
Creating an Item Supplier Country	3-67
Updating an Item Supplier Country	3-69
Deleting an Item Supplier Country	3-69
Creating Item Supplier Manufacturing Country	3-70

Updating Item Supplier Manufacturing Country	3-70
Deleting Item Supplier Manufacturing Country	3-70
Creating Item Supplier Country Location	3-70
Updating Item Supplier Country Locations	3-72
Deleting Item Supplier Country Locations	3-72
Creating Item Supplier Country Dimensions	3-72
Updating Item Supplier Country Dimensions	3-73
Deleting Item Supplier Country Dimensions	3-74
Creating Item VAT	3-74
Deleting Item VAT	3-74
Creating Item UDA	3-75
Updating Item UDA	3-75
Deleting Item UDA	3-75
Creating Item Season	3-76
Deleting Item Season	3-76
Creating Item Images	3-77
Updating Item Images	3-77
Deleting Item Images	3-78
Creating HTS and Assessments	3-78
Updating HTS and Assessments	3-79
Deleting HTS and Assessments	3-79
Creating Expenses	3-80
Updating Expenses	3-81
Deleting Expenses	3-81
Creating Tickets	3-81
Updating Tickets	3-82
Deleting Tickets	3-82
Creating Up-charges	3-82
Deleting Up-charges	3-83
Updating Up-charges Detail	3-83
Creating Item Country	3-84
Deleting Item Country	3-84
Flex Attributes	3-84
Translated Language	3-84
Error Handling	3-86
Message XSD	3-86
Location Trait Subscription API	3-88
Functional Area	3-88
Business Overview	3-88
Error Handling	3-88
Message XSD	3-89

Merchandise Hierarchy Reclassification Subscription API	3-89
Functional Area	3-89
Business Overview	3-89
Creating Merchandise Hierarchy Reclassifications	3-89
Updating Merchandise Hierarchy Reclassifications	3-91
Deleting Merchandise Hierarchy Reclassifications	3-91
Error Handling	3-92
Message XSD	3-92
Merchandise Hierarchy Subscription API	3-92
Functional Area	3-92
Business Overview	3-93
Creating a Company	3-93
Updating a Company	3-93
Creating Divisions	3-94
Updating Divisions	3-94
Deleting Divisions	3-95
Creating Groups	3-95
Updating Groups	3-95
Deleting Groups	3-96
Creating Departments	3-96
Updating Departments	3-97
Deleting Departments	3-98
Creating VAT Information for Departments	3-99
Updating VAT Information for Departments	3-99
Creating Up-charges for Departments	3-99
Updating Up-charges for Departments	3-100
Deleting Up-charges for Departments	3-101
Creating Classes	3-102
Updating Classes	3-102
Deleting Classes	3-102
Creating Subclasses	3-103
Updating Subclasses	3-103
Deleting Subclasses	3-104
Flex Attributes	3-104
Error Handling	3-104
Message XSD	3-105
Order Subscription API	3-105
Functional Area	3-105
Business Overview	3-106
Creating Purchase Orders	3-106
Updating Purchase Orders	3-112

Deleting Purchase Orders	3-114
Creating a Purchase Order Letter of Credit	3-115
Updating a Purchase Order Letter of Credit	3-115
Deleting a Purchase Order Letter of Credit	3-115
Creating Expenses	3-116
Updating Expenses	3-117
Deleting Expenses	3-117
Creating HTS and Assessments	3-118
Updating HTS and Assessments	3-119
Deleting HTS and Assessments	3-119
Publishing Updates	3-120
Flex Attributes	3-120
Error Handling	3-120
Message XSD	3-120
Organizational Hierarchy Subscription API	3-121
Functional Area	3-121
Business Overview	3-121
Error Handling	3-122
Message XSD	3-122
Payment Terms Subscription API	3-122
Functional Area	3-123
Business Overview	3-123
Creating Payment Terms	3-123
Updating Payment Terms	3-126
Error Handling	3-126
Message XSD	3-126
Receiving Subscription API	3-126
Functional Area	3-126
Business Overview	3-126
Receiving Exceptions	3-127
Receipt	3-128
Receipt Detail	3-129
Receipt Carton Detail	3-131
Error Handling	3-132
Message XSD	3-132
RTV Subscription API	3-132
Functional Area	3-132
Business Overview	3-132
New RTVs	3-133
Updated RTVs	3-133
RTV Header	3-133

RTV Details	3-135
Flex Attributes	3-136
Error Handling	3-136
Message XSD	3-136
Stock Count Schedule Subscription API	3-137
Functional Area	3-137
Business Overview	3-137
Creating/Updating Stock Count Requests	3-137
Deleting Stock Count Requests	3-138
Error Handling	3-139
Message XSD	3-139
Stock Order Status Subscription API	3-139
Functional Area	3-139
Business Overview	3-140
Stock Order Statuses	3-140
Stock Order Status Message Details	3-142
Error Handling	3-144
Message XSD	3-144
Store Subscription API	3-144
Functional Area	3-144
Business Overview	3-145
Create or Update Stores	3-145
Delete Stores	3-145
Other Store Attributes	3-145
Other Notes	3-145
Store Message Details	3-145
Flex Attributes	3-154
Error Handling	3-154
Message XSD	3-154
Transfer Subscription API	3-155
Functional Area	3-155
Business Overview	3-155
Creating Transfers	3-155
Transfers with Finishing	3-162
Updating Transfers	3-162
Deleting Transfers	3-163
Publishing Updates	3-163
Flex Attributes	3-163
Error Handling	3-164
Message XSD	3-164
Vendor Subscription API	3-164

Functional Area	3-164
Business Overview	3-164
Vendor Create and Update	3-165
Flex Attributes	3-170
Error Handling	3-171
Message XSD	3-171
Work Order Status Subscription API	3-171
Functional Area	3-171
Business Overview	3-171
Work Order Example	3-172
Work Order Status Creation	3-172
Error Handling	3-173
Message XSD	3-173

4 SOAP Web Services

Using SOAP Services During Batch Window	4-2
Common Characteristics of Merchandising SOAP Services	4-2
Security	4-2
Standard Success Response	4-2
Standard Error Response	4-3
URL Path	4-3
Web Service APIs Process Flow	4-3
Provider Services	4-4
Consumer Services	4-4
Provider Services	4-4
Allocation Service	4-5
Functional Area	4-5
RSB Proxy WSDL	4-5
Merchandising Service WSDL	4-5
Overview	4-5
Operation XSD	4-5
Average Cost Service	4-6
Functional Area	4-6
RSB Proxy WSDL	4-6
Merchandising Service WSDL	4-6
Overview	4-6
Operation XSD	4-6
Cost Change Service	4-6
Functional Area	4-6
RSB Proxy WSDL	4-6

Merchandising Service WSDL	4-7
Overview	4-7
Operation XSD	4-7
Customer Credit Check Service	4-7
Functional Area	4-7
RSB Proxy WSDL	4-7
Merchandising Service WSDL	4-7
Overview	4-7
Operation XSD	4-8
Customer Order Fulfillment Service	4-8
Functional Area	4-8
RSB Proxy WSDL	4-8
Merchandising Service WSDL	4-8
Overview	4-8
Operation XSD	4-9
Customer Order Item Substitution Service	4-9
Functional Area	4-9
RSB Proxy WSDL	4-9
Merchandising Service WSDL	4-9
Business Overview	4-9
Assumptions	4-9
Operation XSD	4-10
Diff Management Service	4-10
Functional Area	4-10
RSB Proxy WSDL	4-10
Merchandising Service WSDL	4-10
Overview	4-10
Operation XSD	4-10
Inventory Back Order Service	4-11
Functional Area	4-11
RSB Proxy WSDL	4-11
Merchandising Service WSDL	4-11
Overview	4-11
Assumptions	4-11
Operation XSD	4-12
Inventory Lookup Service	4-12
Functional Area	4-12
RSB Proxy WSDL	4-12
Merchandising Service WSDL	4-12
Overview	4-12
Assumptions	4-12

Operation XSD	4-13
Item Management Service	4-13
Functional Area	4-13
RSB Proxy WSDL	4-13
Merchandising Service WSDL	4-13
Overview	4-13
Operation XSD	4-14
Pay Term Service	4-15
Functional Area	4-15
RSB Proxy WSDL	4-15
Merchandising Service WSDL	4-15
Overview	4-15
Operation XSD	4-16
Pricing Cost Service	4-16
Functional Area	4-16
RSB Proxy WSDL	4-16
Merchandising Service WSDL	4-16
Overview	4-16
Assumptions	4-17
Operation XSD	4-17
Purchase Order Management Service	4-17
Functional Area	4-17
RSB Proxy WSDL	4-17
Merchandising Service WSDL	4-17
Overview	4-17
Operation XSD	4-18
Report Locator Service	4-19
Functional Area	4-19
RSB Proxy WSDL	4-19
Merchandising Service WSDL	4-19
Overview	4-19
Operation XSD	4-20
Store Order Service	4-20
Functional Area	4-20
RSB Proxy WSDL	4-20
Merchandising Service WSDL	4-20
Overview	4-20
Assumptions	4-21
Operation XSD	4-21
Supplier Service	4-22
Functional Area	4-22

RSB Proxy WSDL	4-22
Merchandising Service WSDL	4-22
Overview	4-22
Operation XSD	4-23
Transfer Service	4-23
Functional Area	4-23
RSB Proxy WSDL	4-23
Merchandising Service WSDL	4-23
Overview	4-23
Operation XSD	4-24
Consumer Services	4-24
Customer Address Service	4-24
Functional Area	4-24
Overview	4-24
Customer Order Address Service	4-25
Functional Area	4-25
Overview	4-25
Get Drill Back Forward URL Service	4-27
Functional Area	4-27
Overview	4-27
GL Chart of Accounts Validation Service	4-27
Functional Area	4-27
Overview	4-28

5 ReSTful Web Services

Introduction	5-1
Common Characteristics	5-2
Security	5-2
Standard Request and Response Headers	5-3
Standard Error Response	5-3
Merchandising URL Paths	5-3
Fiscal Document Generation URL Paths	5-4
Sales Audit URL Paths	5-4
Date Format	5-4
Paging	5-4
Web Service APIs Process Flow	5-5
Merchandising ReSTful Web Services	5-6
Merchandising Common Services	5-6
Vdate	5-6
Procurement Unit Options	5-7

Functional Config Options	5-9
Inventory Movement Unit Options	5-10
Currencies	5-13
Department Search	5-14
Department Load	5-15
Allocation Detail Service	5-16
Business Overview	5-16
Service Type	5-16
ReST URL	5-16
Input Parameters	5-16
Output	5-16
Table Impact	5-18
Background Process Configuration	5-18
Business Overview	5-18
Service Type	5-18
ReST URL	5-18
Input Parameters	5-18
Output	5-19
Table Impact	5-19
Book Transfer ReSTful Web Service	5-19
Functional Area	5-19
Business Overview	5-19
Service Type	5-19
ReST URL	5-19
Input Parameters	5-20
ItemDetail RDO	5-20
Example JSON Input	5-20
Output	5-21
JSON Structure	5-22
Table Impact	5-22
Cancel Customer Order Fulfillment Service	5-23
Functional Area	5-23
Business Overview	5-23
Service Type	5-24
ReST URL	5-24
Input Parameters	5-24
Output	5-25
Output	5-26
Code Detail Service	5-26
Business Overview	5-26
Service Type	5-26

ReST URL	5-27
Input Parameters	5-27
Output	5-27
Table Impact	5-27
Create Customer Order Fulfillment Service	5-28
Functional Area	5-28
Business Overview	5-28
Service Type	5-28
ReST URL	5-28
Input Parameters	5-29
Output	5-34
Output	5-35
Create Franchise Orders	5-36
Functional Area	5-36
Business Overview	5-36
Service Type	5-38
ReST URL	5-38
Input Payload Details	5-38
Response Code: 200 (Success)	5-41
Response Code: 400 (Error)	5-43
Create Inventory Transfer Services	5-44
Functional Area	5-44
Business Overview	5-44
Transfer Number	5-44
Search Items	5-45
Load Items	5-48
Search From Location	5-50
Search To Location	5-53
Load Locations	5-55
Create Transfer	5-57
Create Purchase Order Services	5-58
Order Number	5-59
Terms	5-59
Search Supplier	5-60
Load Supplier	5-63
Search Items	5-65
Load Items	5-68
Search Locations	5-71
Load Locations	5-73
Create Purchase Order	5-74
Currency Rates Service	5-76

Business Overview	5-76
Service Type	5-76
ReST URL	5-76
Input Parameters	5-76
Output	5-76
Table Impact	5-77
Data Privacy Access Service	5-77
Business Overview	5-77
Service Type	5-77
ReSTURL	5-77
Accept	5-77
Query Parameters	5-77
Path Parameters	5-77
Default Response	5-78
Sample Response	5-79
Response Codes and Error Messages	5-79
Success Payloads	5-79
Data Privacy Forget Service	5-80
Business Overview	5-80
Service Type	5-80
ReSTURL	5-80
Accept	5-80
Query Parameters	5-80
Path Parameters	5-80
Default Response	5-81
Response Codes and Error Messages	5-81
Success Payloads	5-81
Diff Detail Service	5-82
Business Overview	5-82
Service Type	5-82
ReST URL	5-82
Input Parameters	5-82
Output	5-82
Table Impact	5-83
EOW Run Signal Service	5-83
Business Overview	5-83
Service Type	5-83
ReST URL	5-83
Input Parameters	5-83
Output	5-83
Table Impact	5-84

Half Data Budget Service	5-84
Business Overview	5-84
Functional Area	5-84
Modify Half Data Budget	5-84
Inventory Adjustment Service	5-86
Functional Area	5-86
Business Overview	5-86
Inventory Adjustment	5-86
Item Detail Service	5-89
Business Overview	5-89
Service Type	5-89
ReST URL	5-89
Input Parameters	5-89
Output	5-89
Table Impact	5-92
Item Loc Inventory Detail Service	5-92
Business Overview	5-92
Service Type	5-93
ReST URL	5-93
Input Parameters	5-93
Output	5-93
Table Impact	5-95
MerchHierarchy Detail Service	5-95
Business Overview	5-95
Service Type	5-95
ReST URL	5-95
Input Parameters	5-95
Output	5-95
Table Impact	5-96
Obligations Upload Services	5-96
Function Area	5-96
Business Overview	5-97
Create Obligation	5-97
Create Obligation Component	5-101
Create Obligation Component Location	5-105
Modify Obligation	5-109
Modify Obligation Comp	5-113
Modify Obligation Component Location	5-117
Delete Obligation	5-121
Delete Obligation Component	5-124
Purchase Order Detail Service	5-128

Business Overview	5-128
Service Type	5-128
ReST URL	5-128
Input Parameters	5-128
Output	5-128
Table Impact	5-132
Recent Inventory Transfer Services	5-132
Functional Area	5-132
Business Overview	5-132
Transfer Location Search	5-132
Transfer Location Load	5-134
Transfer Status List	5-135
Transfer Type List	5-136
Search Transfer User IDs	5-137
Transfer Search	5-138
Get Transfer Detail	5-141
Update Transfer Status	5-143
Recent Purchase Order Services	5-144
Functional Area	5-144
Business Overview	5-144
Cancel Reason Code List	5-144
Origin Code List	5-145
Purchase Order Status List	5-146
Search Purchase Order User ID	5-147
Purchase Order Search	5-149
Get Purchase Order Summary	5-151
Get Purchase Order Items	5-153
Get Purchase Order Item Locations	5-155
Update Purchase Orders Date	5-157
Cancel Purchase Orders	5-158
Approve Purchase Orders	5-160
Reject Purchase Orders	5-164
Reclass Detail Service	5-165
Business Overview	5-165
Service Type	5-165
ReST URL	5-165
Input Parameters	5-166
Output	5-166
Table Impact	5-166
Replenishment Schedule Services	5-166
Functional Area	5-167

Business Overview	5-167
Create Replenishment Schedule	5-167
Modify Replenishment Schedule	5-171
Delete Replenishment Schedule	5-176
Shipment Detail Service	5-178
Business Overview	5-178
Service Type	5-178
ReST URL	5-178
Input Parameters	5-179
Output	5-179
Table Impact	5-180
Stock Count Detail Service	5-180
Business Overview	5-181
Service Type	5-181
ReST URL	5-181
Input Parameters	5-181
Output	5-181
Table Impact	5-182
Store Day User Service	5-183
Functional Area	5-183
Business Overview	5-183
Create Store Day User	5-183
Delete Store Day User	5-184
Store Detail Service	5-186
Business Overview	5-186
Service Type	5-186
ReST URL	5-186
Input Parameters	5-187
Output	5-187
Table Impact	5-189
Supplier Detail Service	5-189
Business Overview	5-189
Service Type	5-189
ReST URL	5-189
Input Parameters	5-189
Output	5-189
Table Impact	5-192
Tax Rules Services	5-192
Functional Area	5-192
Business Overview	5-192
Create Tax Rule	5-192

Update Tax Rule	5-197
Delete Tax Rule	5-202
Tran Data Upload Service	5-207
Functional Area	5-207
Business Overview	5-207
Service Type	5-207
ReST URL	5-207
Input Parameters	5-207
Output	5-209
Table Impact	5-209
Transfer Detail Service	5-210
Business Overview	5-210
Service Type	5-210
ReST URL	5-210
Input Parameters	5-210
Output	5-210
Table Impact	5-212
VAT Detail Service	5-212
Business Overview	5-212
Service Type	5-212
ReST URL	5-213
Input Parameters	5-213
Output	5-213
Table Impact	5-213
Warehouse Detail Service	5-214
Business Overview	5-214
Service Type	5-214
ReST URL	5-214
Input Parameters	5-214
Output	5-214
Table Impact	5-216
Fiscal Document Generation Services	5-216
Fiscal Pre Document Request Service	5-216
Functional Area	5-216
Business Overview	5-216
Service Type	5-216
ReST URL	5-216
Input	5-216
Output	5-221
Table Impact	5-222
Fiscal Document Request Service	5-223

Functional Area	5-223
Business Overview	5-224
Service Type	5-224
ReST URL	5-224
Input	5-224
Output	5-232
Table Impact	5-233
Fiscal Document Status Detail Request Service	5-234
Functional Area	5-234
Business Overview	5-234
Service Type	5-234
ReST URL	5-234
Input Parameters	5-235
Output	5-235
Table Impact	5-236
Next Fiscal Document Request Service	5-237
Functional Area	5-237
Business Overview	5-237
Service Type	5-237
ReST URL	5-237
Input Parameters	5-237
Output	5-238
Table Impact	5-250
Fiscal Document Process Status Request Service	5-250
Functional Area	5-250
Business Overview	5-250
Service Type	5-250
ReST URL	5-251
Input Parameters	5-251
Output	5-253
Table Impact	5-253
Fiscal Document Cancel Request Service	5-254
Functional Area	5-254
Business Overview	5-254
Service Type	5-254
ReST URL	5-254
Input	5-254
Output	5-256
Table Impact	5-256
Fiscal Document Detail Request Service	5-257
Functional Area	5-257

Business Overview	5-257
Service Type	5-257
ReST URL	5-257
Input Parameters	5-257
Output	5-258
Table Impact	5-269
Sales Audit ReSTful Web Services	5-270
Data Privacy Access Service	5-270
Business Overview	5-270
Service Type	5-270
ReSTURL	5-270
Accept	5-270
Query Parameters	5-270
Path Parameters	5-270
Default Response	5-271
Sample Response	5-271
Response Codes and Error Messages	5-272
Success Payloads	5-272
Data Privacy Forget Service	5-272
Business Overview	5-272
Service Type	5-272
ReSTURL	5-272
Accept	5-273
Query Parameters	5-273
Path Parameters	5-273
Default Response	5-274
Response Codes and Error Messages	5-274
Success Payloads	5-274
Get Rejected Transactions Since Previous Query	5-274
Functional Area	5-274
Business Overview	5-274
Sales Service	5-274
Get Store Aggregations	5-286
Business Overview	5-287
Service Type	5-287
ReST URL	5-287
Input Parameters	5-287
Output	5-287
Table Impact	5-288
Get Store Day Date Indicator	5-288
Business Overview	5-288

Service Type	5-288
ReST URL	5-288
Input Parameters	5-288
Output	5-288
Table Impact	5-289
Get Store Days	5-289
Business Overview	5-289
Service Type	5-289
ReST URL	5-289
Input Parameters	5-289
Output	5-289
Table Impact	5-290
Get Store Errors	5-290
Business Overview	5-291
Service Type	5-291
ReST URL	5-291
Input Parameters	5-291
Output	5-291
Table Impact	5-291
Load Transactions	5-291
Functional Area	5-291
Business Overview	5-292
Store Search	5-304
Business Overview	5-304
Service Type	5-304
ReST URL	5-304
Input Parameters	5-304
Output	5-304
Table Impact	5-305
Summary of Open Store Days	5-305
Business Overview	5-305
Service Type	5-305
ReSTURL	5-305
Input Parameters	5-306
Output	5-306
Table Impact	5-306
Summary of Errors	5-306
Business Overview	5-306
Service Type	5-306
ReST URL	5-306
Input Parameters	5-306

Output	5-306
Table Impact	5-307
Summary of Over/Short Amount	5-307
Business Overview	5-307
Service Type	5-307
ReST URL	5-307
Input Parameters	5-307
Output	5-307
Table Impact	5-308
Summary of Over/Short Count	5-308
Business Overview	5-308
Service Type	5-308
ReST URL	5-308
Input Parameters	5-308
Output	5-308
Table Impact	5-309

6 Scheduled Integration

File-based Integration Overview	6-1
Standard File Layouts	6-1
Detail-Only Files	6-1
Master and Detail Files	6-1
Integration Batch Wrapper FAQ	6-3
For input files, are there any specific file format names?	6-3
How and where will files be moved?	6-3
When will files be unzipped?	6-4
What will happen to the files once they are processed?	6-4
Will processed file be archived?	6-4
What will happen if the file is rejected?	6-4
How will rejected files be reprocessed?	6-4
File Movement Flow	6-5
Input/Output File Naming Convention	6-5
Bulk Data Integration Overview	6-7
Outbound Scheduled Integration	6-7
Foundation Data	6-7
Brand Publication API (BDI_Brand_Fnd_PF_From_RMS_EOW_JOB)	6-8
Calendar Publication API (BDI_Calendar_Fnd_PF_From_RMS_EOW_JOB)	6-9
Code Detail Publication API (BDI_CodeDetail_Fnd_PF_From_RMS_JOB)	6-10
Code Head Publication API (BDI_CodeHead_Fnd_PF_From_RMS_JOB)	6-11
Company-wide Closings and Company Closed Exceptions (BDI_CompanyClosed_Fnd_PF_From_RMS_JOB)	6-12

Currency Conversion Rates Publication API (BDI_CurrConvRates_Fnd_PF_From_RMS_EOW_JOB)	6-14
Delivery Slot Publication API (BDI_DeliverySlot_Fnd_PF_From_RMS_JOB)	6-15
Diff Group Export (export_diffgrp.ksh)	6-16
Diff Group Publication API (BDI_DiffGrp_Fnd_PF_From_RMS_JOB)	6-17
Diff ID and Type Export (export_diffs.ksh)	6-18
Diff ID Publication API (BDI_Diff_Fnd_PF_From_RMS_JOB)	6-19
Finisher Address Publication API (BDI_FinisherAddr_Fnd_PF_From_RMS_JOB)	6-20
Location Closed Publication API (BDI_LocClosed_Fnd_PF_From_RMS_JOB)	6-21
Merch Hierarchy Publication API (BDI_MerchHier_Fnd_PF_From_RMS_JOB)	6-22
Merchandise Hierarchy Export (export_merchhier.ksh)	6-23
Organization Hierarchy Publication API (BDI_OrgHier_Fnd_PF_From_RMS_JOB)	6-24
Organizational Hierarchy Export (export_orghier.ksh)	6-25
Partner Address Publication API (BDI_PartnerAddr_Fnd_PF_From_RMS_JOB)	6-26
Partner Org Unit Publication API (BDI_PartOrgUnit_Fnd_PF_From_RMS_JOB)	6-27
Partner Publication API (BDI_Partner_Fnd_PF_From_RMS_JOB)	6-28
Store Address Publication API (BDI_StoreAddr_Fnd_PF_From_RMS_JOB)	6-29
Store Hours Publication API (BDI_StoreHours_Fnd_PF_From_RMS_JOB)	6-30
Store Publication API (BDI_Store_Fnd_PF_From_RMS_JOB)	6-32
Stores Export (export_stores.ksh)	6-33
Supplier Address Publication API (BDI_SupplierAddr_Fnd_PF_From_RMS_JOB)	6-34
Sups Publication API (BDI_Supplier_Fnd_PF_From_RMS_JOB)	6-35
Tax Download - Brazil (taxdnld)	6-36
Ticket Download (tcktdnld)	6-38
UDA Publication API (BDI_Uda_Fnd_PF_From_RMS_JOB)	6-40
UDA Values Publication API (BDI_UdaValues_Fnd_PF_From_RMS_JOB)	6-41
UOM Class Publication API (BDI_UomClass_Fnd_PF_From_RMS_JOB)	6-42
UOM Conversion Publication API (BDI_UomConversion_Fnd_PF_From_RMS_JOB)	6-43
VAT Codes, Regions, and Rates Export (export_vat.ksh)	6-44
VAT Publication API (BDI_Vat_Fnd_PF_From_RMS_JOB)	6-45
Warehouse (BDI_Wh_Fnd_PF_From_RMS_JOB)	6-47
Warehouse Address Publication API (BDI_WhAddr_Fnd_PF_From_RMS_JOB)	6-48
Items	6-49
Item Image Publication API (BDI_ItemImage_Fnd_PF_From_RMS_JOB)	6-49
Item Location Export (export_itemloc.ksh)	6-50
Item Location Publication API (BDI_ItemLoc_Fnd_PF_From_RMS_JOB)	6-51
Item Master Export (export_itemmaster.ksh)	6-53
Item Master Publication API (BDI_ItemHdr_Fnd_PF_From_RMS_JOB)	6-54
Item Supplier Country Dimensions Publication API (BDI_ItSupCtryDim_Fnd_PF_From_RMS_JOB)	6-55
Item Supplier Country Publication API (BDI_ItSupCtry_Fnd_PF_From_RMS_JOB)	6-56

Item Supplier Manufacturing Country Publication API (BDI_ItSupManCtry_Fnd_PF_From_RMS_JOB)	6-57
Item Supplier Publication API (BDI_ItemSupp_Fnd_PF_From_RMS_JOB)	6-58
Item Supplier UOM Publication API (BDI_ItemSuppUom_Fnd_PF_From_RMS_JOB)	6-59
Item VAT Rates Export (export_itemvat.ksh)	6-60
Pack Item Publication API (BDI_PckitemBrkout_Fnd_PF_From_RMS_JOB)	6-61
POS Configuration Data to 3rd Party POS (poscdnld)	6-62
Price History Publication API (BDI_PriceHist_Fnd_PF_From_RMS_JOB)	6-68
Related Items Export (export_relitem.ksh)	6-69
Related Item Publication API (BDI_RelatedItem_Fnd_PF_From_RMS_JOB)	6-70
UDA Item Date Publication API (BDI_UdaltemDate_Fnd_PF_From_RMS_JOB)	6-71
UDA Item FF Publication API (BDI_UdaltemFF_Fnd_PF_From_RMS_JOB)	6-72
UDA Item LOV Publication API (BDI_UdaltemLov_Fnd_From_RMS_JOB)	6-73
Financials	6-74
Daily or Weekly Download of Stock Ledger Data (stlgdnld)	6-75
Finance General Ledger to RFI (BDI_RFI_FinGenLdgr_Tx_PF_From_RMS_JOB)	6-79
Fixed Deal Income (dealfinc)	6-80
Franchise Billing Extract (wfbillex.ksh)	6-81
Item/Location Daily Stock Ledger Transactions (fifgldn1)	6-87
Monthly Stock Ledger Transactions (fifgldn3)	6-89
Open to Buy Download Stock Ledger (otbdlsal)	6-90
Rolled Up Daily Stock Ledger Transactions (fifgldn2)	6-99
Stage G/L Extracts (gl_extract.ksh)	6-101
Tran Data Publication (BDI_TranData_Tx_PF_From_RMS_EOW_JOB)	6-103
Ordering and Inventory	6-104
Purchasing	6-104
Import Management	6-127
Invoices	6-160
Inventory	6-172
Planning and Forecasting	6-182
Brand Extract to Planning (BDI_RPAS_Brand_Fnd_PF_From_RMS_JOB)	6-184
Calendar Extract to Planning and Forecasting (BDI_RPAS_Calendar_Fnd_PF_From_RMS_JOB)	6-186
Currency Rates Extract to Planning and Forecasting (BDI_RPAS_CurrConvRates_Fnd_PF_From_RMS_JOB)	6-189
Differentiator Extract to Planning (BDI_RPAS_Diff_Fnd_PF_From_RMS_JOB)	6-192
Inventory Extract to Planning (BDI_MFP_Inventory_Tx_PF_From_RMS_JOB)	6-194
Merchandise Hierarchy and Item Extract to Planning and Forecasting (BDI_RPAS_MerchHier_Fnd_PF_From_RMS_JOB)	6-197
On Order Extract to Planning (BDI_MFP_OnOrder_Tx_PF_From_RMS_JOB)	6-202
Organization Hierarchy Extract to Planning and Forecasting (BDI_RPAS_OrgHier_Fnd_PF_From_RMS_JOB)	6-205

Out of Stock Extract to Forecasting (BDI_RDF_StockOut_Tx_PF_From_RMS_JOB)	6-209
Store Extract to Planning and Forecasting (BDI_RPAS_Store_Fnd_PF_From_RMS_JOB)	6-212
Supplier Extract to Planning (BDI_RPAS_Supplier_Fnd_PF_From_RMS_JOB)	6-215
Transaction Data Extract to Planning (BDI_MFP_TranData_Tx_PF_From_RMS_JOB)	6-217
UDA Extract to Planning (BDI_RPAS_UdaAndUdaValues_Fnd_PF_From_RMS_JOB)	6-223
UDA Item Extract to Planning and Forecasting (BDI_RDF_UdaltemLov_Fnd_From_RMS_JOB)	6-226
Weekly Sales Extract to Forecasting (BDI_RDF_WeeklySales_Tx_PF_From_RMS_JOB)	6-229
Sales Audit	6-231
Download from Sales Audit to Account Clearing House (ACH) System (saexpach)	6-232
Download of Escheated Vouchers from Sales Audit for Payment (saescheat)	6-238
Export DSD and Escheatment from Sales Audit to Invoice Matching (saexpim)	6-239
Export from Sales Audit to Oracle Retail Analytics (saexpdw)	6-240
Export Inventory Reservation/Release for In Store Customer Order & Layaway Transactions (saordinvexp)	6-271
Export of POS Transactions from Sales Audit to Merchandising (saexprms)	6-274
Export of Revised Sale/Return Transactions from ReSA to SIM/SIOCS (saexpsim)	6-279
Export to Universal Account Reconciliation System from Sales Audit (saexpuar)	6-283
Extract of POS Transactions by Store/Date from Sales Audit for Web Search (ang_saplgen)	6-285
Post User Defined Totals from Sales Audit to General Ledger (saexpgl)	6-286
Inbound Scheduled Integration	6-287
Item, Cost, and Price	6-287
Upload Competitor's Prices (cmpupld)	6-288
Upload Items and Cost Changes (iindbatch.ksh)	6-292
Ordering and Inventory	6-293
Purchasing	6-293
Import Management	6-324
Stock Counts	6-357
Franchise	6-363
Other Inventory	6-383
Sales Processing	6-392
Sales Audit	6-392
Sales Posting	6-451
Forecasting	6-459
Daily Demand Item Forecast Subscription API	6-460
Weekly Demand Item Forecast Subscription API	6-460

Send Us Your Comments

Oracle® Retail Merchandising Operations Guide Volume 2 - Message Publication and Subscription Designs

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

The *Oracle Retail Operations Guides* are designed so that you can view and understand the applications's 'behind-the-scenes' processing.

The *Oracle Retail Merchandising Foundation Cloud Service Operations Guide, Volume 2 - Message Publication and Subscription Designs* provides critical information about the processing and operating details of Oracle Retail Merchandising System (RMS), including the following:

- Publication designs which describe, on a technical level, how Merchandising publishes messages.
- Subscription designs which describe, on a technical level, how Merchandising subscribes to messages.

Audience

This guide is for:

- Systems administration and operations personnel
- Systems analysts
- Integrators and implementers
- Business analysts who need information about Merchandising System processes and interfaces

Documentation Accessibility

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

Access to Oracle Support

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

Related Documents

For more information, see the following documents in the Oracle Retail Integration Bus documentation set:

- *Oracle Retail Merchandising Foundation Cloud Service Release Notes*

- *Oracle Retail Merchandising Foundation Cloud Service Operations Guide, Volume 1 - Batch Overviews and Designs*
- *Oracle Retail Merchandising Foundation Cloud Service Administration Guide*
- *Oracle Retail Merchandising Foundation Cloud Service Implementation Guide*
- *Oracle Retail Merchandising Foundation Cloud Service Deals and Cost Changes User Guide*
- *Oracle Retail Merchandising Foundation Cloud Service Do the Basics Changes User Guide*
- *Oracle Retail Merchandising Foundation Cloud Service Finance User Guide*
- *Oracle Retail Merchandising Foundation Cloud Service Foundation Data User Guide*
- *Oracle Retail Merchandising Foundation Cloud Service Franchise User Guide*
- *Oracle Retail Merchandising Foundation Cloud Service Inventory User Guide*
- *Oracle Retail Merchandising Foundation Cloud Service Items User Guide*
- *Oracle Retail Merchandising Foundation Cloud Service Pricing User Guide*
- *Oracle Retail Merchandising Foundation Cloud Service Purchase Orders and Contracts User Guide*
- *Oracle Retail Merchandising Foundation Cloud Service Replenishment User Guide*

Customer Support

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

<https://support.oracle.com>

When contacting Customer Support, please provide the following:

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

Improved Process for Oracle Retail Documentation Corrections

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

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

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

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

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

Oracle Retail Documentation on the Oracle 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.
monospace	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 volume contains details about Merchandising and Sales Audit integrations. These integrations fall into four main categories:

- **Message-based Integration** - These are covered in two sections: publication and subscription. Publication covers RIB messages published from Merchandising to other solutions. Subscription covers RIB message that are subscribed to by Merchandising from other solutions.
- **SOAP Web Services** - This chapter provides a summary of the provider and consumer SOAP services supported by Merchandising and Sales Audit, including details on security, URLs, and payload information.
- **ReSTful Web Services** - This chapter provides a summary of the ReST services supported by Merchandising and Sales Audit, including details on security, URLs, and payload information.
- **Scheduled Integration** - This chapter provides a summary of integrations that are scheduled either to be run once per day or periodically throughout the day. There are generally two types of integrations - those that expect or produce files and those that move data between integration tables, also referred to as Bulk Data Integration (BDI).

2

RIB Publication Designs

This chapter provides an overview of the RIB publication APIs used in by Merchandising.

Allocation Publication API

This section describes the allocations publications API.

Functional Area

Allocations

Business Overview

Merchandising is responsible for communicating allocation information with external systems such as a store inventory system (SIOCS, for example) and a warehouse management system (like Oracle WMS Cloud).

There are several ways in which allocation information can be created in Merchandising:

- Through integration with the Allocation Cloud Service
- Through Merchandising replenishment, where cross dock orders generate allocations
- Through the Allocation Subscription API, where a third-party system can create allocations and send to Merchandising for execution

Allocations can be created from a virtual warehouse to any type of stockholding location in Merchandising, including other virtual warehouses, and to both company and franchise stores. Allocations include a store type and stockholding indicator at the detail level when allocating to stores, to allow the store and warehouse inventory management system to filter out the data irrelevant to their respective systems. When allocating to a franchise store, the linked franchise orders are not published; only the allocation itself is published. When allocating to another warehouse, the allocation quantities are summed up to the physical warehouse level and the physical warehouse is what is communicated in the integration.

An allocation and its details are not published from Merchandising until it is approved. Modified and deleted allocation information is also sent. Allocation header modification messages will be sent if the status of the allocation is changed to approved (A) or closed (C) or if the allocation release date is changed. Allocation detail modification messages will be sent if the allocated quantity is changed. A header delete message signifies that the completed allocation has been deleted.

Note:

Allocations, when published to external systems, are combined in the RIB with transfers (published in the Transfer Publication API) into a combined Stock Order Publication message.

New Allocations

When an allocation is created, an Allocation Create message request is queued. The Allocation Create message is a flat message containing a full snapshot of the allocation at the time the message is published. The message will not be sent until the allocation has been approved.

The allocation create message contains the following:

Allocation Header

Message Element	Included?	Notes
Allocation Number	Always	Contains the number that uniquely identifies the allocation within the system.
Document Type	Always	This is the type of stock order. This value will be 'A' to signify an allocation. This will differentiate allocations and transfers in the Stock Order Publication API subscribed to by stores and warehouses.
Physical WH	Always	The physical warehouse location from which the allocation will originate.
Warehouse	Always	Contains the number of the virtual warehouse where the allocation will originate.
Item	Always	Specifies the item on the allocation.
Pick Not Before Date	Optional	Contains the earliest date on which the allocation should start being picked for the allocation. This will contain the release date on the allocation or the not before date on the associated purchase order.
Pick Not After Date	Optional	This value is calculated by adding x days to the release date on the allocation. The number of days to add is defined in the description column on code_detail for code DATE and code type DEFT.

Message Element	Included?	Notes
Order Type	Optional	This field contains the type of order. Allocations created against Purchase Orders will be marked as PREDIST order types. Allocations created against Warehouse stock will be populated with the DEFAULT_ORDER_TYPE from the SYSTEM_OPTIONS table which can be AUTOMATIC, MANUAL or WAVE.
Order No	Optional	Contains the order number to which the allocation applies. This is only populated for order-based allocations.
Order Document Type	Always	Always 'P'.
Event	Optional	This field contains the event to which the promotion belongs to. This is an optional field that provides a method to group promotions together for tracking and reporting purposes.
Event Description	Optional	This field contains the description of the promotional event.
Priority	Always	A value which indicates the priority of an allocation. This value will always be 1.
Ticket Type ID	Optional	This field contains a character string which uniquely identifies the ticket type which is associated with the item. This is only populated if the ticket associated with the item on the allocation is configured to be printed upon the receipt of the purchase order for order-based allocations.
Context Type	Optional	This field holds the code for the reason for the allocation. For example, it could indicate the allocation was created for an upcoming promotion. Valid context codes are defined in Codes and Descriptions under the code type CNTX.
Context Value	Optional	Contains a character string relating to the context type.
Allocation Status	Always	Contains the code for the allocation status.
Allocation Details	Always	Child node

Allocation Detail

Message Element	Required?	Notes
Physical To Location	Always	This field contains the physical location to which the allocation is being sent. If the location type is a store, this will be the same as the To Location.
To Location	Always	This field contains the location to which the allocation is being sent. The location type field determines if the location is a store or a virtual warehouse.
Location Type	Always	This field contains the type of location in the To Location field. Valid values are: S - Store W - Warehouse
Store Type	Optional	If the To Location is a store, this field contains the type of store. Valid values are: C - company store F - franchise store
Stockholding	Optional	If the To Location is a store, this field indicates if the store is a stockholding location or not. Valid values are: yes (Y) or no (N).
Quantity Allocated	Always	Contains the total quantity of the item allocated to the to location.
Price	Optional	This field holds the unit retail in the selling unit of measure for the item/to location combination. This field is stored in the local currency of the to location.
Selling UOM	Optional	This field holds the selling unit of measure for the item's single-unit retail at the to location.
Priority	Always	A value which indicates the priority of an allocation detail. This value will always be 1.
Store Order Multiple	Always	This column contains the multiple in which the item needs to be shipped from a warehouse to the store.
In-Store Date	Optional	Indicates the date that the item needs to be at the store.

Message Element	Required?	Notes
Rush Flag	Optional	Indicates if there is a rush on shipping this item to the destination location.
Allocation Detail Component Items	Optional	Child node

Allocation Detail Component Items

This node is only included if the item being allocated is a pack item.

Message Element	Required?	Notes
Component Item	Always	This field contains the alphanumeric identifier of an item within the pack.
Component Price	Always	This field holds the unit retail in local currency and in the component selling unit of measure for the component item at the destination location.
Component Selling UOM	Always	This field holds the selling unit of measure for the item's single-unit retail at the to location.

New Allocation Details

Creating new allocations details for an existing header triggers a message to be sent to notify external systems of the changes. The message for new allocation details contains the same information as the new allocation create message.

Updated Allocations

Allocation updates trigger a message to be sent to notify external systems based on updates made at the allocation header and/or detail level. The message for updated allocations contains the same information as the new allocation create message.

Header Only Updates

Allocation updates trigger a message to be sent to notify external systems based on updates made at the allocation header and/or detail level. The message for updated allocations contain the same information as the new allocation create message.

Detail Only Updates

When updates are made at the allocation detail level without changes to the header information, the full header and detail information are published. A detail update will be published when the quantity allocated has changed.

Full Message Updates

In cases where the system receiving allocations cannot support just receiving the changes, another option is provided that can resend the full allocation details whenever there is a change. This will be published when the system option Publish Full Objects (PUB_FULL_OBJECTS_IND) is set to Deltas and Full (Y). This message will contain the allocation header and its detail level information. This is used for Oracle WMS Cloud integration.

Deleted Allocations

Allocations can be deleted when they are in Approved or Closed status. When an allocation delete message is triggered, a message is sent to external systems to notify them of the changes through an allocation header delete message.

Detail Only Deletes

When allocation details are deleted, a message is sent to external systems to notify them of the change. Both header and detail information are included in the published message.

The allocation delete message contains the following:

Allocation Header

Message Element	Included?	Notes
Allocation Number	Always	Contains the numeric identifier that uniquely identifies the allocation within the system.
Document Type	Always	This is the type of stock order. This value will be 'A' to signify an allocation. This will differentiate allocations and transfers in the Stock Order Publication API subscribed to by stores and warehouses.
Physical Warehouse	Always	Contains the numeric identifier of the physical warehouse location where the allocation will originate.
Warehouse	Always	Contains the numeric identifier of the virtual warehouse location where the allocation will originate.
Item	Always	Unique alphanumeric value that identifies the item.
Allocation Details	Optional	Child node

Allocation Detail

Message Element	Required?	Notes
To Location	Always	This field contains the location to which the allocation is being sent. The location type field determines if the location is a store or a virtual warehouse.
Location Type	Optional	This field contains the type of location in the To Location field. Valid values are S - store and W - warehouse.
Store Type	Optional	If the To Location is a store, this field contains the type of store. Valid values are: C - company store, F - franchise store.
Stockholding Indicator	Optional	If the To Location is a store, this field indicates if the location is a stockholding store or not. Valid values are: yes (Y) or no (N).

Error Handling

When the publication encounters a non-fatal error, messages continue to be processed. For the particular message where the error was encountered, a status of Hospital (H) is sent to the RIB and the status of the message in the queue is set to H. These types of errors occur when no changes in the database have been made and a process to try to re-publish these messages is available. In case the error is a fatal error—for example, when changes to data have already been made—a status of Error (E) is sent to the RIB and the message status in the queue will be in Error status. The error message as well as the object containing the allocation number and details is returned to Merchandising.

Message XSD

Here are the filenames that correspond with each message type. Please consult the mapping documents for each message type in order to get a detailed picture of the composition of each message.

Message Types	Message Type Description	XML Schema Definition (XSD)
AllocCre	Allocation Create Message	AllocDesc.xsd
AllocDtlCre	Allocation Detail Create Message	AllocDesc.xsd
AllocHdrMod	Allocation Header Modify Message	AllocDesc.xsd
AllocDtlMod	Allocation Detail Modify Message	AllocDesc.xsd
AllocFulRep	Allocation Full Replacement Message	AllocDesc.xsd
AllocDel	Allocation Delete Message	AllocRef.xsd
AllocDtlDel	Allocation Detail Delete Message	AllocRef.xsd

ASN Outbound Publication API

This section describes the ASNOUT Publication API.

Functional Area

Inventory

Business Overview

The Advanced Shipping Notification (ASN) outbound message is used to communicate the shipment of merchandise against transfers or allocations. This message is published by Merchandising to stores or warehouses.

Merchandising supports the following shipping functionality:

- On-line Shipping/Receiving.
- Franchise Order Shipment and Return.

On-line Shipping/Receiving

Two system options (ship_rcv_store and ship_rcv_wh) are used to control whether Merchandising on-line shipment/receiving functionality is enabled.

- Ship_rcv_store = 'Y' means a store inventory management application, such as Oracle Retail Store Inventory Operations Cloud Service (SIOCS), is NOT installed and shipping/receiving for stores will be done in Merchandising.
- Ship_rcv_wh = 'Y' means a warehouse management system, such as Oracle Cloud WMS, is NOT installed and shipping/receiving for warehouses will be done in Merchandising.

If either (but not both) of these indicators is set to 'Y', shipments created in Merchandising should be published to the RIB to allow the integration subsystem application to have visibility to the corporately created shipment.

The possible scenarios for on-line shipping/receiving:

Store Inventory System Installed (Yes/No)	Warehouse Management Installed (Yes/No)	System Options Settings	Merchandising Publishes Shipments (Yes/No)	Apps to subscribe to the message
Yes	Yes	Ship_rcv_store = N Ship_rcv_wh = N	No	No
No	No	Ship_rcv_store = Y Ship_rcv_wh = Y	No	No
Yes	No	Ship_rcv_store = N Ship_rcv_wh = Y	Yes - for warehouse-to-store shipments	Store Inventory System

Store Inventory System Installed (Yes/No)	Warehouse Management Installed (Yes/No)	System Options Settings	Merchandising Publishes Shipments (Yes/No)	Apps to subscribe to the message
No	Yes	Ship_rcv_store = Y Ship_rcv_wh = N	Yes - for store-to-warehouse shipments	Warehouse Management System

Merchandising on-line shipping can involve a customer order transfer. For a customer order transfer, customer order number, and fulfillment order number are included in the published information.

Franchise Order Shipment and Return

Franchise stores are a special kind of stores that are not 'owned' by the company; therefore, any shipment to a franchise store is considered a sale. From Merchandising, franchise stores can order goods from company stores or warehouses; they can also return goods back to company stores or warehouses. These orders and returns are created as transfers in Merchandising.

Merchandising supports two kinds of franchise stores - stockholding franchise stores (which Merchandising manages inventory and financials like regular stores) and non-stockholding franchise stores (which Merchandising does NOT manage inventory and financials).

The Store Inventory Operations Cloud Service (SIOCS) manages transactions for stockholding franchise stores, but not for non-stockholding franchise stores. The Shipping and Receiving of non-stockholding franchise orders and returns are handled within Merchandising from the Store perspective even if SIOCS is installed.

For warehouses, if a franchise return from a non-stockholding franchise store is to be processed, the warehouse management system (WMS) will require an ASN against which to receive. Since Merchandising automatically creates the shipment for non-stockholding stores upon the approval of a franchise return, Merchandising needs to publish those shipments for the WMS. Similar to on-line Shipping/Receiving, Merchandising publishes shipments of non-stockholding Franchise Returns to warehouses as ASNOut messages.

Advance Shipment Notification

Shipments created in Merchandising cannot be modified. Upon saving a shipment, the entire shipment is published from Merchandising as one ASNOut message. As a result, Merchandising only needs to support the ASNOut create message type for shipment publishing.

Table 2-1 Advance Shipment Notification Header

Message Element	Included?	Notes
Schedule Number	No	Contains the Schedule identifier number. Not used by RMS.
Auto Receive	Always	Contains the flag to enable auto-receiving in the store inventory system (e.g. SIOCS). Defaulted to "No".

Table 2-1 (Cont.) Advance Shipment Notification Header

Message Element	Included?	Notes
To Location	Always	Contains the location that the shipment will be delivered to.
To Location Type	Always	This field contains the location type of the to_location field. Valid values are 'S' = store, 'E' = finisher, and 'W' = warehouse.
To Store Type	Always	Indicates the store type of the to_location. This will only be populated if to_loc_type is 'S'. Valid values are: 'C' = company store, 'F' = franchise store.
To Stockholding Indicator	Optional	Indicates if the to_location is stockholding or not. Only populated if to_loc_type is 'S'. Valid values are: 'Y' = yes, 'N' = no.
From Location	Always	For transfer and allocation shipments, this field will hold the location from which the shipment was sourced.
From Location Type	Always	This field contains the location type of the from_location field. Valid values are 'S' = store and 'W' = warehouse.
From Store Type	Optional	Indicates the store type of the from_location. This will only be populated if from_location is 'S'. Valid values are: 'C' = company store, 'F' = franchise store. Publications from RWMS will always be NULL for this field.
From Stockholding Indicator	Optional	Indicates if the from_location is stockholding or not. Only populated if from_location is 'S'. Valid values are: 'Y' = yes, 'N' = no. Publications from RWMS will always be NULL for this field.
ASN Number	Always	Holds the bill of lading number associated with a shipment.
ASN Type	Always	Not used by RMS. Set to C for container when ship to location is a store, T for tare when ship to location is a warehouse.
Container Quantity	Always	Contains the number of boxes associated with the shipment.
BOL Number	Always	Holds the transaction sequence number on the message from the transfer shipment confirmation process.
Shipment Date	Always	This field contains the date the transfer or PO was shipped.
Estimated Arrival Date	Always	This field contains the estimated arrival date of a vendor PO shipment. It is updated by EDIUP856. It is used for vendor/lead time analysis.
Ship Address 1	Optional	Value of addr.add_1 for store/warehouse number in addr.key_value_1 and addr.module 'ST' (if to_loc_type is store) or 'WH' (if to_loc_type is warehouse).
Ship Address 2	Optional	Value of addr.add_2 for store/warehouse number in addr.key_value_2 and addr.module 'ST' (if to_loc_type is store) or 'WH' (if to_loc_type is warehouse).
Ship Address 3	No	Not used by RMS.

Table 2-1 (Cont.) Advance Shipment Notification Header

Message Element	Included?	Notes
Ship Address 4	No	Not used by RMS.
Ship Address 5	No	Not used by RMS.
Ship City	Optional	Value of addr.city for store/warehouse number in addr.key_value_1 and addr.module 'ST' (if to_loc_type is store) or 'WH' (if to_loc_type is warehouse).
Ship State	Optional	Value of addr.state for store/warehouse number in addr.key_value_1 and addr.module 'ST' (if to_loc_type is store) or 'WH' (if to_loc_type is warehouse).
Ship ZIP	No	Not used by RMS.
Ship Country Id	Optional	Value of addr.country_id for store/warehouse number in addr.key_value_1 and addr.module 'ST' (if to_loc_type is store) or 'WH' (if to_loc_type is warehouse).
Trailer Number	Always	Not used by RMS. Defaulted to 1.
Seal Number	No	Not used by RMS.
Trans Shipment Number	No	Not used by RMS.
Comments	Optional	Contains any miscellaneous comments about the shipment.
Carrier Code	Always	Contains the courier that will deliver the shipment. Defaulted to "DC".
Carrier Service Code	No	Contains the service level code for the courier that will deliver the shipment.
System Code	No	The code identifying the system associated with the location.
From Location Warehouse	No	For transfer and allocation shipments, this field will hold the virtual warehouse from which the shipment was sourced.

Table 2-2 Advance Shipment Notification Distro

Message Element	Included?	Notes
Distro Number	Always	Specifies the transfer or allocation number associated with the shipment/item/carton.
Distro Doc Type	Always	Specifies what the distro_nbr field corresponds to in RMS: V, D, and T specify transfer, A specifies Allocation.
Customer Order Number	Optional	This is the customer order number that was generated by OMS and contains multiple fulfillment numbers.
Fulfill Order Number	Optional	Based on a customer order - OMS will generate fulfillments to specific locations based on availability. RMS will generate Transfers based on the fulfillment request.
Consumer Direct	No	Not used by RMS.
Comments	No	Not used by RMS.

Table 2-3 Advance Shipment Notification Carton

Message Element	Included?	Notes
Final Location	No	Not used by RMS.
Container Id	Always	Identifies the UCC-128 carton number for shipments originating from the Advance Shipment Notification process as carton shipments. This field will be zero for all shipments that are not at a carton level.
Container Weight	No	Not used by RMS.
Container Length	No	Not used by RMS.
Container Width	No	Not used by RMS.
Container Height	No	Not used by RMS.
Container Cube	No	Not used by RMS.
Expedite Flag	Always	Indicates if the shipment should be expedited. For transfers, the value will be 'Y' if the transfer's freight_code is 'E'; otherwise 'N'. For allocations, the field is mapped to alloc_detail.rush_flag.
In Store Date	Always	Contains the date to be included in the RMS publication to the RIB for communication to the warehouse. For transfers, the value is always NULL. For allocations, the value is alloc_detail.in_store_date.
Tracking Number	No	This is a unique tracking number that is used to track containers through carrier's system.
Freight Charge	No	Not used by RMS.
Master Container Id	No	Not used by RMS.
Comment	No	Contains any comments about the shipment container.
Weight	No	Actual weight shipped for the container.
Weight Unit of Measure	No	Unit of measurement for weight (e.g. pounds, kilograms) that was shipped.
Carrier Shipment Number	No	This field represents the shipment number that manifest systems use to group multi-container shipments for a carrier. Typically a 1 of 4 type message.
Original Item Id	No	The ID of the item being replaced. Populated only when this record is for a substitute item on a customer order.

Table 2-4 Advance Shipment Notification Item

Message Element	Included?	Notes
Item Id	Always	Unique identifier for the item.
Unit Quantity	Always	Contains the number of items expected to be received based on the supplier's advance shipment notification for this item/shipment combination.
Gross Cost	No	Contains the gross cost.

Table 2-4 (Cont.) Advance Shipment Notification Item

Message Element	Included?	Notes
Priority Level	No	Not used by RMS.
Order Line Number	No	Is used to carry the customer order line number value for customer orders. Is a derived value for non-customer orders.
Lot Number	No	Not used by RMS.
Final Location	Always	The destination location of the shipment.
From Disposition	No	This value is used to determine if the inventory is available or unavailable, based on the code's INV_STATUS value on the INV_STATUS_CODES table.
To Disposition	No	Not used by RMS.
Voucher Number	No	Contains the voucher number.
Voucher Expiration Date	No	Not used by RMS.
Container Quantity	Always	Not used by RMS. Defaulted to 1.
Comments	No	Contains any comment about the item in the shipment.
Unit Cost	Always	Contains the unit cost of the item in the shipment.
Base Cost	No	This value will be used to get the base cost (BC) from RFM for a transfer, which will flow into RMS.
Weight	No	Actual weight shipped.
Weight Unit of Measure	No	Unit of measurement for weight (e.g. pounds, kilograms) shipped.

Table 2-5 Advance Shipment Notification UIN

Message Element	Included?	Notes
UIN	No	Universal Identification Number

Error Handling

When the publication encounters a non-fatal error, messages continue to be processed. For the message where the error was encountered, a status of Hospital (H) is sent to the RIB and the status of the message in the queue is set to H. In case the error is a fatal error, a status of Error (E) is sent to the RIB and the next message in the queue is not retrieved until the error is resolved.

Message XSD

Here is the filename that corresponds with the message type. Please consult the RIB documentation for this message type to get a detailed picture of the composition of the message.

Message Types	Message Type Description	XML Schema Definition (XSD)
asnoutcre	ASN Out Create Message	ASNOutDesc.xsd

Available Inventory for Store and Warehouse Publication API

This section describes the Store and Warehouse Publication API.

Functional Area

Inventory

Business Overview

Merchandising publishes store and warehouse available inventory information to support the order management system requirements for ordering. Internally, Oracle Retail Order Broker (OROB) subscribes to this information. The following criteria must be met in order to publish inventory information when the stock on hand on a store or warehouse is updated:

- Items must be sellable
- Items must be a customer orderable location (based on the flags at the store and virtual warehouse level)
- Locations must be stockholding

This API always publishes inventory for virtual warehouses but will publish store information only based on the setting of the system option `Publish Store Available to Sell Updates`. If checked, then store information will also be published.

Item-Location Level Available to Sell Message

Message Element	Included?	Notes
Item	Always	The transaction item whose inventory information is being communicated.
Location	Always	Store or virtual warehouse where the item is to be found.
Location Type	Always	This contains the type of location. Valid values are store (S) or warehouse (W).

Message Element	Included?	Notes
Available to Sell Quantity	Always	Contains the updated quantity available to sell for the item-location in the standard unit of measure. Available to sell is calculated as: Stock on Hand - (Transfer Reserved + Customer Reservations + Non-sellable + return to vendor + in-progress sales quantity) All values above include also the pack component quantity, where applicable.
System Code	Always	Identifies the originating application, which is required by some subscribing systems. This value is defaulted from the system option Integration System Code.

Error Handling

If any errors are encountered while publishing the message, a fatal error with status E (Error) is sent to RIB. This would bring down the RIB adapter thus preventing any further messages from being processed until this is resolved.

Message XSD

Here are the filenames that correspond with each message type. Please consult the RIB documentation for each message type in order to get a detailed picture of the composition of each message.

Message Types	Message Type Description	XML Schema Definition (XSD)
COInvAvailMod	Customer Order Store Inventory Available	COInvAvailDesc.xsd

Banner and Channel Publication API

This section describes the banners and channels publication API.

Functional Area

Foundation

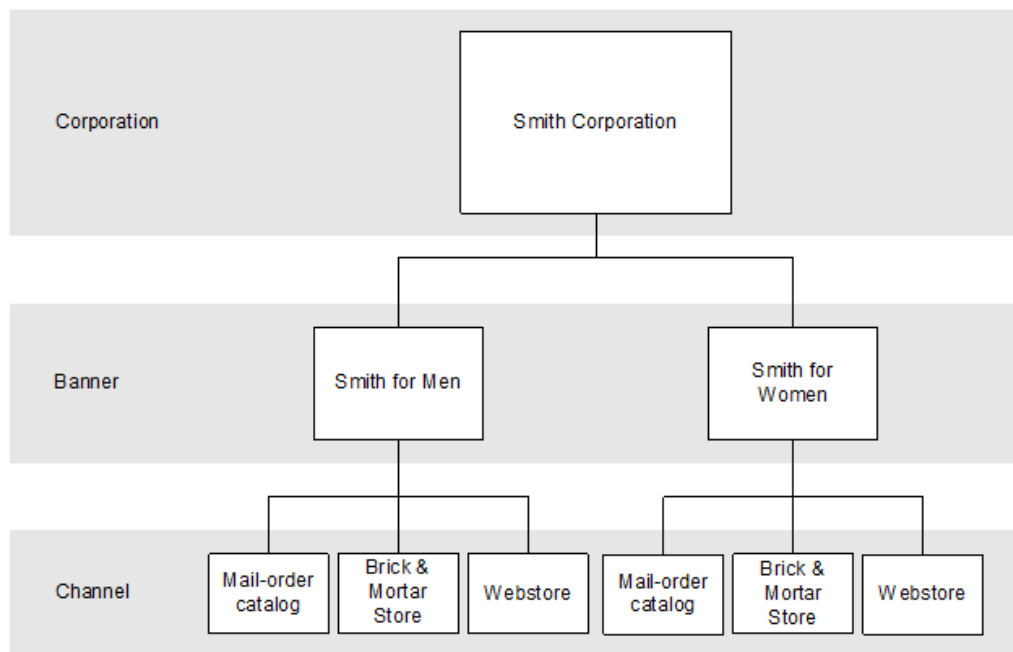
Business Overview

Merchandising publishes details about new and updated banners and channels in order that external systems that use this information can be informed of the updates. Updates are provided synchronously in a near-real time manner. A banner usually represents the name on

your “brand” name for your retail stores, and a channel is meant to represent the selling channels used by that banner, such as brick and mortar stores, web store, outlet, and so on.

The following diagram shows a sample of the structure of banners and channels within a retail organization.

Figure 2-1 Banners and Channels Within a Corporation



New Banners

Creating a new banner triggers a message to be sent through the RIB to notify external systems. The full details are sent for the new banner as part of the create message, the ID and description.

Updated Banners

When an existing banner is updated, an update message is triggered to provide the details of the update through the RIB. The update message, like create, will contain the full details in the message for all fields in the message.

Table 2-6 Banners – Create and Update

Message Element	Required?	Notes
Banner	Always	The unique number to identify a banner.
Banner Name	Always	The name of the banner in the primary language.

Deleted Banners

When an existing banner is deleted, this will also trigger a delete transaction to be sent through the RIB to notify external systems that this banner is no longer valid. The delete message will include only the ID of the banner being deleted.

Table 2-7 Banners – Delete

Message Element	Required?	Notes
Banner	Always	The number of the banner being deleted.

New Channels

Creating a new channel triggers a message to be sent through the RIB to notify external systems. The full details are sent for the channel as part of the create message, including the ID and description for the channel, the banner in which the channel belongs, and a channel type.

Updated Channels

When an existing channel is updated, an update message is triggered to provide the details of the update through the RIB. The update message, like create, will contain the full details in the message for all fields in the message.

Table 2-8 Channels – Create and Update

Message Element	Required?	Notes
Channel	Always	The unique number used to identify the channel.
Channel Name	Always	The name of the channel in the primary language.
Channel Type	Always	Indicates the type of channel (e.g., Brick & Mortar, Ecommerce, and so on). The channel type sent is just the code; the descriptions for these codes are not sent but can be found on the Codes and Descriptions under code type CHTY.
Banner	Always	The number of the banner in which the channel exists.

Deleted Channels

When an existing channel is deleted, this will also trigger a delete transaction to be sent through the RIB to notify external systems that this banner is no longer valid. The delete message will include only the ID of the channel being deleted.

Table 2-9 Channels – Create and Update

Message Element	Required?	Notes
Channel	Always	The number of the channel that is being deleted.

Error Handling

When the publication encounters a non-fatal error, messages continue to be processed. For the message where the error was encountered, a status of Hospital (H) is sent to the RIB and the status of the message in the queue is set to H. These types of errors occur when no changes in the database have been made and a process to try to re-publish the messages is available. In case the error is a fatal error, a status of Error (E) is sent to the RIB and the next message in the queue is not retrieved until the error is resolved.

Message XSD

Below are the filenames that correspond with each message type. Please consult the Oracle Retail Integration Guide for each message type for the details on the composition of each message.

Message Types	Message Type Description	XML Schema Definition (XSD)
BannerCre	Banner Create	BannerDesc.xsd
ChannelCre	Channel Create	ChannelDesc.xsd
BannerMod	Banner Modify	BannerDesc.xsd
ChannelMod	Channel Modify	ChannelDesc.xsd
BannerDel	Transfer Delete	BannerRef.xsd
ChannelDel	Channel Delete	ChannelRef.xsd

Brand Publication API

Functional Area

Foundation Data

Business Overview

This API publishes the creation, update and deletion of brand information to the RIB such that all the downstream applications (including an external system) may subscribe to it and have information in sync with Merchandising. Publishing is provided synchronously in a near-real-time manner.

New Brand

Creating a new brand triggers a message to be sent to notify external systems. Brand name and description are sent as part of the create message.

Table 2-10 Brand – Create and Update

Message Element	Included?	Notes
Brand name	Always	This field contains the brand name.

Table 2-10 (Cont.) Brand – Create and Update

Message Element	Included?	Notes
Brand Description	Always	This field contains the description of the brand.

Updated Brand

Updating a brand triggers a message to be sent to notify external systems. Brand name and description are sent as part of the update message.

Deleted Brand

Deleting a brand triggers a message to be sent to notify external systems. Brand name is sent as part of the delete message.

Table 2-11 Brand – Delete

Message Element	Included?	Notes
Brand name	Always	This field contains the brand name being deleted.

Error Handling

When the publication encounters a non-fatal error, messages continue to be processed. For the message where the error was encountered, a status of Hospital (H) is sent to the RIB and the status of the message in the queue is set to H. These types of errors occur when no changes in the database have been made and a process to try to re-publish the messages is available. In case the error is a fatal error, a status of Error (E) is sent to the RIB and the next message in the queue is not retrieved until the error is resolved.

Message XSD

Here are the filenames that correspond with each message type. Please consult the RIB documentation for each message type to get a detailed picture of the composition of each message.

Message Types	Message Type Description	XML Schema Definition (XSD)
brandcre	Brand Create Message	BrandDesc.xsd
brandupd	Brand Modify Message	BrandDesc.xsd
branddel	Brand Delete Message	BrandRef.xsd

Code and Diff Type Publication API

This section describes the seed data publication API.

Functional Area

Foundation Data

Business Overview

Merchandising defines and publishes code types, codes, and their descriptions, along with differentiator (diff) type information for data seeding in external systems, as they are usually fairly static and do not frequently change after initial implementation. However, changes and deletes are also managed in this integration. Subscribing to this information in an external system allows it to interpret information included in other Merchandising integrations, such as diffs associated with an item or statuses and other codes associated with purchase orders, transfers, and so on.

Code types and codes published are defined in Merchandising and are mostly used to display lists in the Merchandising UI like item types, shipping methods, supplier types, location types, and so on. Most code types and codes are part of the base implementation of Merchandising, but it is also possible to add or update codes and descriptions to suit your implementation. Code types and codes defined in Merchandising are also used in Sales Audit and Pricing. All code types and codes are published by this integration, regardless of whether or not they are flagged as Used in your implementation.

Diff Types are used to qualify the diff IDs included in other integration. Examples of diff types are size, color, flavor, and so on.

New Code Types

The creation of a new code type triggers the generation of a code header creation message. All code types defined in the merchandising is published to the external systems. The table below summarize the details included in this message.

Code Header

Table 2-12 Code Header Message Elements

Message Element	Included?	Notes
Code Type	Always	This field contains the code type which serves as a grouping mechanism for the codes.
Code Type Description	Always	This field contains the description of the code type.

Updated Code Types

Modifying the description of a code type will trigger the creation of a code header modification message. The update message will contain the details for all fields that changed to inform subscribing applications of the changes made in Merchandising. See the [New Code Types](#) section for details on the message.

Deleted Code Types

When a code type is removed, it will trigger a code header delete transaction message to an external system.

Code Header

Table 2-13 Code Header Message Elements

Message Element	Included?	Notes
Code Type	Always	This field contains the code type being deleted.

New Codes

The creation of a new code triggers the generation of a code detail creation message. All codes defined in the merchandising is published to the external systems. The table below summarize the details included in this message.

Code Detail

Table 2-14 New Codes Code Detail Message Elements

Message Element	Included?	Notes
Code Type	Always	This field contains a valid code type for the code.
Code	Always	This field contains the new code that has been added.
Code Description	Always	This field contains the description associated with the code.
Required Indicator	Always	This field indicates whether or not the code is required by Merchandising. Valid values are Yes (Y) or No (N).
Code Sequence	Always	This number is used to order the codes for display purposes.

Updated Codes

Modifying the description, required indicator, or code sequence of a code will trigger the creation of code detail modification message. The update message will contain the details for all fields that changed to inform subscribing applications of the changes made in Merchandising. See the [New Codes](#) section for details on the message.

Deleted Codes

When a code is removed, it will trigger a code detail delete transaction message to an external system.

**Note:**

Setting a code's **Used** flag to `No` in Merchandising does not trigger a delete message to be published.

Code Detail

Table 2-15 Delete Codes Code Detail Message Elements

Message Element	Required?	Notes
Code Type	Always	This field contains the code type for the code being deleted.
Code	Always	This field contains the code being deleted.

New Diff Types

The creation of a new diff type triggers the generation of a diff type create message. The tables below summarize the details of the message.

Diff Type

Table 2-16 New Diff Types Diff Type Message Elements

Message Element	Required?	Notes
Diff Type	Always	Contains the value used to uniquely identify the diff type being added or updated.
Diff Type Description	Always	Contains the diff type description.

Child Nodes

- Flex Attributes (optional)

Flex Attributes

Table 2-17 Flex Attributes

Message Element	Required?	Notes
Name	Always	Contains the attribute name configured for a diff type flex attribute.
Value	Optional	Contains the value of the attribute associated with the diff type for a character or number attribute.

Table 2-17 (Cont.) Flex Attributes

Message Element	Required?	Notes
Value Date	Optional	Contains the value of the attribute associated with the diff type, if the attribute is defined as a date type.

Updated Diff Types

Modifying diff type information will trigger the creation of a diff type modification message. The update message will contain the details for all fields that changed in the message in a format similar to that described above, to inform subscribing applications of the changes made in Merchandising.

Whenever a diff type is modified, all active CFAS defined for the diff type will be included in the message.



Note:

Only active CFAS will be included

Deleted Diff Types

When a diff type is removed, it will trigger a diff type delete transaction message to an external system. CFAS attributes are not included in the delete message, but a delete of the diff type implies that the flex attributes should also be deleted in the subscribing solution.

Diff Type

Table 2-18 Diff Type Message Elements

Message Element	Required?	Notes
Diff Type	Always	Contains the value used to uniquely identify the diff type being deleted.

Error Handling

When the publication encounters a non-fatal error, messages continue to be processed. For the particular message where the error was encountered, a status of Hospital (H) is sent to the RIB and the status of the message in the queue is set to H. These types of errors occur when no changes in the database have been made and a process to try to re-publish the messages is available. In case the error is a fatal error, a status of Error (E) is sent to the RIB and the next message in the queue is not retrieved until the error is resolved.

Message XSD

Below are the filenames that correspond with each message type. Please consult the Oracle Retail Integration Guide for each message type for the details on the composition of each message.

Table 2-19 Message XSD

Message Types	Message Type Description	XML Schema Definition (XSD)
CodeHdrCre	Code Header Create	CodeHdrDesc.xsd
CodeHdrMod	Code Header Modify	CodeHdrDesc.xsd
CodeHdrDel	Code Header Delete	CodeHdrRef.xsd
CodeDtlCre	Code Detail Create	CodeDtlDesc.xsd
CodeDtlMod	Code Detail Modify	CodeDtlDesc.xsd
CodeDtlDel	Code Detail Delete	CodeDtlRef.xsd
diffypecre	Diff Type Create	DiffTypeDesc.xsd
diffypemod	Diff Type Modify	DiffTypeDesc.xsd
diffypedel	Diff Type Delete	DiffTypeRef.xsd

Company Closed Publication API

This section describes the company closed publication API.

Functional Area

Foundation Data

Business Overview

Merchandising publishes details about new and updated company-wide closings for external systems that use this information to be informed of the updates, including company closed exceptions.

New Closings

Creating a company closing date triggers a message to be sent to notify external systems. The close date and description are sent as part of the create message.

Table 2-20 Company Closed - Create and Update

Message Element	Included?	Notes
Close Date	Always	This field contains the date that the company is closed.
Close Description	Optional	This field contains the description of the close, such as the reason for the company closure.

Updated Closings

Updating the description for a company closing date triggers a message to be sent to notify external systems. The close date and modified close description are sent as part of the message.

Deleted Closings

When a company closing date is deleted, this will also trigger a delete transaction to be sent to notify external systems that this company close date is no longer valid. The delete message will include only the close date.

Table 2-21 Company Closed - Delete

Message Element	Included?	Notes
Close Date	Always	This field contains the date that the company is closed.

New Closing Exceptions

Creating a company closing exception triggers a message to be sent to notify external systems. The close date, exception location, location type, and an indicator for whether or not the location is open for sales, receiving, and/or shipping are sent as part of the create message.

Table 2-22 Company Closed Exception - Create and Update

Message Element	Included?	Notes
Close Date	Always	This field contains the date on which an activity will not occur.
Location	Always	This field contains the unique location number.
Location Type	Always	This field contains the location type. Valid values are: <ul style="list-style-type: none"> S - Store W - Warehouse
Sales Indicator	Always	This field indicates whether or not the location is open for sales. A value of Yes (Y) indicates the location is open for sales. Valid values are Yes (Y) and No (N).
Receive Indicator	Always	This field indicates whether or not the location is open for receiving. A value of Yes (Y) indicates the location is open for receiving. Valid values are Yes (Y) and No (N).
Ship Indicator	Always	This field indicates whether or not the location is open for shipping. A value of Yes (Y) indicates the location is open for shipping. Valid values are Yes (Y) and No (N).

Updated Closing Exceptions

Updating a company closing exception triggers a message to be sent to notify external systems. The close date, exception location, location type, and an indicator for whether or not the location is open for sales, receiving, and/or shipping are sent as part of the message.

Deleted Closing Exceptions

When a company closing exception is deleted, this will also trigger a delete transaction to be sent to notify external systems that this company closed exception is no longer valid. The delete message will include only the close date and location.

Table 2-23 Company Closed Exception - Delete

Message Element	Included?	Notes
Location	Always	This field contains the unique location number.
Close Date	Always	This field contains the date on which an activity will not occur.

Error Handling

If any errors are encountered while publishing the message, a fatal error with status E (Error) is sent to RIB. This would bring down the RIB adapter thus preventing any further messages from being processed until this is resolved.

Message XSD

Below are the filenames that correspond with each message type. Please consult the Oracle Retail Integration Guide for each message type for the details on the composition of each message.

Message Types	Message Type Description	XML Schema Definition
CoClosedCre	Company Closing Create	CompanyClosedDesc.xsd
CoClosedExcCre	Company Closing Exception Create	CompanyClosedExcepDesc.xsd
CoClosedMod	Company Closing Modify	CompanyClosedDesc.xsd
CoClosedExcMod	Company Closing Exception Modify	CompanyClosedExcepDesc.xsd
CoClosedDel	Company Closing Delete	CompanyClosedRef.xsd
CoClosedExcDel	Company Closing Exception Delete	CompanyClosedExcepRef.xsd

Country Publication API

This section describes the countries publication API.

Functional Area

Foundation Data

Business Overview

Seed object publication to the RIB allows Merchandising to send country information as well as currency rates so that external systems will have all of the latest information regarding countries and currency rates.

Seed object publication consists of a message containing country and currency rate information from the tables COUNTRY and CURRENCY_RATES. One message will be synchronously created and placed in the message queue each time a COUNTRY and CURRENCY_RATES record is created, modified or deleted in Merchandising. When a COUNTRY or CURRENCY_RATES record is created or modified, the message will contain a full snapshot of the modified record. When a COUNTRY record is deleted, the message will contain a partial snapshot of the deleted record. Messages are retrieved from the message queue in the order they were created.

New Country

The data seeding of country information and creation of a new country triggers the generation of a country creation message. The country creation message publishes the country ID, along with the country's attributes.

Country

Table 2-24 Country

Message Element	Included?	Notes
Country ID	Always	This contains the unique country identifier that has been added or updated.
Country Description	Always	This contains the name of the country.

Updated Country

Modifying a country information as part of various business processes will trigger the creation of country modification message. The update message will contain the details for all fields that changed in the message to inform subscribing applications of the changes made in Merchandising.

Deleted Country

When a country is removed, it will trigger a country delete transaction message to an external system.

Table 2-25 Deleted Country

Message Elements	Required?	Notes
Country ID	Always	This contains the unique country identifier that has been deleted.

Country

Message Elements	Included?	Notes
Country ID	Always	This contains the unique country identifier that has been deleted.

Error Handling

When the publication encounters a non-fatal error, messages continue to be processed. For the particular message where the error was encountered, a status of Hospital (H) is sent to the RIB and the status of the message in the queue is set to H. These types of errors occur when no changes in the database have been made and a process to try to re-publish the messages is available. In case the error is a fatal error, a status of Error (E) is sent to the RIB and the next message in the queue is not retrieved until the error is resolved.

Message XSD

Here are the filenames that correspond with each message type. Please consult the RIB documentation for each message type in order to get a detailed picture of the composition of each message.

Message Types	Message Type Description	XML Schema Definition (XSD)
countrycre	Country Create Message	CountryDesc.xsd
countrymod	Country Modify Message	CountryDesc.xsd
countrydel	Country Delete Message	CountryRef.xsd

Customer Order Fulfillment Confirmation Publication API

This section describes the customer order fulfillment confirmation publication API.

Functional Area

Customer Orders

Business Overview

When Merchandising is integrated with an order management system (OMS) or Oracle Retail Order Broker (OROB), one of the supported deployment methods is interfacing customer order fulfillment requests into Merchandising through the Oracle Retail Integration Bus (RIB). When Merchandising processes the customer order fulfillment request from OROB, it will also publish a confirmation message back, indicating whether the order was created fully, partially, or not created. The confirmation message details are shown below.

Merchandising will only publish confirmation messages associated to customer orders that create a Purchase Order or Transfer, which is used for fulfillment from a supplier

or a warehouse, or if the fulfillment requires movement between two locations, such as from warehouse to store for customer pickup.

Table 2-26 Fulfill Order Confirmation Message

Message Element	Included?	Notes
Customer Order Number	Always	This is the customer order number from OROB or OMS that is being confirmed.
Fulfillment Order Number	Always	This is the fulfillment order number from OROB or OMS.
Confirmation Type	Always	Contains the confirmation type. Valid values are: <ul style="list-style-type: none"> • C - Order Completely Created • P - Order Partially Created • X - Order Could not be Created
Confirmation Number	Optional	This field contains the PO or Transfer number in Merchandising related to the fulfillment order line. This is populated if fulfillment status is P or C.
Confirmation Details	Optional	Child Node
Fulfillment Location ID	Optional	This field contains the fulfillment location for the order.
System Code	Always	Used by OROB to identify the system associated with this message. This is defaulted from the Merchandising system option.

Table 2-27 Fulfill Order Confirmation Detail Message

Message Element	Required?	Notes
Item	Always	This field indicates the item ordered by the customer.
Reference Item	Optional	This field indicates the reference item ordered by the customer. It is used only if a specific UPC is ordered.
Confirmed Quantity	Always	This field indicates the quantity of the item that can be sourced or fulfilled on the order.
Confirmed Quantity UOM	Always	This field indicates the unit of measure of the confirmation quantity.
Item Line Number	Optional	This field indicates the detail item line number on the order.

Error Handling

When the publication encounters a non-fatal error, messages continue to be processed. For the message where the error was encountered, a status of Hospital (H) is sent to the RIB and the status of the message in the queue is set to H. These types of errors occur when no changes in the database have been made and a process to try to re-publish the messages is available. In case the error is a fatal error, a status of Error (E) is sent to the RIB and the next message in the queue is not retrieved until the error is resolved.

Message XSD

Below are the filenames that correspond with each message type. Please consult the Oracle Retail Integration Guide for each message type for the details on the composition of each message.

Message Types	Message Type Description	XML Schema Definition (XSD)
fulfilordcfmcre	Customer Order Fulfillment Confirmation Create	FulfilOrdCfmDesc.xsd FulfilOrdCfmDtl.xsd

Delivery Slot Publication API

This section describes the delivery slot publication API.

Functional Area

Replenishment

Business Overview

This API publishes delivery slot details to external systems. This information is further subscribed by integrating systems like Oracle Retail Store Inventory Management (SIM). Within Merchandising, delivery slots are only used with the Store Order method of replenishment, allowing you the option to have multiple deliveries per day for the same store/day.

Create Delivery Slot

Delivery slot creation will result in publishing the ID, description, and a sequence to notify the integrating systems.

Update Delivery Slot

Modifying an existing delivery slot in Merchandising, will result in publishing a delivery slot modify message to notify the integrating systems. This includes the ID, description, and sequence value of delivery slot.

Table 2-28 Delivery Slot Create and Update

Message Element	Included?	Notes
Delivery Slot ID	Always	This field specifies a code indicating the delivery timeframe for stores that is being created or updated. For example, AM or PM.
Delivery Slot Description	Always	This field holds the delivery slot description. This could contain the time element for the delivery schedule (i.e., Afternoon Slot 3 PM).
Delivery Slot Sequence	Always	This column will specify the display sequence for the delivery slots.

Delete Delivery Slot

Deleting an existing delivery slot in Merchandising will result in publishing a delivery slot deletion message to notify the Integrating systems. Only the slot ID is included for deletes.

Table 2-29 Delivery Slot Delete

Message Element	Included?	Notes
Delivery Slot ID	Always	This field specifies a code indicating the delivery timeframe for stores that is being deleted.

Error Handling

If any errors are encountered while publishing the message, a fatal error with status E (Error) is sent to RIB. This would bring down the RIB adapter thus preventing any further messages from being processed until this is resolved.

Message XSD

Below are the filenames that correspond with each message type. Please consult the Oracle Retail Integration Guide for each message type for the details on the composition of each message.

Message Types	Message Type Description	XML Schema Definition (XSD)
Dlvysltcre	Delivery Slot Create Message	DeliverySlotDesc.xsd
Dlvysltmod	Delivery Slot Modify Message	DeliverySlotDesc.xsd
Dlvysltdel	Delivery Slot Delete Message	DeliverySlotRef.xsd

Differentiator Group Publication API

This section describes the differentiator groups publication API.

Functional Area

Items

Business Overview

Merchandising publishes differentiator groups and its details through the RIB to external systems. Diff groups are published when a new diff group is created, updates are made to existing diff group, or an existing diff group is deleted.

New Diff Group Header

Creating a new diff group header triggers a message to be sent to notify external systems. The full details are also always sent for the new diff group.

New Diff Group Detail

Creating a new diff group detail record for an existing diff group also triggers a message to be sent to notify external systems. The details that are sent for the detail creation are the existing diff group ID, diff ID, and the display sequence, which indicates the order that diffs should be displayed within the group.

Updated Diff Group Header

When an existing diff group header is updated, an update message is triggered to provide the details of the update. The update message, like create, will contain the full details in the message for all fields at the header level.

Updated Diff Group Detail

When an existing detail for a diff group is updated, an update message is triggered to provide the details of the update. The update message, like create, will contain the full details in the message for all fields.

Table 2-30 Diff Group Header Create and Update

Message Element	Required?	Notes
Diff Group ID	Always	This field contains the ID of the diff group being added or updated.
Diff Group Type	Always	This field contains the code for the types of diffs contained in this group, such as S for size.
Diff Group Description	Always	This field contains the description of the diff group.
Diff Group Type Description	Optional	This field contains the description of the diff type, such as Size or Color.

Table 2-31 Diff Group Detail Create and Update

Message Element	Required?	Notes
Diff Group ID	Always	This field contains the ID of the diff group being added or updated.
Diff ID	Always	This field contains the ID of the diff that is included in the group.
Display Sequence	Optional	This field contains the display sequence for the diff in the group.

Deleted Diff Group

When an existing diff group is deleted, this will also trigger a delete transaction to be sent to notify external systems that this diff group is no longer valid. The delete message will include only the diff group ID. All the child records for the diff group should also be deleted when processing a diff group delete in an external system.

Deleted Diff Group Detail

When the existing details of diff group are deleted, this will also trigger a delete transaction to be sent to notify external systems that these details are no longer part of the diff group. The delete message will include the diff group ID and the diff ID. The diff ID would be the detail record from the diff group that has been deleted.

Table 2-32 Diff Group Header

Message Element	Required?	Notes
Diff Group ID	Always	This field contains the ID of the diff group being deleted.

Table 2-33 Diff Group Detail

Message Element	Required?	Notes
Diff Group ID	Always	This field contains the ID of the diff group whose detail is being deleted.
Diff ID	Always	This field contains diff being deleted from the group.

Error Handling

If any errors are encountered while publishing the message, a fatal error with status E (Error) is sent to RIB. This would bring down the RIB adapter thus preventing any further messages from being processed until this is resolved

Message XSD

Below are the filenames that correspond with each message type. Please consult the Oracle Retail Integration Guide for each message type for the details on the composition of each message.

Message Types	Message Type Description	XML Schema Definition (XSD)
DiffGrpHdrCre	Diff Group Create	DiffGrpHdrDesc.xsd
DiffGrpDtlCre	Diff Group Detail Create	DiffGrpDtlDesc.xsd
DiffGrpHdrMod	Diff Group Modify	DiffGrpHdrDesc.xsd
DiffGrpDtlMod	Diff Group Detail Modify	DiffGrpDtlDesc.xsd
DiffGrpDel	Diff Group Delete	DiffGrpRef.xsd
DiffGrpDtlDel	Diff Group Detail Delete	DiffGrpDtlRef.xsd

Differentiator Publication API

This section describes the differentiators publication API.

Functional Area

Items

Business Overview

Merchandising publishes details about new and updated differentiator identifiers (diff IDs) in order that external systems that use this information can be informed of the updates. Updates are provided synchronously in a near-real time manner. When the external system receives information about an item that includes the new differentiator ID, that system understands what the differentiator ID refers to.

New Differentiator IDs

When a new differentiator ID is created in Merchandising, it triggers a message to be sent to notify external systems. The full details are sent for the new differentiator ID as part of the create message, the differentiator ID, differentiator type, differentiator description, industry code, industry sub code, and differentiator type description.

Updated Differentiator IDs

When an existing differentiator ID is updated, an update message is triggered to provide the details of the update. The update message, like create, will contain the full details in the message for all fields in the message.

Table 2-34 Differentiator Create and Update

Message Element	Required?	Notes
Differentiator ID	Always	This field contains a unique identifier of the differentiator.
Differentiator Type	Always	This field contains a code indicating the type of differentiator, for example C for Color.
Differentiator Description	Always	This field contains the description of the differentiator.
Industry Code	Optional	This field is used to hold the unique code used by industry standards to identify the differentiator.
Industry Sub Code	Optional	This field is used to hold a sub-grouping code used by industry standards to further identify the differentiator.
Differentiator Type Description	Optional	This field holds the description of the diff type (for example, Color, Size, and so on).

Deleted Differentiator IDs

When an existing differentiator ID is deleted, this will also trigger a delete transaction to be sent to notify external systems that this differentiator ID is no longer valid. The delete message will include only the ID of the differentiator being deleted.

Table 2-35 Differentiator Delete

Message Element	Required?	Notes
Differentiator ID	Always	This field contains a unique identifier of the differentiator.

Error Handling

When the publication encounters a non-fatal error, messages continue to be processed. For the message where the error was encountered, a status of Hospital (H) is sent to the RIB and the status of the message in the queue is set to H. In case the error is a fatal error, a status of Error (E) is sent to the RIB and the next message in the queue is not retrieved until the error is resolved.

Message XSD

Below are the filenames that correspond with each message type. Please consult the Oracle Retail Integration Guide for each message type for the details on the composition of each message.

Message Types	Message Type Description	XML Schema Definition (XSD)
diffcre	Diff Create	DiffDesc.xsd
diffmod	Diff Modify	DiffDesc.xsd
diffdel	Diff Delete	DiffRef.xsd

Item Location Publication API

This section describes the item location publication API.

Functional Area

Items

Business Overview

Merchandising defines and publishes item/location relationships. The details about item/location relation creation, modification, de-activation, and deletion are important for other systems for the smooth functioning of several business processes. Attributes included in this integration are pulled from both the Item Location and Item Location Traits tables in Merchandising.

To support multi-channel environments in Merchandising, a physical warehouse is divided into one or more virtual warehouses to better track and manage goods by channel. Stores and virtual warehouses are the locations that hold inventory and are associated to an item. Therefore, when creating, modifying, or deleting an item/location relationship where the location is a virtual warehouse, the physical warehouse information is also included in the message published. This is to cater to external systems that require the physical warehouse rather than the virtual warehouse.

In general, pricing updates are sent from Pricing to inform dependent solutions of a pending price change or clearance markdown. But, for solutions that do not subscribe to pricing events from Pricing, when the selling price for an item is changed, an item/location update will be published to inform other solutions of the updated price, including an update to the clearance flag.

New Item Location Relationship

The creation of a new item/location relationship triggers the generation of an item/location creation message. Please note though, that item/location records will not be published before the corresponding item record is published. Merchandising ensures that when an item creation message has not been published yet, the corresponding item/location creation message is not picked up for publication. The item/location relationship creation message publishes the item identifier, along with the location-specific attributes.

The message includes the following:

Item/Location Create and Update

Message Element	Included?	Notes
Item	Always	This is the item ID the item/location change is related to.
Item Level	Always	Indicates the level of the item within an item family. Valid values are 1, 2, or 3.
Tran Level	Always	This is the level at which inventory is tracked for this item's family. Valid values are 1, 2, or 3. Items having Item Level > Tran Level (indicating it is a below-transaction-level item) will not have its changes published.

Child Nodes

- Item/Location Physical Location Details

Item/Physical Location Create and Update

Message Element	Included?	Notes
Physical Location	Always	This is the Physical WH associated to the virtual warehouse location. If the location is a Store, this will contain the Store ID.
Location Type	Always	This is the type of location in the location field. Valid values are Store (S) or Warehouse (W).
Store Type	Optional	This is populated for store location types only. This will indicate whether a particular store is a franchise (F) or company store (C).
Stockholding Indicator	Optional	This is populated for store location types only. If the store is a non-stockholding store, message will not be subscribed to SIM/SIOCS.
Returnable Indicator	Always	This contains a value of Yes (Y) if the item can be returned to the location and No (N) if not.

Child Nodes

- Item/Location Virtual Location Details (required for item/locations update)
- Item/Location Virtual Location Replenishment Details (required for Replenishment item/location update)

Item/Virtual Location Create and Update

Message Element	Included?	Notes
Location	Always	Contains the location ID.
Location Type	Always	This is the type of location in the location field. Valid values are S (store) and W (warehouse).
Local Item Description	Optional	This contains the local description of the item.
Local Item Short Description	Optional	This contains the local short description of the item.
Status	Optional	This contains the status of the item at the location. Valid values are: <ul style="list-style-type: none"> • Active (A) - Item is valid and can be ordered and sold • Inactive (I) - Item is valid but cannot be ordered or sold • Discontinued (C) - Item is valid and sellable but no longer orderable • Delete (D) - Item is invalid and cannot be ordered or sold
Primary Supplier	Optional	This contains the primary supplier site for the item/location.
Primary Country	Optional	This contains the primary country of sourcing for the item/location.
Receive As Type	Optional	This determines whether the stock on hand for a pack component item or the buyer pack itself will be updated when a buyer pack is received at a warehouse. Valid values are Each (E) or Pack (P).
Taxable Indicator	Optional	This field determines if the item is taxable at the store. Valid values are Yes (Y) or No (N).
Source Method	Optional	This field determines the primary sourcing for this item/location. Valid values are Supplier (S) or Warehouse (W).
Source WH	Optional	This determines which warehouse is the sourcing location for this item/location. This value is required if the sourcing method is Warehouse.
Unit Retail	Optional	This defines the unit retail price in the standard unit of measure for the item/location combination. This field will be stored in the local currency.
Selling Unit Retail	Optional	This defines the unit retail price in the selling unit of measure for the item/location combination. This field will be stored in the local currency.
Selling UOM	Optional	This defines the selling unit of measure for an item's single-unit retail at the location.

Message Element	Included?	Notes
Store Price Indicator	Optional	This determines if an item at a particular store location can have the unit retail marked down by the store.
Purchase Type	Optional	This defines whether the item is owned, consignment stock, or a concession item at the location. Valid values are: <ul style="list-style-type: none"> • 0 - Owned • 1 - Consignment • 2 - Concession
UIN Type	Optional	This contains the type of unique identification number (UIN) used to identify the instances of the item at the location. Valid values are found in code type UINT.
UIN Label	Optional	This contains the label for the unique identification number (UIN) when displayed. Valid values are found in code type ULBL.
Capture Time	Optional	This determines when the unique identification number (UIN) should be captured for an item during transaction processing. Valid values are found in code type CPTM.
External UIN Indicator	Always	This indicates if a unique identification number (UIN) is being generated in the external system. Valid values are Yes (Y) or No (N).
Ranged Indicator	Optional	This determines if the location is ranged intentionally by a user for replenishment/selling or incidentally ranged by other processes. Valid values are Yes (Y) or No (N).
Item Parent	Optional	This uniquely identifies the item/group at the level above the item.
Item Grandparent	Optional	This uniquely identifies the item/group two levels above the item.
TI	Optional	This determines the number of shipping units (cases) that make up one tier of a pallet. Multiply TI x HI to get total number of cases for a pallet.
HI	Optional	This determines the number of tiers that make up a complete pallet (height). Multiply TI x HI to get total number of cases for a pallet.
Store Order Multiple	Optional	This contains the multiple in which the item needs to be shipped from a warehouse to the location. Valid values are found in ORML.
Daily Wastage Percentage	Optional	This defines the average percentage lost from inventory on a daily basis due to natural wastage.
Ticket Price Units	Optional	This defines the units in the ticket price in terms of the Price UOM for ticketing.
Ticket Price	Optional	This defines the retail to be used on the ticket in terms of the Price UOM for ticketing.
Ticket UOM	Optional	Unit of measure used on the ticket for this item.

Message Element	Included?	Notes
Primary Variant	Optional	This field is used to address sales of PLUs (that is, above transaction level items) when inventory is tracked at a lower level (that is, UPC). This field will only contain a value for items one level higher than the transaction level. Valid choices will be any transaction level item that is a child of this item.
Primary Cost Pack	Optional	This contains a simple pack item containing the item in the item column for this record.
Inbound Handling Days	Optional	This indicates the number of inbound handling days for an item at a warehouse type location.
Regular Unit Retail	Optional	This defines the unit retail in the standard unit of measure for the item/location when not on clearance. This field is stored in the local currency.
Multi-units	Optional	This defines the quantity to be purchased in order to get the multi-unit retail in the multi-selling unit of measure for the item/location.
Multi-unit Retail	Optional	This defines the multi-unit retail in the multi-selling unit of measure for the item/location. This field is stored in the local currency.
Multi-selling UOM	Optional	This defines the selling unit of measure for the item/location if a multi-unit retail has been defined.
Clearing Indicator	Optional	This determines if item is on clearance at the store. Valid values are Yes (Y) or No (N).
Costing Location	Optional	This contains the costing location for the franchise store. This field may contain a store or a warehouse.
Costing Location Type	Optional	This defines the type of location is in the costing location field. Valid values are Store (S) or Warehouse (W).
Currency Code	Always	This defines the currency code under which the location operates.
Launch Date	Optional	This holds the date when the item is initially sold at the location.
Quantity Key Options	Optional	This field determines whether the quantity key on a POS can be used for this item at the location. Valid values are found in code type RPO.
Manual Price Entry	Optional	This field determines whether the price for the item/location can be entered manually on a POS. Valid values are found in code type RPO.
Deposit Code	Optional	This determines whether a deposit is associated with this item at the location. Valid values are found in code DEPO.
Food Stamp Indicator	Optional	This determines whether the item is approved for food stamps at the location. Valid values are Yes (Y) or No (N).
WIC Indicator	Optional	This determines whether the item is approved for WIC at the location. Valid values are Yes (Y) or No (N).

Message Element	Included?	Notes
Proportional Tare Percentage	Optional	This is the proportion of the total weight of a unit of an item that is the packaging. For example, if the tare item is bulk candy, this is the proportional of the total weight of one piece of candy that is the candy wrapper.
Fixed Tare Value	Optional	This is the tare of the packaging. For example, if the tare item is bulk candy, this is weight of the bag and twist tie.
Fixed Tare UOM	Optional	This contains the unit of measure value associated with the tare value.
Reward Eligible Indicator	Optional	This determines whether the item is legally valid for various types of bonus point/award programs at the location. Valid values are Yes (Y) or No (N).
National Brand Comparison Item	Optional	This contains the nationally branded item to which it will be compared to. Will contain a valid item ID.
Return Policy	Optional	This determines the return policy for the item at the location. Valid values are found in code type RETP.
Stop Sale Indicator	Optional	This defines if the sale of the item should be stopped immediately at the location (that is, in case of recall, or something similar). Valid values are Yes (Y) or No (N).
Elect Marketing Clubs	Optional	This contains the code that represents the marketing clubs to which the item belongs at the location. Valid values are found in code type MKTC.
Report Code	Optional	This determines to which reports the location should run. Valid values are found in code type REPC.
Required Shelf Life on Selection	Optional	This contains the required shelf life for an item on selection in days.
Required Shelf Life on Receipt	Optional	This contains the required shelf life for an item on receipt in days.
Store Re-orderable Indicator	Optional	This determines whether the store may re-order the item. Valid values are Yes (Y) or No (N).
Rack Size	Optional	This determines the rack size that should be used for the item.
Full Pallet Item	Optional	This determines whether a store must reorder an item in full pallets only. Valid values are Yes (Y) or No (N).
In-Store Market Basket	Optional	This contains the in-store market basket code for the item/location combination. Valid values are found in code type STMB.
Storage Location	Optional	This contains the current storage location or bin number for the item at the location.
Alternate Storage location	Optional	This contains the preferred alternate storage location or bin number for the item at the location.
Refundable Indicator	Optional	This determines if the item is refundable at the location or not. Valid values are Yes (Y) or No (N).
Back Order Indicator	Optional	This determines if the item can be back-ordered to the location. Valid values are Yes (Y) or No (N).

Message Element	Included?	Notes
Promotable Indicator	Optional	This determines whether the retailer is allowed to specify if the item is promotable or not. Valid values are Yes (Y) or No (N).
Unit Cost	Optional	This contains the current unit cost of the item based on the primary supplier/country for the location in local currency.
Pickup Lead Time	Optional	This defines the time it takes to get the item from the supplier to the location.
Cost UOM	Optional	This is used to allow costs to be managed in a different UOM than the standard UOM.
Calculation Basis	Optional	This determines if the cost for the consignment/ concession item will be managed either based on cost per unit or as a percentage of retail. Valid values are: <ul style="list-style-type: none"> • C - Cost per Unit • P - Purchase Rate
Purchase Rate	Optional	This contains the percentage of the retail price which will determine the cost paid to the supplier for a consignment or concession item, if the calculation basis for the item/location is Purchase Rate.
RFID Indicator	Optional	This allows the retailer to specify if the item should be RFID tagged or not.

Child Nodes

- Flex Attributes (optional)

Modified Item Location Relationship

Modifying an item/location attribute as part of various business processes will trigger the creation of an item/location modification message. The update message will contain the details for all fields that changed in the message to inform subscribing applications of the changes made in Merchandising.

Deleted Item Location Relationship

When an item/location relationship is removed from an item (not just moved to Deleted status), it will trigger an item/location delete transaction message to an external system. The delete message contains the item/location to be deleted.

Item/Location Delete

Message Element	Included?	Notes
Item	Always	This is the item ID where a location relationship is being deleted.
Item/Physical Location		Child Node

Item/Physical Location Delete

Message Element	Included?	Notes
Physical Location	Always	This is the Physical WH associated to a virtual warehouse location being deleted. If the location is a Store, this will contain the Store ID.
Location Type	Always	This is the type of location in the location field. Valid values are Store (S) or Warehouse (W).
Store Type	Optional	This is populated for store location types only. This will indicate whether a particular store is a franchise (F) or company store (C).
Stockholding Indicator	Optional	This is populated for store location types only. If the store is a non-stockholding store, message will not be subscribed to SIM/SIOCS.
Item/Virtual Location		Child Node

Item/Virtual Location Delete

Message Element	Included?	Notes
Location	Always	Contains the location ID being deleted for the item.

Item Replenishment Attributes

The creation, modification or deletion of item replenishment attributes for a location will trigger the creation of an item/location replenishment modification message. This will publish a message to subscribing systems to inform them of the replenishment attributes updates for the item on a particular location.

Item/Location Virtual Location Replenishment Details

Message Element	Included?	Notes
Location	Always	This is the location ID where the item/location change is related to. This will contain the Virtual Warehouse if the location type is Warehouse or the Store ID if the location type is a Store.
Location Type	Always	This is the type of location in the location field. Valid values are Store (S) or Warehouse (W).
Primary Replenishment Supplier	Always	The supplier from which the specified location will source the replenishment demand for the specified item location.

Message Element	Included?	Notes
Replenishment Method	Always	The code for the algorithm that will be used to calculate the recommended order quantity for the item location. Valid values are: <ul style="list-style-type: none"> • C - Constant • M - Min/Max • F - Floating point • T - Time Supply • D - Dynamic • SO - Store Orders
Reject Store Order Indicator	Optional	Contains an indicator that determines whether uploaded store orders can be rejected. If the indicator is No (N), then store orders for all need dates are valid. If Yes (Y), store orders with need dates on or after the next delivery date are valid.
Next Delivery Date	Optional	The next delivery date calculated for the next review cycle.
Multiple Runs Per Day Indicator	Always	Indicates if an item can be replenished multiple times per day at the location.

Flex Attributes

If any custom flex attributes (CFAS) for the item/location has been added or modified, it will trigger an item/location modify message. All of the entity's active flex attributes from all attribute groups are published as key-value pairs based on the group set view. This CFAS object is embedded in the outbound Item/Location message.

Flex Attributes

Message Elements	Included?	Notes
Name	Always	The flex attribute defined by the business
Value	Optional	The value of the flex attribute defined by the business
Value Date	Optional	The date value of the flex attribute if the flex attribute is defined as a date.

Error Handling

When the publication encounters a non-fatal error, messages continue to be processed. For the particular message where the error was encountered, a status of Hospital (H) is sent to the RIB and the status of the message in the queue is set to H. These types of errors occur when no changes in the database have been made and a process to try to re-publish the messages is available. In case the error is a fatal error, a status of Error (E) is sent to the RIB and the next message in the queue is not retrieved until the error is resolved.

Message XSD

Below are the filenames that correspond with each message type. Please consult the Oracle Retail Integration Guide for each message type for the details on the composition of each message.

Message Types	Message Type Description	XML Schema Definition (XSD)
ItemLocCre	Item Location Create Message	ItemLocDesc.xsd
ItemLocMod	Item Location Modify Message	ItemLocDesc.xsd
ItemLocDel	Item Location Delete Message	ItemLocRef.xsd
ItemLocReplMod	Item Location Replenishment Modify Message	ItemLocDesc.xsd

Item Publication API

This section describes the item publication API.

Functional Area

Items

Business Overview

Merchandising publishes messages about items to the Oracle Retail Integration Bus (RIB). In situations where a retailer creates a new item in Merchandising, the message that ultimately is published to the RIB contains a hierarchical structure of the item itself along with all components that are associated with that item. Items and item components make up what is called the Items message family.

After the item creation message has been published to the RIB for use by external applications, any modifications to the basic item or its components cause the publication of individual messages specific to that component. Deletion of an item and component records has similar effects on the message modification process, with the exception that the delete message holds only the key(s) for the record.

When publishing an item header modification, reference item creation, reference item modification, and reference item deletion, a second full replacement message from Merchandising will be published when the publish full objects indicator is Yes. This message payload will contain item header (with custom flex attributes), item BOM and item UPC information. Based on the message type, RIB will route the full replacement message to appropriate applications.

Deposit Items

A deposit item is a product that has a portion which is returnable to the supplier and sold to the customer, with a deposit taken for the returnable portion. Because the contents portion of the item and the container portion of the item must be managed in separate financial accounts (as the container item would be posted to a liabilities account) with different attributes, the retailer must set up two separate items. All

returns of used deposit items (the returned item) are managed as a separate product, to track these products separately and as a generic item not linked to the actual deposit item (for example, bottles being washed and having no label).

The retailer can never put a container item on a transfer. The container item is added to returns to vendors (RTVs) automatically when the retailer adds the associated content item.

Deposit item attributes in Merchandising enable contents, container and crate items to be distinguished from one another. Additionally, it is possible to link a contents item to a container item for the purposes of inventory management.

In addition to contents and container items, many deposit items are delivered in plastic crates, which are also given to the customer on a deposit basis. These crates are sold to a customer as an additional separate product. Individual crates are not linked with contents or container items. Crates are specified in the system with a deposit item attribute.

From a receiving perspective, only the content item can be received. The receipt of a PO shows the container item, but the receipt of a transfer does not. Similar to RTV functionality, online purchase order functionality automatically adds the container. The system automatically replicates all transactions for the container item in the stock ledger. In sum, for POs and RTVs, the container item is included; for transfers, no replication occurs.

Catch-Weight Items

Retailers can order and manage products for the following types of catch-weight item:

- Type 1: Purchase in fixed weight simple packs: sell by variable weight (for example, bananas).
- Type 2: Purchase in variable weight simple packs: sell by variable weight (for example, ham on the bone sold on a delicatessen counter).
- Type 3: Purchase in fixed weight simple packs containing a fixed number of eaches: sell by variable weight eaches (for example, pre-packaged cheese).
- Type 4: Purchase in variable weight simple packs containing a fixed number of eaches: sell by variable weight eaches (for example, pre-packaged sirloin steak).

 **Note:**

Oracle Retail suggests that catch-weight item cases be managed through the standard simple pack functionality.

For catch-weight items to be managed in Merchandising, the following item attributes are available:

- Cost UOM: All items in Merchandising will be able to have the cost of the item managed in a separate unit of measure (UOM) from the standard UOM. Where this is in a different UOM class from the standard UOM, case dimensions must be set up.
- Catch-weight item pack details: Tolerance values and average case weights are stored for catch-weight item cases to allow the retailer to report on the sizes of cases received from suppliers.
- Maximum catch-weight tolerance threshold.
- Minimum catch-weight tolerance threshold.

Retailers can set up the following properties for a catch-weight item:

- Order type
- Sale type

Retailers can also specify the following, at the item-supplier-country level:

- Cost unit of measure (CUOM).

Receiving and inventory movement impact on catch-weight items

Inventory transaction messages include purchase order receiving, stock order receiving, returns to vendor, direct store delivery receiving, inventory adjustments and bill of lading. These messages include attributes that represent, for catch-weight items, the actual weight of goods involved in a transaction. These attributes are weight and weight UOM.

When Merchandising subscribes to inventory transaction messages containing such weight data, the transaction weight will be used for two purposes:

- To update weighted average cost (WAC) using the weight rather than the number of units and to update the average weight value of simple packs.

Note:

The WAC calculation does not apply to return to vendors (RTVs).

Item Transformation

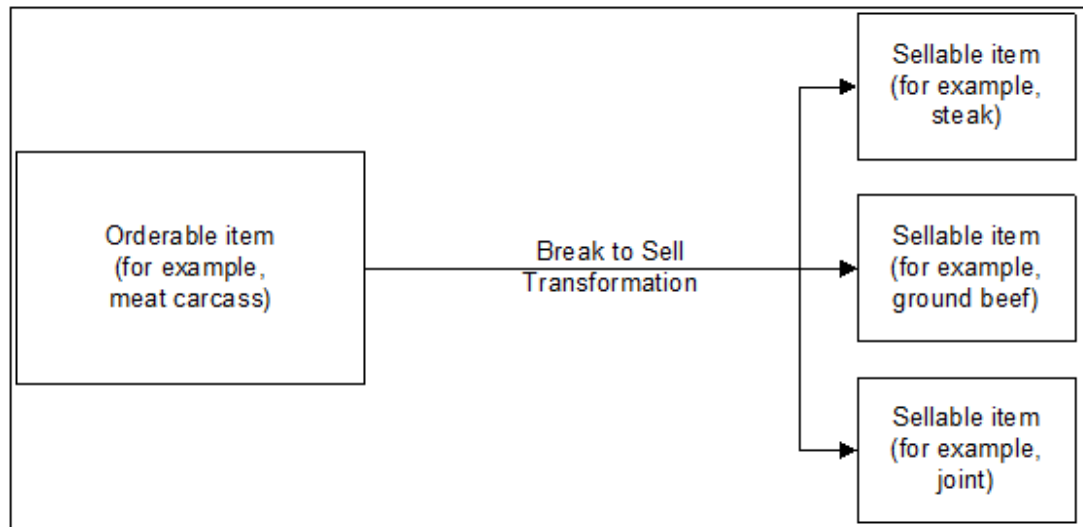
Item transformation allows retailers to manage items where the actual transformation of a product cannot be adequately recorded due to in-store processes.

With product transformation, new 'transform' items are set up as either sellable only or orderable only.

- **Sellable only items:** A sellable only item has no inventory in the system, so inventory records cannot be viewed from the item maintenance screens. Sellable only items do not hold any supplier links and therefore have no cost prices associated with them.
- **Orderable only items:** Orderable only items hold inventory, but are not sellable at the POS system. Therefore, no information is sent to the POS system for these items, and no unit retail prices by zone are held for these items.

To hold the relationship between the orderable items and the sellable items, Merchandising stores the transformation details. These details are used to process sales and inventory transactions for the items.

The following diagram shows how item transformation works:

Figure 2-2 Item Transformation

Item and Item Component Descriptions

The item message family is a logical grouping for all item data published to the RIB. The components of item messages and their base tables in Merchandising are:

- Item from the ITEM_MASTER table
- Item-supplier from ITEM_SUPPLIER
- Item-supplier-country from ITEM_SUPP_COUNTRY
- Item-supplier-country-dimension from ITEM_SUPP_COUNTRY_DIM (DIM is the each, inner, pallet, and case dimension for the item, as specified)
- Item-image from ITEM_IMAGE
- Item-UDA identifier-UDA value from UDA_ITEM_LOV (UDA is a user-defined attribute and LOV is list of values)
- Item-UDA identifier from UDA_ITEM_DATE (for the item and UDA date)
- Item-UDA identifier from UDA_ITEM_FF (for UDA, free-format data beyond the values for LOV and date)
- Item-pack components (Bill of Material [BOM]) from PACKITEM_BREAKOUT
- Item UPC reference from ITEM_MASTER.ITEM_NUMBER_TYPE (values held as code type 'UPCT' on code_head and code_detail tables)
- Item ticket from ITEM_TICKET
- Item relationship details from RELATED_ITEM_HEAD
- Related Items details from RELATED_ITEM_DETAIL
- VAT item details from VAT_ITEM (only when default tax type is not GTS)

New Item Message Processes

The creation of a new item in Merchandising begins with an item in a worksheet status on the ITEM_MASTER table. At the time an item is created, other relationships are being defined as well, including the item, supplier, and country relationships, user-defined attributes (UDAs), related items and others. These item relationship processes in effect become components of a new item message published to the RIB. This section describes the item creation message process and includes the basic item message itself along with the other component relationship messages that become part of the larger item message.

Basic Item Message

As described in the preceding section, item messages can originate in a number of Merchandising tables. Creating, modifying or deleting an item triggers a message to be sent through the RIB to notify external systems. Additions, modifications and deletions to item family records for existing approved items are published in the order that they are placed on the queue.

New Item Message Publication

The publication of a new item and its components to the RIB is done using a hierarchical message. All the components that comprise the creation of an item, the item/supplier for example, remain in the queue until the item approval modification message has been published. Any modifications or deletions that occur between item creation in “W” (worksheet) status and “A” (Approved) status are applied to the “create” messages or deleted from the queue as required.

For example, if an item UDA is added before item approval and then later deleted before item approval, the item UDA “create” message would be deleted from the queue before publishing the item. If an item/supplier record is updated for a new item before the item is approved, the “create” message for that item/supplier is updated with the new data before the item is published. When the “modify” message that contains the “A” (Approved) status is the next record on the queue, the procedure formats a hierarchical message that contains the item header information and all the child detail records to pass to RIB.

Subordinate Data and XML Tags

While a new item is being created, item components are also being created. Described earlier in this overview, these component item messages pertain to the item-supplier, item-supplier-country, UDAs, and so on.

For example, when a new item-supplier is created this triggers a message to be created. When the item is approved, this item-supplier message will be added to the item creation message.

Table 2-36 Item Header

Message Element	Included?	Notes
Item	Always	Unique alphanumeric value that identifies the item.

Table 2-36 (Cont.) Item Header

Message Element	Included?	Notes
Item Number Type	Always	Indicates the type of the number, for example UPC-12. The value values for this column are maintained in code type UPCT.
Format Id	Optional	This field will hold the format ID that corresponds to the item's variable UPC. This value is only used for items with variable UPCs.
Prefix	Optional	This column holds the prefix for variable weight UPCs. The prefix determines the format of the eventual UPC and will be used to decode variable weight UPCs that are uploaded from the POS. It is the clients responsibility to download this value to their POS system.
Item Parent	Optional	Alphanumeric value that uniquely identifies the item/group at the level above the item. This value must exist as an item in another row on the item_master table.
Item Grandparent	Optional	Alphanumeric value that uniquely identifies the item/group two levels above the item. This value must exist as both an item and an item parent in another row on the item_master table.
Pack Indicator	Always	Indicates if the item is a pack. A pack item is a collection of items that may be either ordered or sold as a unit. Packs require details (i.e. component items and qtys, etc.) that other items do not.
Item Level	Always	Number indicating which of the three levels the item resides. The item level determines if the item stands alone or if it is part of a family of related items. The item level also determines how the item may be used throughout the system.
Transaction Level	Always	Number indicating which of the three levels transactions occur for the item's group. The transaction level is the level at which the item's inventory is tracked in the system. The transaction level item will be counted, transferred, shipped, etc. The transaction level may be at the current item or up to 2 levels above or below the current item. Only one level of the hierarchy of an item family may contain transaction level items.
Diff 1	Optional	Diff_group or diff_id that differentiates the current item from its item_parent. For an item that is a parent, this field may be either a group (i.e. Men's pant sizes) or a value (6 oz). For an item that is not a parent, this field may contain a value (34X34, Red, etc.). Valid values are found on the diff_group and diff_id tables.
Diff 1 Type	Optional	This field will hold a value of all possible differential types. (for example, 'S' - size, 'C' - color, 'F' - flavor, 'E' - scent, 'P' - pattern). Valid values are stored on the code_detail table with the code_type 'DIFF' and on the diff_type table.

Table 2-36 (Cont.) Item Header

Message Element	Included?	Notes
Diff 2	Optional	Diff_group or diff_id that differentiates the current item from its item_parent. For an item that is a parent, this field may be either a group (i.e. Men's pant sizes) or a value (6 oz). For an item that is not a parent, this field may contain a value (34X34, Red, etc.). Valid values are found on the diff_group and diff_id tables.
Diff 2 Type	Optional	This field will hold a value of all possible differential types. (for example, 'S' - size, 'C' - color, 'F' - flavor, 'E' - scent, 'P' - pattern). Valid values are stored on the code_detail table with the code_type 'DIFF' and on the diff_type table.
Diff 3	Optional	Diff_group or diff_id that differentiates the current item from its item_parent. For an item that is a parent, this field may be either a group (i.e. Men's pant sizes) or a value (6 oz). For an item that is not a parent, this field may contain a value (34X34, Red, etc.). Valid values are found on the diff_group and diff_id tables.
Diff 3 Type	Optional	This field will hold a value of all possible differential types. (for example, 'S' - size, 'C' - color, 'F' - flavor, 'E' - scent, 'P' - pattern). Valid values are stored on the code_detail table with the code_type 'DIFF' and on the diff_type table.
Diff 4	Optional	Diff_group or diff_id that differentiates the current item from its item_parent. For an item that is a parent, this field may be either a group (i.e. Men's pant sizes) or a value (6 oz). For an item that is not a parent, this field may contain a value (34X34, Red, etc.). Valid values are found on the diff_group and diff_id tables.
Diff 4 Type	Optional	This field will hold a value of all possible differential types. (for example, 'S' - size, 'C' - color, 'F' - flavor, 'E' - scent, 'P' - pattern). Valid values are stored on the code_detail table with the code_type 'DIFF' and on the diff_type table.
Department	Always	Number identifying the department to which the item is attached. The item's department will be the same as that of its parent (and, by transitivity, to that of its grandparent). Valid values for this field are located on the deps table.
Department Name	Always	Contains the name of the department.
Class	Always	Number identifying the class to which the item is attached. The item's class will be the same as that of its parent (and, by transitivity, to that of its grandparent). Valid values for this field are located on the class table.
Class Name	Always	Contains the name of the class.

Table 2-36 (Cont.) Item Header

Message Element	Included?	Notes
Subclass	Always	Number identifying the subclass to which the item is attached. The item's subclass will be the same as that of its parent (and, by transitivity, to that of its grandparent). Valid values for this field are located on the subclass table.
Subclass Name	Always	Contains the name of the subclass.
Status	Always	Status of the item. Valid values are: W = Worksheet: item setup in progress, cannot be used in system; S = Submitted: item setup complete and awaiting approval, cannot be use in system; A = Approved: item is approved and can now be used throughout the system. An item is not published until it is Approved.
Item Description	Always	Long description of the item. This description is used throughout the system to help online users identify the item. For items that have parents, this description will default to the parent's description plus any differentiators. For items without parents this description will default to null.
Short Description	Optional	Shortened description of the item. This description is the default for downloading to the POS. For items that have parents, this description will default to the parent's short description. For items without parents, this description will default to null.
Description Up	Optional	All uppercase description of the item (same as upper(item_desc)). This field is not displayed in the on-line item maintenance screen, but is used in item list of value search processing throughout the system.
Primary Reference Item Indicator	Optional	Indicates if the sub-transaction level item is designated as the primary sub-transaction level item. For transaction level items and above the value in this field will be 'N'o.
Cost Zone Group Id	Optional	Cost zone group associated with the item. This field is only required when elc_ind (landed cost indicator) is set to 'Y' on the system_options table.
Standard Unit of Measure	Always	Unit of measure in which stock of the item is tracked at a corporate level.
Unit of Measure Conv Factor	Optional	Conversion factor between an "Each" and the standard_uom when the standard_uom is not in the quantity class (e.g. if standard_uom = lb and 1 lb = 10 eaches, this factor will be 10). This factor will be used to convert sales and stock data when an item is retailed in eaches but does not have eaches as its standard unit of measure.
Package Size	Optional	Holds the size of the product printed on any packaging (for example, 24 ounces).
Package Unit of Measure	Optional	Holds the unit of measure associated with the package size.
Merchandise Indicator	Always	Indicates if the item is a merchandise item (Y, N).

Table 2-36 (Cont.) Item Header

Message Element	Included?	Notes
Store Order Multiple	Always	Merchandise shipped from the warehouses to the stores must be specified in this unit type. Valid values are: C = Cases; I = Inner; E = Eaches.
Forecast Indicator	Always	Indicates if this item will be interfaced to an external forecasting system (Y, N).
Mfg Rec Retail	Optional	Manufacturer's recommended retail price for the item. Used for informational purposes only. This field is stored in the primary currency.
Retail Label Type	Optional	This field indicates any special label type associated with an item (i.e. pre-priced or cents off). This field is used for reporting purposes only. Values for this field are defined by the 'RTLTL' code on code detail.
Retail Label Type Description	Optional	Contains the retail label type description corresponding to the retail_label_type value in item_master.
Retail Label Value	Optional	This field represents the value associated with the retail label type.
Handling Temp	Optional	Holds the temperature information associated with the item. Valid values for this field are in the code type 'HTMP' on the code_head and code_detail tables.
Handling Temp Description	Optional	Contains the Handling temperature description corresponding to the handling_temp value in item_master.
Handling Sensitivity Description	Optional	Contains the Handling sensitivity description corresponding to the handling_sensitivity value in item_master.
Catch Weight Indicator	Always	Indicates whether the item should be weighed when it arrives at a location. Valid values for this field are 'Y' and 'N'.
Waste Type	Optional	Identifies the wastage type as either sales or spoilage wastage. Sales wastage occurs during processes that make an item saleable (i.e. fat is trimmed off at customer request). Spoilage wastage occurs during the products shelf life (i.e. evaporation causes the product to weigh less after a period of time). Valid values are: SP - spoilage; and SL - Sales.
Waste Type Description	Optional	Contains the Waste Type description corresponding to the Waste_Type value in item_master.
Waste Percent	Optional	Average percent of wastage for the item over its shelf life. Used in inflating the retail price for wastage items.
Default Waste Percent	Optional	Default daily wastage percent for spoilage type wastage items. This value will default to all item locations and represents the average amount of wastage that occurs on a daily basis.

Table 2-36 (Cont.) Item Header

Message Element	Included?	Notes
Const Dimension Indicator	Optional	Indicates that the dimensions of the product are always the same, regardless of the supplier. If this field is Y, the dimensions for all suppliers will be defaulted to the primary supplier/primary country dimensions. Editing of dimensions for the item will only be allowed for the primary supplier/primary country.
Simple Pack Indicator	Always	Indicates if pack item is a simple pack or not. This field will only be available if the item is a pack item. A simple pack is an item whose components are all the same item (for example, a six pack of cola). Valid values are: Y = Yes, this item is a simple pack; N = No if this is not a simple pack.
Contains Inner Indicator	Always	Indicates if pack item contains inner packs. Vendor packs will never contain inner packs and this field will be defaulted to 'N'. This field will only be available if the item is a pack item. Valid values are: Y = Yes, this pack contains inner packs; N = No, this pack does not contain inner packs.
Sellable Indicator	Always	Indicates if pack item may be sold as a unit. If it is 'Y' then the pack will have its own unique unit retail. If it is 'N' then the pack's unit retail is the sum of each individual item's total retail within the pack. This field will only be available if the item is a pack item. Value values are: Y = Yes, this pack may be sold as a unit; N = No, this pack may not be sold as a unit.
Orderable Indicator	Always	Indicates if pack item is orderable. If it is 'Y' then the suppliers of the pack must supply all components in the pack. If it is 'N' then the components may have different suppliers. This field will only be available if the item is a pack item. Valid values are: Y = Yes, this pack may be ordered, N = No, this pack may not be ordered.
Pack Type	Optional	Indicates if pack item is a vendor pack or a buyer pack. A vendor pack is a pack that the vendor or supplier recognizes and sells to the retailer. If the pack item is a vendor pack, communication with the supplier will use the vendor pack number. A buyer pack is a pack that a buyer created for internal ease of use. If the pack item is a buyer pack communication with the supplier will explode the pack out to its component items. This field will only be used if the item is a pack item. If the pack item is not orderable this field must be null. Valid values are: V = Vendor; B = Buyer.

Table 2-36 (Cont.) Item Header

Message Element	Included?	Notes
Order As Type	Optional	Indicates if pack item is receivable at the component level or at the pack level (for a buyer pack only). This field is required if pack item is an orderable buyer pack. This field must be NULL if the pack is sellable only or a vendor pack. This field will only be available if the item is a pack item. Valid Values are: E = Eaches (component level) P = Pack (buyer pack only).
Comments	Optional	Holds any comments associated with the item.
Unit Retail	Optional	This field holds the unit retail in the standard unit of measure for the item/zone combination. This field is stored in the local currency.
Item Service Level	Optional	Holds a value that restricts the type of shipment methods that RCOM can select for an item.
Gift Wrap Indicator	Always	This field will contain a value of 'Y' if the item is eligible to be gift wrapped.
Ship Alone Indicator	Always	This field will contain a value of 'Y' if the item should be shipped to the customer is a separate package versus being grouped together in a box.
Vendor Style	No	Not used by RMS.
Standard Unit Weight	No	Not used by RMS.
Single Price Flag	No	Not used by RMS.
Pre Ticket Flag	No	Not used by RMS.
Planned Residual	No	Not used by RMS.
Sortable	No	Not used by RMS.
Item Master UDA1	No	Not used by RMS.
Item Master UDA2	No	Not used by RMS.
Item Master UDA3	No	Not used by RMS.
Item Master UDA4	No	Not used by RMS.
Item Master UDA5	No	Not used by RMS.
Item Master UDA6	No	Not used by RMS.
Item Master UDA7	No	Not used by RMS.
Item Master UDA8	No	Not used by RMS.
Item Master UDA9	No	Not used by RMS.
Item Master UDA10	No	Not used by RMS.
Item Master UDA11	No	Not used by RMS.
Item Master UDA12	No	Not used by RMS.
Item Master UDA13	No	Not used by RMS.
Item Master UDA14	No	Not used by RMS.
Item Master UDA15	No	Not used by RMS.
Ship Alone	No	Not used by RMS.
Slottable	No	Not used by RMS.

Table 2-36 (Cont.) Item Header

Message Element	Included?	Notes
Freight Class	No	Not used by RMS.
New Item	No	Not used by RMS.
Brand	No	Not used by RMS.
Break Case Ups	No	Not used by RMS.
Rigid	No	Not used by RMS.
Fragile	No	Not used by RMS.
Container Type	No	Not used by RMS.
Conveyable Flag	No	Not used by RMS.
Hazard Matl Code	No	Not used by RMS.
Velocity	No	Not used by RMS.
High Value Indicator	No	Not used by RMS.
Ticket Type	No	Not used by RMS.
Color	No	Not used by RMS.
Size1	No	Not used by RMS.
Fit	No	Not used by RMS.
Shade	No	Not used by RMS.
Single Contain Bulk	No	Not used by RMS.
Unit Pick System Code	No	Not used by RMS.
Roundable Flag	No	Not used by RMS.
Perishable Indicator	No	Not used by RMS.
Division	No	Not used by RMS.
Vendor Number	No	Not used by RMS.
Knitting WIP Code	Optional	WIP code used to designate the Kitting process used in item_master
Unit Ticket Quantity	No	Not used by RMS.
Item Length	No	Not used by RMS
Item Width	No	Not used by RMS
Item Height	No	Not used by RMS
Item Weight	No	Not used by RMS
Item Cube	No	Not used by RMS
Standard Unit Quantity	No	Not used by RMS
Expiration Days	No	Not used by RMS
Putaway By Volume	No	Not used by RMS
Putaway Plan	No	Not used by RMS
Item Type	No	Not used by RMS
Sorter Group	No	Not used by RMS
SKU Optimization	No	Not used by RMS
Ext Source System	No	Not used by RMS

Table 2-36 (Cont.) Item Header

Message Element	Included?	Notes
Item Xform Indicator	Optional	This indicator will show that an item is associated with an item transformation. The item will be either the sellable item or orderable item in the transformation process.
Order Type	Optional	This determines how catch weight items are ordered. Valid values are: F - fixed weight V - Variable weight Valid values are held on the CODE_DETAIL table with a code type = ORDT
Sale Type	Optional	This indicates the method of how catch weight items are sold in store locations. Valid values are: V - variable weight each L - Loose weight. Valid values are held on the CODE_DETAIL table with a code type = STPE
Deposit Item Type	Optional	This is the deposit item component type. A NULL value in this field indicates that this item is not part of a deposit item relationship. The possible values are - E - Contents A - Container Z - Crate T - Returned Item (Empty bottle) P - Complex pack (with deposit items) The Returned item is flagged only to enable these items to be mapped to a separate GL account if required.
Container Item	Optional	This holds the container item number for a contents item. This field is only populated and required if the DEPOSIT_ITEM_TYPE = E.
Deposit in Price Per UOM	Optional	This field indicates if the deposit amount is included in the price per UOM calculation for a contents item ticket. This value is only required if the DEPOSIT_ITEM_TYPE = E. Valid values are: I - Include deposit amount E - Exclude deposit amount
Transport Indicator	No	Not used by RMS.
Notional Pack Indicator	Always	Indicates to the Store Inventory System if a sellable simple pack needs to be broken down to its component item.
SOH Inquiry at Pack Indicator	Always	Indicates to the Store Inventory System if a notional simple pack item's inventory should be displayed in packs.
Purchase Type	Optional	Indicates if an item is a normal, consignment or concession item - 'N' for normal, 'S' for consignment, 'C' for concession.
Product Classification	Optional	This contains item combinability codes (with code type 'PCLA') which provide a way to define which items can be combined (packed or boxed) together.
Item Aggregate Indicator	Always	Indicator to aggregate inventory and sales for the item. Currently, this indicator is used by allocation and MFP to aggregate inventory for items. For staple items this indicator should be N.
Diff 1 Aggregate Indicator	Always	Indicator for the corresponding diff. Indicator to aggregate inventory and sales for an item at Parent/Diff level (for example, Style/Color or Style/Size).

Table 2-36 (Cont.) Item Header

Message Element	Included?	Notes
Diff 2 Aggregate Indicator	Always	Indicator for the corresponding diff. Indicator to aggregate inventory and sales for an item at Parent/Diff level (for example, Style/Color or Style/Size).
Diff 3 Aggregate Indicator	Always	Indicator for the corresponding diff. Indicator to aggregate inventory and sales for an item at Parent/Diff level (for example, Style/Color or Style/Size).
Diff 4 Aggregate Indicator	Always	Indicator for the corresponding diff. Indicator to aggregate inventory and sales for an item at Parent/Diff level (for example, Style/Color or Style/Size).
Item Description Secondary	Optional	Secondary descriptions of the item. This field can only be populated when <code>system_options.secondary_desc_ind = Y</code> .
Original Retail	Optional	The field indicated the original retail price of the item per unit. This field is stored in the primary currency.
Original Retail Currency Code	Optional	The field indicates the currency code of the original retail price.
Mfg Retail Currency Code	Optional	The field indicates the currency code of the Manufacturer's retail price.
Catch Weight Type	Optional	This column will hold catch weight type for a simple pack catch weight item. The value is based on the component items <code>order_type</code> and <code>sale_type</code> : 2 - <code>order_type = Variable Weight, sale_type = Loose Weight</code> 4 - <code>order_type = Variable Weight, sale_type = Variable Weight Each</code> The column will be set only at the time of Item approval.
Catch Weight UOM	Optional	Indicates the UOM for Catchweight Items.
Brand Name	Optional	This field contains the brand associated to an item.
Unique Class	Optional	This field uniquely identify the class which the item belongs to. Class is not a unique ID in the merchandise hierarchy. The combination of Dept/Class is unique, but requires use of a composite key. The composite key is generally used in user interfaces. The unique ID can be used in back end processing or in systems that cannot have a composite key in the merchandise hierarchy.
Unique Subclass	Optional	This field uniquely identify the subclass which the item belongs to. Subclass is not a unique ID in the merchandise hierarchy. The combination of Dept/Class/Subclass is unique, but requires use of a composite key. The composite key is generally used in user interfaces. The unique ID can be used in back end processing or in systems that cannot have a composite key in the merchandise hierarchy.
Diff 1 Level	Optional	This field will contain either ID or GROUP, based on whether the <code>diff_1</code> is a group diff or a diff id.

Table 2-36 (Cont.) Item Header

Message Element	Included?	Notes
Diff 1 Description	Optional	This field will hold the description of differentiator.
Diff 2 Level	Optional	This field will contain either ID or GROUP, based on whether the diff_1 is a group diff or a diff id.
Diff 2 Description	Optional	This field will hold the description of differentiator.
Diff 3 Level	Optional	This field will contain either ID or GROUP, based on whether the diff_1 is a group diff or a diff id.
Diff 3 Description	Optional	This field will hold the description of differentiator.
Diff 4 Level	Optional	This field will contain either ID or GROUP, based on whether the diff_1 is a group diff or a diff id.
Diff 4 Description	Optional	This field will hold the description of differentiator.
Prod Classification Description	Optional	This field will hold the description of the product classification.
Primary Image URL	Optional	This field will hold the primary item image URL. It is only included for the ItemCre and ItemFulRep message types. Updates to the primary image for an item are published in the Item Image node.

Table 2-37 Item Supplier

Message Element	Included?	Notes
Item	Always	Unique identifier for the item.
Supplier	Always	Unique identifier for the supplier.
Primary Supplier Indicator	Always	This field indicates whether this supplier is the primary supplier for the item. An item can have one and only one primary supplier.
VPN	Optional	This field contains the Vendor Product Number for the Item/Supplier.
Supp Label	Optional	This field contains the supplier label. It will only be populated if item is a parent item.
Consignment Rate	Optional	This field contains the consignment rate for this item for the supplier.
Supplier Diff 1	Optional	This field contains the first supplier differentiator and/or description. This field may only contain a value for items with an item_parent.
Supplier Diff 2	Optional	This field contains the second supplier differentiator and/or description. This field may only contain a value for items with an item_parent.
Supplier Diff 3	Optional	This field contains the second supplier differentiator and/or description. This field may only contain a value for items with an item_parent.
Supplier Diff 4	Optional	This field contains the second supplier differentiator and/or description. This field may only contain a value for items with an item_parent.
Pallet Name	Always	Code referencing the name used to refer to the pallet.

Table 2-37 (Cont.) Item Supplier

Message Element	Included?	Notes
Case Name	Always	Code referencing the name used to refer to the case.
Inner Name	Always	Code referencing the name used to refer to the inner.
Supplier Discontinue Date	Optional	Date when the supplier discontinues an item.
Direct Ship Indicator	Always	A value of 'Y' (Yes) in this field indicates that any item associated with this supplier is eligible for a direct shipment from the supplier to the customer.
Primary Case Size	Optional	This field indicates the primary case size for the item supplier when an orderable item is configured for informal case types.

Table 2-38 Item Supplier Country

Message Element	Included?	Notes
Item	Always	Unique identifier for the item.
Supplier	Always	Unique identifier for the supplier.
Origin Country Id	Always	The country where the item was manufactured or significantly altered.
Primary Supplier Indicator	Always	This field indicates whether this supplier is the primary supplier for the item. Each item can have one and only one primary supplier.
Primary Country Indicator	Always	This field indicates whether this country is the primary country for the item/supplier. Each item/supplier combination can have one and only one primary country.
Unit Cost	Always	This field contains the current corporate unit cost for the item from the supplier /origin country. This field is stored in the Supplier's currency.
Lead Time	Optional	This field contains the number of days that will elapse between the date an order is written and the delivery to the store or warehouse from the supplier.
Pick up Lead Time	Optional	Contains the time that will be taken to get the item from the supplier to the initial receiving location.
Supplier Pack Size	Always	This field contains the quantity that orders must be placed in multiples of for the supplier for the item.
Inner Pack Size	Always	This field contains the break pack size for this item from the supplier.
Round Level	Always	This column is used to determine how order quantities will be rounded to case, layer and pallet.
Minimum Order Quantity	Optional	This field contains the minimum quantity that can be ordered at once from the supplier for the item.
Max Order Quantity	Optional	This field contains the maximum quantity that can be ordered at once from the supplier for the item.

Table 2-38 (Cont.) Item Supplier Country

Message Element	Included?	Notes
Packing Method	Optional	This field indicates whether the packing method for the item in the container is flat or hanging.
Default Unit of Purchase	Optional	Contains the default unit of purchase for the item/ supplier/country. Valid values include 'C' (Case) and 'P' (Pallet).
Ti	Always	Number of shipping units (cases) that make up one tier of the pallet. Multiply TI x HI to get total number of units (cases) in a Pallet.
Hi	Always	Number of tiers that make up a complete pallet. Multiply TI x HI to get total number of units (cases) in a Pallet.
Cost Unit of Measure	Always	A cost UOM is held to allow costs to be managed in a separate UOM to the standard UOM.
Tolerance Type	Optional	The unit of tolerance for catch weight simple packs. Valid values are: 'A' - actual 'P' - percent.
Max Tolerance	Optional	The maximum tolerance value for the catch weight simple pack.
Min Tolerance	Optional	The minimum tolerance value for a catch weight simple pack.
Round to Inner Pct	Always	This column will hold the Inner Rounding Threshold value. During rounding, this value is used to determine whether to round partial Inner quantities up or down. If the Inner-fraction in question is less than the Threshold proportion, it is rounded down; if not, it is rounded up.
Round to Case Pct	Always	This column will hold the Case Rounding Threshold value. During rounding, this value is used to determine whether to round partial Case quantities up or down. If the Case-fraction in question is less than the Threshold proportion, it is rounded down; if not, it is rounded up.
Round to Layer Pct	Always	This column will hold the Layer Rounding Threshold value. During rounding, this value is used to determine whether to round partial Layer quantities up or down. If the Layer-fraction in question is less than the Threshold proportion, it is rounded down; if not, it is rounded up.
Round to Pallet Pct	Always	This column will hold the Pallet Rounding Threshold value. During rounding, this value is used to determine whether to round partial Pallet quantities up or down. If the Pallet -fraction in question is less than the Threshold proportion, it is rounded down; if not, it is rounded up.
Supplier Hierarchy Type 1	Optional	This field identifies partner type of supplier hierarchy level 1. This field will always have the partner type S1 which indicates manufacturer.
Supplier Hier Level 1	Optional	Highest level of supplier hierarchy which is there to indicate a partner, such as a manufacturer, in the supply chain that gives rebates to the retailer.
Supplier Hierarchy Type 2	Optional	This field identifies partner type of supplier hierarchy level 2. This field will always have the partner type S2 which indicates distributor.

Table 2-38 (Cont.) Item Supplier Country

Message Element	Included?	Notes
Supplier Hier Level 2	Optional	Second highest level of supplier hierarchy which is there to indicate a partner, such as a distributor, in the supply chain that gives rebates to the retailer.
Supplier Hierarchy Type 3	Optional	This field identifies partner type of supplier hierarchy level 3. This field will always have the partner type S3 which indicates wholesaler.
Supplier Hier Level 3	Optional	Third highest level of supplier hierarchy which is there to indicate a partner, such as a wholesaler, in the supply chain that gives rebates to the retailer.
Negotiated Item Cost	Optional	This will hold the supplier negotiated item cost for the primary delivery country of the item. Once a location is associated with the item, the primary locations negotiated item cost will be stored in this field.
Extended Base Cost	Optional	This will hold the extended base cost for the primary delivery country of the item. Once a location is associated with the item, the primary locations extended base cost will be stored in this field. Extended base cost is the cost inclusive of all the taxes that affect the WAC. In case of GTAX, Extended Base Cost = Base Cost + Non-recoverable taxes. In case of VAT, Extended Base Cost = Base Cost.
Inclusive Cost	Optional	This will hold the inclusive cost for the primary delivery country of the item. Once a location is associated with the item, the primary locations inclusive cost will be stored in this field. This cost will have both the recoverable and non-recoverable taxes included. In case of GTAX, Inclusive Cost = Base Cost + Non-recoverable taxes + Recoverable Taxes. In case of VAT, Inclusive Cost = Base Cost + VAT.
Base Cost	Optional	This field will hold the tax exclusive cost of the item.
Purchase Type	Optional	This contains a code to indicate whether the item is normal merchandise (i.e. owned by the retailer), consignment stock or a concession item. Valid values are 0 - Owned, 1 - Consignment, 2 - Concession.
Calculation Basis	Optional	This indicates if the cost for the consignment/ concession item will be managed either based on cost per unit or as a percentage of retail. Valid values are C - Cost per Unit, P - Purchase Rate.
Purchase Rate	Optional	This column contains the percentage of the retail price which will determine the cost paid to the supplier for a consignment or concession item.

Table 2-39 Item Supplier Country Dimension

Message Element	Included?	Notes
Item	Always	Unique identifier for the item.
Supplier	Always	Unique identifier for the supplier.

Table 2-39 (Cont.) Item Supplier Country Dimension

Message Element	Included?	Notes
Origin Country Id	Always	The country where the item was manufactured or significantly altered.
DIM Object	Always	Specific object whose dimensions are specified in this record (e.g. case, pallet, each). Valid values for this field are in the code type DIMO on the codes tables.
Presentation Method	Optional	Describes the packaging (if any) being taken into consideration in the specified dimensions. Valid values for this field are in the code type PCKT on the codes tables.
Method Description	Optional	The description for the Presentation Method.
Length	Optional	Length of dim_object measured in units specified in terms of the LWH Unit of Measure.
Width	Optional	Width of dim_object measured in units specified in terms of the LWH Unit of Measure.
Height	Optional	Height of dim_object measured in units specified in terms of the LWH Unit of Measure.
LWH Unit of Measure	Optional	Unit of measurement for length, width, and height.
Weight	Optional	Weight of dim_object measured in units specified in terms of the Weight Unit of Measure.
Net Weight	Optional	Net weight of the dim_object (weight without packaging) measured in units specified in terms of the Weight Unit of Measure.
Weight Unit of Measure	Optional	Unit of measurement for Weight and Net Weight.
Liquid Volume Unit of Measure	Optional	Unit of measurement for liquid volume. Liquid volumes are only convertible to other liquid volumes.
Stat Cube	Optional	Statistical value of the dim_object's dimensions to be used for loading purposes.
Tare Weight	Optional	Amount of weight to be subtracted for packaging materials. Used to calculate the true net weight of the dim_object.
Tare Type	Optional	Indicates whether the tare weight for the dim_object is wet or dry.

Table 2-40 Item User Defined Attribute List of Value

Message Element	Included?	Notes
Item	Always	Unique identifier for the item.
UDA ID	Always	This field contains a number identifying the user-defined attribute.
UDA Value	Always	Contains the value for the user-defined attribute.
UDA Description	Always	This field contains a description of the User-Defined Attribute.
UDA Value Description	Always	This field contains a description of the UDA value.

Table 2-41 Item User Defined Attribute FF

Message Element	Included?	Notes
Item	Always	Unique identifier for the item.
UDA Id	Always	This field contains a number identifying the user-defined attribute.
UDA Text	Always	This field contains the text value of the user-defined attribute for the item.

Table 2-42 Item User Defined Date

Message Element	Included?	Notes
Item	Always	Unique identifier for the item.
UDA Id	Always	This field contains a number identifying the user-defined attribute.
UDA Date	Always	This field contains the effective date for the UDA ID.

Table 2-43 Item Image

Message Element	Included?	Notes
Item	Always	Unique identifier for the item.
Image Name	Always	Name of the image.
Image Addr	Always	Path where the image file is stored.
Image Type	Always	This field contains the type of the image of the item. Valid values are defined as member of IITD code type.
Primary Indicator	Always	Field to specify the display sequence order of images associated to the item per priority.
Image Description	Always	This field contains the description associated with the image of the item.

Table 2-44 Item UPC

Message Element	Included?	Notes
Item	Always	Unique identifier for the item.
Item Parent	Always	Unique identifier for the item/group at the level above the item.
Primary Ref Item Indicator	Always	Indicates if the sub-transaction level item is designated as the primary sub-transaction level item. For transaction level items and above, the value in this field will be 'N'o.
Format ID	Optional	This field will hold the format id that corresponds to the item's variable UPC. This value is only used for items with variable UPCs.

Table 2-44 (Cont.) Item UPC

Message Element	Included?	Notes
Prefix	Optional	This field holds the prefix for variable weight UPCs. The prefix determines the format of the eventual UPC and will be used to decode variable weight UPCs that are uploaded from the POS.
Item Number Type	Always	Code specifying what type the item is. Valid values for this field are in the code type UPCT on the code_head and code_detail tables.

Item BOM information will be published only for an item creation message and full replacement message.

Table 2-45 Item BOM

Message Element	Included?	Notes
Pack No	No	Unique identifier for the pack item.
Item	No	Unique identifier for the component item.
Pack Quantity	No	Contains the quantity of each item in the pack item.

Table 2-46 Item Ticket Header

Message Element	Included?	Notes
Item	Always	Unique identifier for the pack item.
Ticket Type Id	Always	This field uniquely identifies the ticket type which is associated with the item.
Ticket Type Description	Always	This field contains a description of the ticket or label type.
Shelf Edge Label Indicator	Always	Indicates if the ticket type is a shelf edge label.
PO Print Type	Optional	This field indicates when the ticket type for the given item should be printed by default, upon the approval or receipt of the purchase order. Valid values are: A and R.
Print on PC Indicator	Always	This field indicates whether this type of ticket should be printed for this item when a permanent price change goes into effect. Valid values in this field are: Y - Yes N - No.
Ticket Over Pct	Optional	This field contains a percent which indicates the number of tickets which should be printed for a given event. If the event causing tickets to be printed is approving or receiving a purchase order, then this field indicates the quantity of tickets which should be printed greater than the quantity of the purchase order. If the event causing the tickets to be printed is a permanent price change, this field would indicate the quantity of tickets which should be printed greater than the stock on hand.

Table 2-47 Item Ticket Detail

Message Element	Included?	Notes
Ticket Item Id	Optional	This field contains a character string which uniquely identifies an attribute which will appear on a ticket or label such as retail price or price per unit of measure.
Ticket Item Id Description	Optional	This field contains the description of the ticket_item_id.
UDA Id	Optional	This field contains a number which uniquely defines a user-defined attribute which is to be printed on this ticket type.

Table 2-48 Item Supplier Country Manufacturer

Message Element	Included?	Notes
Item	Always	Unique identifier for the item.
Supplier	Always	Unique identifier for the supplier.
Manufacturer Country Id	Always	Country ID of manufacturer.
Primary Manufacturer Country Indicator	Always	Primary country of manufacturer indicator.

Table 2-49 Related Item Header

Message Element	Included?	Notes
Item	Always	The main item against which the relationship is defined.
Relationship Id	Always	A system generated numeric field that uniquely identifies the relationship record in related_item_head table.
Relationship Name	Always	This is the user entered name of the relationship.
Relationship Type	Always	Describes the type of relationship. Valid values are in the codes table under code_type 'IREL'.
Mandatory Indicator	Always	Indicates whether the relationship is mandatory or not. Valid values: 'Y' (yes), 'N' (no).

Table 2-50 Related Item Detail

Message Element	Included?	Notes
Related Item	Always	Item for which related items are defined. Valid transaction level/sellable item. Multiple items can be defined for a relationship.
Priority	Optional	Relationship priority. Applicable only for relationship type Substitute.
Effective Date	Optional	The related item can be used on transactions starting this date.
End Date	Optional	The related item can be used on transactions until this date. A value of null means that it is always effective.

VAT Item BOM information will be published only when default tax type is not GTS.

Table 2-51 VAT Item

Message Element	Included?	Notes
Item	Always	Unique identifier for the item.
Active Date	Always	This field holds the active date for record on future cost.
Vat Type	Always	This field indicates the type of vat either Retail or Cost or Both.
Vat Code	Always	This field indicates a code to uniquely identify a VAT rate.
Vat Rate	Always	This column contains vat rate associated with a given VAT code.
Reverse Vat Indicator	Always	This field indicates if the item is subject to reverse charge VAT at the vat region. Valid values are Y and N.
Vat Region	Always	Contains the number of the Value Added Tax region a location belongs to.

Table 2-52 Item Supplier Unit of Measure

Message Element	Included?	Notes
Item	Always	Unique identifier for the item.
Supplier	Always	Unique identifier for the supplier.
UOM	Always	The unit of measure that the item is being measured in.
Value	Always	This field will store the equivalent value of the Item/Suppliers shipping carton in the associated unit of measure.

Modify and Delete Messages

Updates and deletions of item data can be included in a larger item creation message. If not part of a larger hierarchical message, they are published individually as a flat, non-hierarchical message. Update and delete messages are much smaller than the large hierarchy in a newly created item message.

Modify Messages

If an existing item is modified, this will trigger an update message to provide details of the update through the RIB. The message will include any updated levels in the item hierarchy

Delete Messages

Delete messages are published in the same way that modify messages are. For example, if an item-supplier-country relationship is deleted from Merchandising, the dependent item-supplier-country dimension records are also deleted.

Table 2-53 Item

Message Element	Included?	Notes
Item	Always	Unique identifier for the item.

Table 2-54 Item Supplier

Message Element	Included?	Notes
Item	Always	Unique identifier for the item.
Supplier	Always	Unique identifier for the supplier.

Table 2-55 Item Supplier Country

Message Element	Included?	Notes
Item	Always	Unique identifier for the item.
Supplier	Always	Unique identifier for the supplier.
Country Id	Always	The country where the item was manufactured or significantly altered.

Table 2-56 Item Supplier Country Dimension

Message Element	Included?	Notes
Item	Always	Unique identifier for the item.
Supplier	Always	Unique identifier for the supplier.
Country Id	Always	The country where the item was manufactured or significantly altered.
DIM Object	Always	Dim_object description fetched from the code_detail table.

Table 2-57 Item UPC

Message Element	Included?	Notes
Item	Always	Unique identifier for the item.

Table 2-58 Item User Defined Attribute FF

Message Element	Included?	Notes
Item	Always	Unique identifier for the item.
UDA Id	Always	This field contains a number identifying the user-defined attribute.
UDA Text	Always	This field contains the text value of the user-defined attribute for the item.

Table 2-59 Item User Defined Attribute Date

Message Element	Included?	Notes
Item	Always	Unique identifier for the item.
UDA Id	Always	This field contains a number identifying the user-defined attribute.
UDA Date	Always	This field contains the effective date for the UDA ID.

Table 2-60 Item User Defined Attribute List of Value

Message Element	Included?	Notes
Item	Always	Unique identifier for the item.
UDA Id	Always	This field contains the unique identified of the Attribute Group being added or delete to the item list.
UDA Value	Always	Contains the value of the Attribute Group.

Table 2-61 Item Image

Message Element	Included?	Notes
Item	Always	Unique identifier for the item.
Image Name	Always	Name of the image.

Table 2-62 Item Ticket

Message Element	Included?	Notes
Item	Always	Unique identifier for the item.
Ticket Type Id	Always	This field uniquely identifies the ticket type which is associated with the item.

Table 2-63 Item Supplier Country Manufacturer

Message Element	Included?	Notes
Item	Always	Unique identifier for the item.
Supplier	Always	This field uniquely identifies the ticket type which is associated with the item.
Country Id	Always	The country where the item was manufactured or significantly altered.

Table 2-64 Related Item

Message Element	Included?	Notes
Item	Always	Unique identifier for the item.

Table 2-64 (Cont.) Related Item

Message Element	Included?	Notes
Relationship Id	Always	A system generated numeric field that uniquely identifies the relationship record in related_item_head.
Related Item	Always	Item for which related items are defined. Valid transaction level/sellable item. Multiple items can be defined for a relationship.

Table 2-65 VAT Item

Message Element	Included?	Notes
Item	Always	Unique identifier for the item.
Active Date	Always	This field holds the active date for record on future cost.
Vat Type	Always	This field indicates the type of vat either Retail or Cost or Both.
Vat Region	Always	Contains the number of the Value Added Tax region a location belongs to.

Table 2-66 Item Supplier Unit of Measure

Message Element	Included?	Notes
Item	Always	Unique identifier for the item.
Supplier	Always	Unique identifier for the supplier.
UOM	Always	The unit of measure that the item is being measured in.

Flex Attributes

If any custom flex attributes (CFAS) for the item have been added or modified, it will trigger an item header modify message. All the entity's active flex attributes from all attribute groups are published as key-value pairs based on the group set view. This CFAS object is embedded in the outbound Item message.

If any CFAS for the item-supplier has been added or modified, it will trigger an item supplier modify message. All the entity's active flex attributes from all attribute groups are published as key-value pairs based on the group set view. This CFAS object is embedded in the outbound item supplier message.

Similarly, if any CFAS for the item-supplier-country has been added or modified, it will trigger an item supplier modify message. All the entity's active flex attributes from all attribute groups are published as key-value pairs based on the group set view. This CFAS object is embedded in the outbound item supplier country message.

Flex attributes can only be added to or updated; they cannot be deleted.

Table 2-67 Flex Attributes

Message Element	Required?	Notes
Name	Always	Holds the attribute name.
Value	Optional	Holds the value of the attribute for number and character type attributes
Value Date	Optional	Holds the date for date type attributes.

Full Message Updates

In cases where the integrating system is unable to receive only the changes, another option is provided that can resend the item details whenever there is a change, in addition to the deltas. This will be published when the system option Publish Full Objects (PUB_FULL_OBJECTS_IND) is set to Deltas and Full (Y). This message will contain the header level information, its flex attributes, pack and reference item information. This is used for Oracle WMS Cloud integration.

Error Handling

When the publication encounters a non-fatal error, messages continue to be processed. For the message where the error was encountered, a status of Hospital (H) is sent to the RIB and the status of the message in the queue is set to H. In case the error is a fatal error, a status of Error (E) is sent to the RIB and the next message in the queue is not retrieved until the error is resolved.

Message XSD

Here are the filenames that correspond with each message type. Please consult RIB documentation for each message type to get a detailed picture of the composition of each message.

Message Types	Message Type Description	XML Schema Definition (XSD)
itemcre	Item Create Message	ItemDesc.xsd
itemmod	Item Modify Message	ItemDesc.xsd
itemdel	Item Delete Message	ItemRef.xsd
itemsupcre	Item Supplier Create Message	ItemSupDesc.xsd
itemsupmod	Item Supplier Modify Message	ItemSupDesc.xsd
itemsupdel	Item Supplier Delete Message	ItemSupRef.xsd
itemsupctycre	Item Supplier Country Create Message	ItemSupCtyDesc.xsd
itemsupctymod	Item Supplier Country Modify Message	ItemSupCtyDesc.xsd
itemsupctydel	Item Supplier Country Delete Message	ItemSupCtyRef.xsd
iscdimcre	Item Supplier Country Dimension Create Message	ISCDimDesc.xsd

Message Types	Message Type Description	XML Schema Definition (XSD)
iscdimmod	Item Supplier Country Dimension Modify Message	ISCDimDesc.xsd
iscdimdel	Item Supplier Country Dimension Delete Message	ISCDimRef.xsd
itemupccre	Item UPC Create Message	ItemUPCDesc.xsd
itemupcmo	Item UPC Modify Message	ItemUPCDesc.xsd
itemupcdel	Item UPC Delete Message	ItemUPCRef.xsd
itemudaffcre	Item UDA Free Form Text Create Message	ItemUDAFFDesc.xsd
itemudaffmo	Item UDA Free Form Text Modify Message	ItemUDAFFDesc.xsd
itemudaffdel	Item UDA Free Form Text Delete Message	ItemUDAFFRef.xsd
itemudalovcre	Item UDA LOV Create Message	ItemUDALOVDesc.xsd
itemudalovmo	Item UDA LOV Modify Message	ItemUDALOVDesc.xsd
itemudalovdel	Item UDA LOV Delete Message	ItemUDALOVRef.xsd
itemudadatecre	Item UDA Date Create Message	ItemUDADateDesc.xsd
itemudadatemo	Item UDA Date Modify Message	ItemUDADateDesc.xsd
itemudadatedel	Item UDA Date Delete Message	ItemUDADateRef.xsd
itemimagecre	Item Image Create Message	ItemImageDesc.xsd
itemimagemo	Item Image Modify Message	ItemImageDesc.xsd
itemimagedel	Item Image Delete Message	ItemImageRef.xsd
relitemheadcre	Item Relationship Create Message	RelatedItemDesc.xsd
relitemheadmo	Item Relationship Modify Message	RelatedItemDesc.xsd
relitemheaddel	Item Relationship Delete Message	RelatedItemRef.xsd
relitemdetcre	Related Item Create Message	RelatedItemDesc.xsd
relitemdetmo	Related Item Modify Message	RelatedItemDesc.xsd
relitemdetdel	Related Item Delete Message	RelatedItemRef.xsd
itemfulrep	Item Full Replacement Message	ItemDesc.xsd

Merchandise Hierarchy Publication API

This section describes the merchandise hierarchy publication API.

Functional Area

Merchandise Hierarchy

Business Overview

This API publishes information regarding all the levels of the merchandise hierarchy to the RIB such that all the downstream applications may subscribe to it and have merchandise hierarchy information in sync with Merchandising.

New Division

When a new division is created in Merchandising, it triggers a message to be sent to notify external systems. The full details are sent for the new division as part of the create message, the division ID, division name, buyer, merchandiser, and total market amount.

Updated Division

When an existing division is updated, an update message is triggered to provide the details of the update. The update message, like create, will contain the full details in the message for all fields.

The format used when creating or modifying a division is shown below.

Table 2-68 Division Create and Update

Message Element	Included?	Notes
Division	Always	Contains the number which uniquely identifies the division of the company.
Division Name	Always	Contains the name which, along with the division number, identifies the division of the company.
Buyer	Optional	Contains the number of the buyer associated with the division.
Merchandiser	Optional	Contains the number of the merchandiser associated with the division.
Total Market Amount	Optional	Contains the total market amount that is expected for the division. The field can be used for analytics or reporting.

Deleted Division

When an existing division is deleted, this will also trigger a delete transaction which will be sent to notify external systems that this division is no longer valid. The delete message will include only the ID of the division being deleted.

Table 2-69 Division Delete

Message Element	Included?	Notes
Division	Always	Contains the number which uniquely identifies the division of the company.

New Group

When a new group is created in Merchandising, it triggers a message notifying external systems. The full details are sent for the new group as part of the create message, the group ID, group name, buyer, merchandiser, and division.

Updated Group

When an existing group is updated, an update message is triggered to provide the details of the update. The update message, like create, will contain the full details in the message for all fields.

Table 2-70 Group Create and Update

Message Element	Included?	Notes
Group Number	Always	Contains the number which uniquely identifies the group.
Group Name	Always	Contains the description which, along with the group number, identifies the group.
Buyer	Optional	Contains the number of the buyer associated with the group.
Merchandiser	Optional	Contains the number of the merchandiser associated with the group.
Division	Always	Contains the number of the division of which the group is a member.

Deleted Group

When an existing group is deleted, this will also trigger a delete transaction message to notify external systems that this group is no longer valid. The delete message will include only the group ID being deleted.

Table 2-71 Group Delete

Message Element	Included?	Notes
Group Number	Always	Contains the number which uniquely identifies the group.

New Department

When a new department is created in Merchandising, it triggers a message notifying external systems. The full details are sent for the new department as part of the create message, the department ID, department name, buyer, purchase type, total market amount, merchandiser, group, budgeted markup, profit calculation type, markup calculation type, OTB calculation type, budgeted intake percentage, and department-level VAT inclusive indicator. The custom flex attributes, if applicable, are also sent as a part of the message.

Updated Department

When an existing department is updated, an update message is triggered to provide the details of the update. The update message, like create, will contain the full details in the message for all fields, including custom flex attributes.

Table 2-72 Department Create and Update

Message Element	Included?	Notes
Department	Always	Contains the number which uniquely identifies the department.
Department Name	Always	Contains the description which identifies the department.
Buyer	Optional	Contains the number which uniquely identifies the buyer for that department.
Purchase Type	Always	Contains a code which indicates whether items in this department are, by default, owned (normal) merchandise, consignment, or concession. Items in the department can be a mix of these item types, this acts only as a default. Valid values are: 0 - Normal Merchandise 1 - Consignment Stock 2 - Concession Stock
Total Market Amount	Optional	Contains the total market amount that is expected for the department. The field can be used for analytics or reporting.
Merchandiser	Optional	Contains the number which uniquely describes the merchandiser for that department.
Group Number	Always	Contains the number of the group in which the department exists.
Budgeted Markup	Always	Contains the markup percent of cost that is a target for this department.
Profit Calc Type	Always	Contains a number which indicates whether profit will be calculated by cost or retail, indicating the accounting method to be used for the department. Valid values are: 1 = Direct Cost 2 = Retail Inventory
Markup Calc Type	Always	Contains the code which determines how markup is calculated in this department. Valid values are: C - Cost R - Retail
OTB Calc Type	Always	Contains the code letter which determines how open to buy is calculated in this department. Valid values are: C - Cost R - Retail

Table 2-72 (Cont.) Department Create and Update

Message Element	Included?	Notes
Maximum Average Counter	Optional	A maximum average counter will hold the maximum count of days with acceptable data to include in an average for items within the department.
Average Tolerance Percentage	Optional	A tolerance percentage value used in averaging for items within this value. This value will set up a range for appropriate data and constrain outliers. This column will hold 70% as 70 - not 0.70.
Budgeted Intake	Always	Contains the markup percent of retail that is a target for this department.
Department VAT Include Indicator	Always	This flag is no longer used by Merchandising and will always be defaulted to Y.
Flex Attributes	Optional	Child Node

Deleted Department

When an existing department is deleted, this will trigger a delete transaction message to notify external systems that this department is no longer valid. The delete message will include only the department ID being deleted.

Table 2-73 Department Delete

Message Element	Included?	Notes
Department	Always	Contains the number which uniquely identifies the department being deleted.

New Class

When a new class is created in Merchandising, it triggers a message notifying external systems. The full details are sent for the new class as part of the create message, the class ID, class name, class VAT indicator, and department. The custom flex attributes, if applicable, are also sent as a part of the message.

Updated Class

When an existing class is updated, an update message is triggered to provide the details of the update. The update message, like create, will contain the full details in the message for all fields including custom flex attributes.

Table 2-74 Class Create and Modify

Message Element	Included?	Notes
Class	Always	Contains the number which uniquely identifies the class within the department.
Class name	Always	Contains the name of the class which, along with the class number, identifies the class.

Table 2-74 (Cont.) Class Create and Modify

Message Element	Included?	Notes
Class VAT Indicator	Always	This field is no longer used by Merchandising and will always default to N.
Department	Always	Contains the number of the department of which the class is a member.
Unique Class ID	Always	This field contains the number that uniquely identifies the department/class combination. This column is populated by the system and not displayed to end users. This will be used for integration to downstream systems as many downstream systems are unable to accommodate the composite dept/class key in their hierarchy and item structures.
Custom Flex Attributes	Optional	Child Node

Deleted Class

When an existing class is deleted, this will also trigger a delete transaction message to notify external systems that this class is no longer valid. The delete message will include only the class ID being deleted.

Table 2-75 Class Delete

Message Element	Included?	Notes
Department	Always	Contains the number which uniquely identifies the department.
Class	Always	Contains the number which uniquely identifies the class within the department.

New Subclass

When a new subclass is created in Merchandising, it triggers a message notifying external systems. The full details are sent for the new subclass as part of the create message, the subclass ID, subclass name, department, and class. It also contains a unique ID for the subclass and class, which are not dependent on knowing the department and class IDs displayed in Merchandising. The custom flex attributes, if applicable, are also sent in as a part of the message.

Updated Subclass

When an existing subclass is updated, an update message is triggered to provide details of the update. The update message, like create, will contain the full details in the message for all fields including custom flex attributes.

Table 2-76 Subclass Create and Update

Message Element	Included?	Notes
Subclass	Always	Contains the number which uniquely identifies the subclass within the department and class.
Subclass name	Always	Contains the name of the subclass which, along with the subclass number, uniquely identifies the subclass.
Class	Always	Contains the class number of which the subclass is a member.
Department	Always	Contains the department number of which the subclass is a member.
Custom Flex Attributes	Optional	Child Node
Unique Subclass ID	Always	This field contains the number that uniquely identifies the department/class/subclass combination. This column is populated by the system and not displayed to end users. This will be used for integration to downstream systems as many downstream systems are unable to accommodate the composite dept/class/subclass key in their hierarchy and item structures.
Unique Class ID	Always	This field contains the number that uniquely identifies the department/class combination. This column is populated by the system and not displayed to end users. This will be used for integration to downstream systems as many downstream systems are unable to accommodate the composite dept/class key in their hierarchy and item structures.

Deleted Subclass

When an existing subclass is deleted, this will trigger a delete transaction message notifying external systems that this class is no longer valid. The delete message will include only the subclass being deleted, along with its class and department.

Table 2-77 Subclass Delete

Message Element	Included?	Notes
Department	Always	Contains the number which uniquely identifies the department.
Class	Always	Contains the number which uniquely identifies the class within the department.
Subclass	Always	Contains the number which uniquely identifies the subclass within the department and class.

Custom Flex Attributes

If any custom flex attributes (CFAS) for the department, class, or subclass have been added or modified, it will trigger an update message. The node of the integration that supports this will contain the name of the attribute as it is defined in the group set level view, the value of the custom attribute. If it is a date attribute, the date value is in a separate field. Flex attributes can only be added to or updated; they cannot be deleted.

Table 2-78 Flex Attributes

Message Element	Included?	Notes
Name	Always	Holds the attribute name.
Value	Optional	Holds the value of the attribute for number and character type attributes
Value Date	Optional	Holds the date for date type attributes.

Error Handling

When the publication encounters a non-fatal error, messages continue to be processed. For the message where the error was encountered, a status of Hospital (H) is sent to the RIB and the status of the message in the queue is set to H. In case the error is a fatal error, a status of Error (E) is sent to the RIB and the next message in the queue is not retrieved until the error is resolved.

Message XSD

Below are the filenames that correspond with each message type. Please consult the Oracle Retail Integration Guide for each message type for the details on the composition of each message.

Message Types	Message Type Description	XML Schema Definition (XSD)
divisoncre	Division Create Message	MrchHrDivDesc.xsd
divisonmod	Division Modify Message	MrchHrDivDesc.xsd
divisiondel	Division Delete Message	MrchHrDivRef.xsd
groupcre	Group Detail Create Message	MrchHrGrpDesc.xsd
groupmod	Group Detail Modify Message	MrchHrGrpDesc.xsd
groupdel	Group Detail Delete Message	MrchHrGrpRef.xsd
deptcre	Department Detail Create Message	MrchHrDeptDesc.xsd
deptmod	Department Detail Modify Message	MrchHrDeptDesc.xsd
deptdel	Department Detail Delete Message	MrchHrDeptRef.xsd
classcre	Class Detail Create Message	MrchHrClsDesc.xsd
classmod	Class Detail Modify Message	MrchHrClsDesc.xsd
classdel	Class Detail Delete Message	MrchHrClsRef.xsd
subclasscre	Subclass Detail Create Message	MrchHrScIsDesc.xsd
subclassmod	Subclass Detail Modify Message	MrchHrScIsDesc.xsd
subclassdel	Subclass Detail Delete Message	MrchHrScIsRef.xsd

Order Publication API

This section describes the order publication API.

Functional Area

Purchase Orders

Business Overview

Merchandising publishes new and updated purchase orders to inform external subscribing systems, such as the warehouse management system or store inventory management (SIM/SIOCS), of the changes. Only orders that have been approved are published and will be visible to external systems. Updates are provided synchronously in a near-real-time manner.

There are several ways a purchase order can be created in Merchandising, such as through the purchase order screen, using spreadsheet upload, from customer orders and franchise orders fulfilled from a supplier, through item replenishment, by a supplier in a vendor-managed inventory environment, and so on. All purchase orders are published in a similar manner, regardless of their source.

Merchandising publishes two sets of order messages to the RIB for two kinds of subscribing applications. The primary order information is sent at the physical location level, which includes both physical warehouses and stores. This is intended to be used by the warehouse and store applications. The second set of information contains order quantity by virtual location. Applications that understand virtual warehouses subscribe to these messages.

New Purchase Orders

Creating a new purchase order triggers a message that notifies external systems of the changes. The full details in the message are sent for the new order as part of the create message. It includes details such as which supplier is being ordered from, order dates, the items that are ordered and the quantity, the size of the pack being ordered, the location that each item is going to be delivered to, and so on. If the order is for a customer order, (order_type = 'CO'), the customer order number and fulfillment order number retrieved from the ORDCUST table will be included in the header message and published.

New Purchase Order Details

Adding item/locations to an existing order triggers a detail create message to be created and published after the order is re-approved. This action will create one header modify message for the status change and each detail addition will create one detail create message. The detail create message will still contain both complete header and detail information.

Purchase Order Header

Message Element	Included?	Notes
Document Type	Always	This field indicates the type of document this message is for. Valid value is P for purchase order.
Order Number	Always	This field contains the number to uniquely identify an order within the system.
Order Type	Always	This field indicates the type of order. Valid values are: <ul style="list-style-type: none"> • N/B - Non Basic • ARB - Automatic Reorder of Basic • BRB - Buyer Reorder of Basic • CO - Customer Order

Message Element	Included?	Notes
Order Type Description	Always	This field contains the order type description as shown above.
Department	Optional	This field contains the department number for orders limited to a single department and will be null for orders involving items in more than one department.
Department Name	Optional	This field contains the name of the department.
Buyer	Optional	This field contains the number associated with the buyer for the order.
Buyer Name	Optional	This field contains the name of the authorized buyer.
Supplier	Always	This field contains the supplier site that will provide the merchandise specified in the order.
Promotion	Optional	This field contains the promotion associated with the order.
Promotion Description	Optional	This field contains a description of the promotion. This value comes from Pricing.
QC Indicator	Always	This field determines whether quality control will be required when items for this order are received. Valid values are Yes (Y) and No (N).
Not Before Date	Optional	This field contains the first date that delivery of the order will be accepted.
Not After Date	Optional	This field contains the last date that delivery of the order will be accepted.
Open To Buy End of Week Date	Optional	This field contains the OTB week the order amount should be placed into for the order type.
Earliest Ship Date	Optional	This field contains the earliest date when the items on the purchase order can be shipped by the supplier. This represents the earliest ship date of all the items on the order.
Latest Ship Date	Optional	This field contains the date after which the items on the purchase order can no longer be shipped by the supplier. This represents the greatest latest ship date of all the items on the order.
Close Date	Optional	This field contains the date when the order was closed.
Terms	Always	This field is an indicator identifying the sales terms for the order. These terms specify when payment is due and if any discounts exist for early payment.
Terms Code	Always	This field contains the alphanumeric representation of Term Name which acts as the Term code in Oracle Financials.
Freight Terms	Optional	This field contains the ID of the freight terms are related to the order. Valid values are in the FREIGHT_TERMS table.
Customer Order Indicator	Always	This field indicates whether the order is generated for a customer order fulfillment.

Message Element	Included?	Notes
Payment Method	Optional	This field indicates how the purchase order will be paid. Valid options are: <ul style="list-style-type: none"> • LC - Letter of Credit • WT - Wire Transfer • OA - Open Account
Payment Method Description	Optional	This field contains the description of the payment method as shown above.
Backhaul Type	Optional	This field contains the type of backhaul allowance that will be applied to the order. Valid values are Calculated (C) or Flat rate (F).
Backhaul Type Description	Optional	This field contains the description of the backhaul type as shown above.
Backhaul Allowance	Optional	This field contains the backhaul allowance amount.
Ship Method	Optional	This field contains the method used to ship the items on the purchase order from the country of origin to the country of import.
Ship Method Description	Optional	This field contains the description for the shipping method.
Purchase Type	Optional	This field indicates what is included in the supplier's cost of the item. Valid values are: <ul style="list-style-type: none"> • BACK - Backhaul • DLV – Delivered • FOB – Free on Board • PUP – Pick-up
Purchase Type Description	Optional	Contains the description for the purchase type as shown above.
Status	Always	Indicates the current status of the order. Valid values include: <ul style="list-style-type: none"> • W - Worksheet • S - Submitted • A - Approved • C - Closed
Ship Pay Method	Optional	This field contains the code indicating the payment terms for freight charges associated with the order. Valid values are found in code type SHMT.
Ship Pay Method Description	Optional	This field contains the description of the given shipment payment method. Valid values are found in code type SHMT.
Transportation Responsibility	Optional	This field contains the code indicating the type of location that is responsible for the transportation of the order. Valid values are found in code type FOBT.
Transportation Responsibility Code Description	Optional	This field contains the description of the transportation responsibility code. Valid values are found in code type FOBT.
Transportation Responsibility Description	Optional	This field contains a user entered field describing the code for the location responsible for the transportation of the order.

Message Element	Included?	Notes
Title Pass	Optional	This field contains the code indicating where the title for goods is passed from the vendor to the purchaser. Valid values are found in code type FOBT.
Title Pass Code Description	Optional	This field contains the description for title pass code. Valid values are found in code type FOBT.
Title Pass Description	Optional	This contains a user entered field describing the code where the title of the merchandise is to be passed.
Vendor Order Number	Optional	This field contains the vendor's number for an order. These orders may have originated by the vendor or this number can be associated to a Merchandising order when the order is created on-line.
Exchange Rate	Optional	The exchange rate used to convert the order cost from the order's currency to local currency. This value is used when the entity exchange rate configuration's Ignore Entity Exchange Rate option for purchase order cost is set to No (N). When the Ignore Entity Exchange Rate flag is Yes (Y), the exchange rate is ignored and the value from the Merchandising currency rates table on receipt date or approval date is used.
Factory	Optional	This field contains the factory at which the items on the purchase order are made. This field is only available when the system is running Import Management functionality.
Factory Description	Optional	This field contains the description of the factory.
Agent	Optional	This field contains the agent that is used in the purchase of the items on the purchase order. This field is only available when the system is running Import Management functionality.
Agent Description	Optional	This field contains the description of the agent.
Discharge Port	Optional	This field contains the port at which the items on the purchase order will enter the country of import. This field is only available when the system is running Import Management functionality.
Discharge Port Description	Optional	This field contains the description of the discharge port.
Lading Port	Optional	This field contains the port from which the items on the purchase order are shipped. This field is only available when the system is running Import Management functionality.
Lading Port Description	Optional	This field contains the description of the lading port.
Freight Contract Number	Optional	This field contains the number of the contract with a shipper that will give specific freight rates. This field is only available when the system is running Import Management functionality.
Order Context (PO Type)	Optional	Contains a code representing the order context or type, which is a retailer defined attribute. Valid values are in the PO_TYPE table.
PO Type Description	Optional	This field contains the description of the PO Type.

Message Element	Included?	Notes
Pre Mark Indicator	Always	This field indicates whether a supplier has agreed to pre-mark boxes on the order for the final destination location for a cross-docked order. Valid values are Yes (Y) and No (N).
Currency Code	Always	This field contains the currency code for the order. When creating an order within Merchandising, this field will always default to the currency of the supplier, but it can be overwritten.
Contract Number	Optional	This field contains the contract number associated with this order.
Pickup Location	Optional	This field contains the location at which the order will be picked up, if the order is a Pickup order.
Pickup Number	Optional	This field contains the reference number for the Pickup order.
Pickup Date	Optional	This field contains the date when the order can be picked up from the supplier. This field is only required if the Purchase Type of the order is Pickup.
Appointment Datetime	Optional	This field contains the date and time of the receiving appointment at the warehouse.
Comments	Optional	This field contains the miscellaneous comments attached to the purchase order.
Customer Order Number	Optional	This field holds the customer order number from the order management system for orders to be fulfilled from the supplier.
Fulfill Order Number	Optional	This field holds the number from the order management system (OMS) related to the fulfillment details. One or more fulfillment orders could relate back to a single customer order in OMS.
PO Detail	Optional	Child node
Flex Attribute	Optional	Child node
Reinstate Indicator	Always	This field indicates if the order was previously cancelled and has now been re-approved. It will contain a value of Y when the order is first reinstated. Any subsequent updates to the PO will have a value of N.

Purchase Order Detail

This component of the order rolls the order quantities for the purchase order up to the physical warehouse level. For stores, the quantity values here are the same as on the purchase order.

Message Element	Included?	Notes
Item	Always	This field contains the unique alphanumeric value to identify the item included in the purchase order.
Reference Item	Optional	This field contains the unique alphanumeric value to identify the reference item (barcode).
Physical Location Type	Always	This field contains the type of location in the location field. Valid values are store (S) or warehouse (W).

Message Element	Included?	Notes
Physical Location	Always	This field contains the actual physical location that the item will be shipped to. This field may contain a store or a physical warehouse.
Physical Store Type	Optional	This field indicates the store type of the physical location, if the physical location type is a store. Valid values are company store (C) or franchise store (F).
Physical Stockholding Indicator	Optional	Indicates if the store is stockholding or not, if the physical location type is a store. Valid values are Yes (Y) and No (N).
Physical Quantity Ordered	Optional	This field indicates the total quantity ordered to the physical location.
Transaction UOM	Optional	This field holds the original transaction unit of measure to be passed on to the store system. It is only used for customer orders shipped to a store for fulfillment. This will be based on the standard UOM for the item.
Unit Cost	Optional	This field contains the unit cost for the item/location on the order, including discounts, but not landed cost components in order currency.
Origin Country ID	Optional	This field contains the identifier of the country into which the items on the order are being imported.
Supplier Pack Size	Optional	This field contains the supplier pack size for this order/item.
Earliest Ship Date	Optional	This field contains the earliest date that the item can be shipped by the supplier.
Latest Ship Date	Optional	The date after which the item can be shipped by the supplier.
Pickup Location	Optional	This field contains the location at which the item will be picked up, if the order is a Pickup order.
Pickup Number	No	Not used in Merchandising
Packing Method	Optional	This field indicates whether the packing method of the item in the container is flat or hanging. Valid values are defined under code type PKMT.
Round Level	Optional	This column will be used to determine how order quantities will be rounded. Valid values are in code type ORL.
Door Indicator	No	Not used in Merchandising
Priority Level	No	Not used in Merchandising
New Item	Optional	This flag is used to identify a new item versus an existing item on the order. This is set to Y only if the system option for Publish RIB Objects is set to Full and Deltas (Y).
Quarantine	No	Not used in Merchandising
Received Unit Quantity	No	Not used in Merchandising
TSF PO Link ID	Optional	This field contains a reference number to link the item on a transfer to the purchase orders. This is used by orders created for items on Warehouse Cross Link replenishment.
PO Virtual Location Detail	Optional	Child node

Message Element	Included?	Notes
Cost Source	Optional	This field indicates the basis for the cost of the item/location on the order. Valid values are found in code type ORCS.
Estimated In-Stock Date	Optional	This field contains the date that the item on the PO is expected to be in stock at the location.
Flex Attribute	Optional	Child node
Item Line Number	Optional	This field indicates the item line number. This is used for customer order POs.

Purchase Order Virtual Location Detail

This node of the message breaks out the order quantities down to the virtual warehouse level. For stores, the order quantities will be the same as in the physical location detail node.

Message Element	Included?	Notes
Location Type	Always	This field contains the type of location in the location field. Valid values are store (S) or virtual warehouse (W).
Location	Always	This field contains the store number or the virtual warehouse number, depending on the virtual location type.
Quantity Ordered	Optional	This field contains the total quantity of the item ordered to this location.
Flex Attribute	Optional	Child node

Updated Purchase Orders

Purchase order updates trigger a message to notify external systems, based on updates made at the order header level and/or the detail level. The message that will be published will either contain the header information that changed in the message for header updates such as date changes, just the detail level information for detail level changes such as quantity changes or addition of items, or both.

Header Only Updates

When updates are made at the order header level such as updating dates or changing the status of the order, only header information is published. Information specific to items and locations are not included. Changes in header information, unapproving a purchase order, or any action that may change of status of a pre-approved order, such as closing or reinstating an order trigger header level publication. Creating and updating header level custom flex attributes (CFAS) also trigger header level update publishing.

Detail Only Updates

When updates are made at the order detail level without changes to the header information, only detail information is published. Detail level updates include changes in quantity ordered, unit cost, and estimated in stock date. When an order has already been approved, no deletes can be made at the detail level; but the quantity on the order can be cancelled. This action will trigger an update message to be sent. Creating and updating detail level custom flex attributes also trigger detail level publishing.

Full Message Updates

In cases where the solution receiving purchase orders cannot support just receiving the changes, another option is provided that can resend the full purchase order details whenever there is a change. This will be published, along with the delta messages, in cases where the system option Publish Full Objects (PUB_FULL_OBJECTS_IND) is set to Deltas and Full (Y). This message payload will contain a full snapshot of the order with the flex attributes (of both header and detail). This is also used for Oracle WMS Cloud integration.

Deleted Purchase Orders

Deleting approved purchase orders can be done by cancelling all items in the order, which will set the order status to closed. The purchase order will not be removed from the system. Because both the order status and order quantities are changed, both header and detail information are published to external systems. The information included in the header and detail messages are the same as that described under the Updated Purchase Orders section.

Table 2-79 PO Header Delete Message

Message Element	Included?	Notes
Document Type	Always	This field indicates the type of document this message is for. Valid value is P for purchase order.
Order Number	Always	This field contains the number that uniquely identifies an order within the system.
PO Detail	Optional	Child node

Table 2-80 PO Detail Delete Message

Message Element	Included?	Notes
Item	Always	This field contains the unique alphanumeric value to identify the item being deleted.
Physical Location Type	Always	This field contains the type of physical location where an item is being deleted. Valid values are store (S) or warehouse (W).
Physical Location	Always	This field contains the identifier of the physical location of the item being deleted.
Physical Store Type	Optional	This field indicates the store type of the physical location, if a store. Valid values are company store (C) or franchise store (F).
Physical Stockholding Indicator	Optional	This field indicates if the physical location is stockholding, if the location type is store. Valid values are Yes (Y) and No (N).

Flex Attributes

If any custom flex attributes (CFAS) for the order have been added or modified, it will trigger an order header or detail update message, as described above. The node of the integration that supports this will contain the name of the attribute as it is defined in

the group set level view, the value of the custom attribute. If it is a date attribute, the date value is in a separate field. You can define group set views at the header level (ORDHEAD), order/item level (ORDSKU) or at the order/item/location (ORDLOC) levels. Flex attributes can only be added to or updated on an order; they cannot be deleted.

Table 2-81 Flex Attributes

Message Element	Required?	Notes
Name	Always	Holds the attribute name.
Value	Optional	Holds the value of the attribute for number and character type attributes
Value Date	Optional	Holds the date for date type attributes.

Error Handling

When the publication encounters a non-fatal error, messages continue to be processed. For the message where the error was encountered, a status of Hospital (H) is sent to the RIB and the status of the message in the queue is set to H. These types of errors occur when no changes in the database have been made and a process to try to re-publish these messages is available. In case the error is a fatal error, for example, when changes to data have already been made, a status of Error (E) is sent to the RIB and the message status in the queue will be in Error status. The error message as well as the object containing the order number and order details is returned to Merchandising.

Message XSD

Below are the filenames that correspond with each message type. Please consult the Oracle Retail Integration Guide for each message type for the details on the composition of each message.

Message Types	Message Type Description	XML Schema Definition (XSD)
POCre	Purchase Order Create Message	PODesc.xsd
POCre (CustFlexAttriVo)	Purchase Order Flex Attribute Create Message	PODesc.xsd
PODtICre	Purchase Order Detail Create Message	PODesc.xsd
PODtICre (CustFlexAttriVo)	Purchase Order Detail Flex Attribute Create Message	PODesc.xsd
POHdrMod	Purchase Order Modify Message	PODesc.xsd
POHdrMod (CustFlexAttriVo)	Purchase Order Header Flex Attribute Modify Message	PODesc.xsd
PODtIMod	Purchase Order Detail Modify Message	PODesc.xsd
PODtIMod (CustFlexAttriVo)	Purchase Order Detail Flex Attribute Modify Message	PODesc.xsd
PODel	Purchase Order Delete Message for unpublished orders	PORef.xsd
PODtIDel	Purchase Order Detail Delete Message for unpublished orders	PORef.xsd
POFulRep	Purchase Order with Full payload Message	PODesc.xsd

Organizational Hierarchy Publication API

This section describes the organization hierarchy publication API.

Functional Area

Organizational Hierarchy

Business Overview

Merchandising publishes details about new, updated, and deleted levels of the organizational hierarchy (chain, area, region, and district) to external systems such that all the downstream applications and external systems may be informed of the updates. Publishing is provided synchronously in a near-real-time manner.

The entities sent as part of the new, update, or delete messages will always include the hierarchy level and hierarchy value. The following values will be published:

- If the message relates to chain, the message will contain a hierarchy level of 'CH' and the chain number.
- If the message relates to area, the message will contain a hierarchy level of 'AR' and the area number.
- If the message relates to a region, the message will contain a hierarchy level of 'RE' and the region number.
- If the message relates to a district, the message will contain a hierarchy level of 'DI' and the district number.

New Chains/Areas/Regions/Districts

Creating a new organizational level triggers a message to notify external systems. The hierarchy level, hierarchy value, description, manager name, currency code, parent ID and parent level are sent for the new organizational level as part of the create message.

Table 2-82 Organization Hierarchy – Create and Update

Message Element	Included?	Notes
Hierarchy level	Always	This field contains the level of the organizational hierarchy. Valid values are: <ul style="list-style-type: none"> • CH - Chain • AR - Area • RE - Region • DI - District
Hierarchy Value	Always	This field contains the hierarchy value.
Hierarchy Description	Optional	This field contains the description of the hierarchy value.
Manager Name	Optional	This contains the manager's name.
Currency Code	Optional	This field contains the currency code for the hierarchy.

Table 2-82 (Cont.) Organization Hierarchy – Create and Update

Message Element	Included?	Notes
Parent ID	Optional	This field contains the number that identifies the hierarchy level.
Parent Level	Optional	This field contains the parent level of the current organizational hierarchy. Valid values are: <ul style="list-style-type: none"> • CH - Chain • AR - Area • RE - Region

Updated Chains/Areas/Regions/Districts

Updating an organizational level triggers a message to notify external systems. The hierarchy level, hierarchy value, description, manager name, currency code, parent ID and parent level are sent for the updated organizational level as part of the update message.

Deleted Chains/Areas/Regions/Districts

Deleting an organizational level triggers a message to be sent to notify external systems. The hierarchy level, hierarchy value, parent ID, and parent level are sent for the deleted organizational level.

Table 2-83 Organization Hierarchy – Delete

Message Element	Included?	Notes
Hierarchy level	Always	This field contains the level of the organizational hierarchy. Valid values are: <ul style="list-style-type: none"> • CH - Chain • AR - Area • RE - Region • DI - District
Hierarchy Value	Always	This field contains the hierarchy value.
Parent ID	Optional	This field contains the number that identifies the hierarchy level.
Parent Level	Optional	This field contains the parent level of the current organizational hierarchy. Valid values are: <ul style="list-style-type: none"> • CH - Chain • AR - Area • RE - Region

Error Handling

If any errors are encountered while publishing the message, a fatal error with status E (Error) is sent to RIB. This would bring down the RIB adapter thus preventing any further messages from being processed until this is resolved.

Message XSD

Here are the filenames that correspond with each message type. Please consult the RIB documentation for each message type in order to get a detailed picture of the composition of each message.

Message Types	Message Type Description	XML Schema Definition (XSD)
chaincre	Chain Create Message	OrgHierDesc.xsd
chainmod	Chain Modify Message	OrgHierDesc.xsd
chaindel	Chain Delete Message	OrgHierRef.xsd
areacre	Area Create Message	OrgHierDesc.xsd
areamod	Area Modify Message	OrgHierDesc.xsd
areadel	Area Delete Message	OrgHierRef.xsd
regioncre	Region Create Message	OrgHierDesc.xsd
regionmod	Region Modify Message	OrgHierDesc.xsd
regiondel	Region Delete Message	OrgHierRef.xsd
districtcre	District Create Message	OrgHierDesc.xsd
districtmod	District Modify Message	OrgHierDesc.xsd
districtdel	District Delete Message	OrgHierRef.xsd

Partner Publication API

This section describes the partner publication API.

Functional Area

Partners

Business Overview

This API publishes partner, partner address information and custom flex attributes through the RIB to external systems. This information is further subscribed to by integrating systems like Oracle Warehouse Management Cloud (WMS) and Store Inventory and Operations Cloud Service (SIOCS) that need to keep their partners synchronized with Merchandising. Partner information is published when new partners are created, updates are made to existing partners, or existing partners are deleted. Similarly, addresses are published when they are added, modified, or deleted.

Partner Creation

Partners can be setup in Merchandising through the application UI and it results in this message being published when a partner is created. The partner creation message is published from Merchandising only after the partner has been completely setup, including the mandatory addresses. Once all these criteria are met for a valid create message, the messages will be combined and sent to the RIB.

Table 2-84 Partner Header Create and Update

Message Element	Required?	Notes
Partner Type	Always	Specifies the type of the partner. Valid partner types are held on the code tables under code type PTNR.
Partner ID	Always	Unique number that identifies the partner within the system.
Partner Description	Always	Contains the partner's name.
Currency Code	Always	The code identifying the currency the partner uses for business transactions.
Language	Optional	The partner's preferred language.
Status	Always	Indicates if the partner is currently active. Valid values are A = Active or I = Inactive.
Contact Name	Optional	Name of the partner's representative contact
Contact Phone	Optional	Phone number of the partner's representative contact.
Contact Fax	Optional	Fax number of the partner's representative contact.
Contact Telex	Optional	The telex number of the partner's representative contact.
Contact Email	Optional	The e-mail address of the partner's representative contact.
Manufacturer's ID	Optional	The manufacturer's tax identification number. Used for partner type Manufacturer.
Principal Country ID	Optional	The country id to which the partner is assigned.
Line of Credit	Optional	The line of credit the company has at the bank in the partner's currency. Used for partner type Bank.
Outstanding Credit	Optional	The total amount of credit that the company has used or has charged against in the partner's currency.
Open Credit	Optional	The total amount that the company can still charge against in the partner's currency.
Year to Date Credit	Optional	The total amount of credit the company has used this year to date in the partner's currency.
Year to Date Drawdowns	Optional	The year to date payments the bank has made on behalf of the company in the partner's currency.
Tax ID	Optional	The unique tax identification number of the partner. This will be used for reporting during the Customs Entry process.
Terms	Always	Payment terms for the partner.
Service Performed Required Indicator	Always	Indicates if the partner's services (e.g., snow removal) must be confirmed as performed before paying an invoice from that partner.
Invoice Pay Location	Optional	Indicates where invoices from this supplier are paid - at the store or centrally through corporate accounting.
Invoice Receive Location	Optional	Indicates where invoices from this supplier are received - at the store or centrally through corporate accounting.
Import Country ID	Optional	Import country of the Import Authority.
Primary Import Authority Indicator	Always	Indicates if an Import Authority is the primary Import Authority for an import country.

Table 2-84 (Cont.) Partner Header Create and Update

Message Element	Required?	Notes
Comments	Optional	Comments associated with the partner.
Transfer Entity ID	Optional	The Transfer entity with which an external finisher is associated.
VAT Region	Optional	Tax region with which a partner is associated.
Org Unit ID	Optional	The org unit id with which partner is associated.
Partner Name Secondary	Optional	Secondary name of the partner.
Auto Receive Stock Indicator	Optional	Indicates whether the system will update the stock for the external finisher when the 1st leg of the transfer is shipped.
Partner Address 1	Optional	The partner's primary address line 1.
Partner Address 2	Optional	The partner's primary address line 2.
City	Optional	The city of the partner's primary address.
County	Optional	The county of the partner's primary address.
State	Optional	The state of the partner's primary address.
Country ID	Optional	The country of the partner's primary address.
Postal Code	Optional	The postal code of the partner's primary address.
Partner Org Unit Details	Optional	Child Node
Address Details	Optional	Child Node
Flex Attributes	Optional	Child Node

Table 2-85 Partner Org Unit Detail

Message Element	Required?	Notes
Org Unit ID	Optional	Org unit ID of the partner.

Table 2-86 Address Detail

Message Element	Required?	Notes
State Name	Optional	The state name for the partner address of this type.
Country Name	Optional	The country name for the partner address of this type.
Address Key	Always	The unique address ID from Merchandising.
Address Type	Always	The type for the address. All address types are published for partners.
Primary Address Type Indicator	Optional	Indicates if the address type is primary.
Primary Address Indicator	Always	Indicates whether this address is the primary address for the address type.
Address Line 1	Always	The first line of the address.
Address Line 2	Optional	The second line of the address.
Address Line 3	Optional	The third line of the address.

Table 2-86 (Cont.) Address Detail

Message Element	Required?	Notes
City	Always	The name of the city that is associated with the address.
State	Optional	The state or province code for the address.
Country ID	Always	The country code for the address.
Post	Optional	The postal code for the address.
Contact Name	Optional	The name of the contact for the partner at this address.
Contact Phone	Optional	The phone number of the contact person at this address.
Contact Telex	Optional	The telex number of the partner representative contact.
Contact Fax	Optional	The fax number of the contact person at this address.
Contact Email	Optional	The email address of the partner representative contact.
Oracle Vendor Site ID	Optional	The unique identifier of this address in the Oracle Financials systems, if used.
County	Optional	The county name that is associated with the address.
Jurisdiction Code	Optional	ID associated to the tax jurisdiction of the country-state relationship.
Flex Attributes	Optional	Child Node

Partner Header Modification

Updating an existing partner will trigger header modification message to notify the integrating systems. This message will contain the header level information, the primary address and flex attributes of the partner that changed.

Partner Detail Creation

Adding new addresses for an existing partner will trigger a partner detail creation message. This message is sent along with the header information.

Partner Detail Modification

Any updates made to the partner's address information will trigger a partner detail modification message. This message is sent along with the partner header information.

Deleting Partners

Deleting an existing partner will trigger a partner delete message to notify the integrating systems. This message will contain the partner type and partner ID that was deleted.

Deleting Partner Details

Partner addresses can be deleted only if they are not mandatory or if they are mandatory but are not marked as the primary address for that address type. The deletion will trigger a partner detail delete message. The message will contain the partner type, partner ID and the address key of the address that was deleted.

Table 2-87 Partner Delete

Message Element	Required?	Notes
Partner Type	Always	Specifies the type of the partner.
Partner ID	Always	Unique number that, with the partner type, identifies the partner being deleted.
Address Detail	Optional	Child Node

Table 2-88 Address Detail Delete

Message Element	Required?	Notes
Address Key	Always	The unique identifier of the address being deleted.

Flex Attributes

If any custom flex attributes (CFAS) for the partner or partner address have been added or modified, it will trigger an update message. The node of the integration that supports this will contain the name of the attribute as it is defined in the group set level view, the value of the custom attribute. If it is a date attribute, the date value is in a separate field. Flex attributes can only be added to or updated; they cannot be deleted.

Table 2-89 Flex Attributes

Message Element	Required?	Notes
Name	Always	Holds the attribute name.
Value	Optional	Holds the value of the attribute for number and character type attributes
Value Date	Optional	Holds the date for date type attributes.

Error Handling

When the publication encounters a non-fatal error, messages continue to be processed. For the message where the error was encountered, a status of Hospital (H) is sent to the RIB and the status of the message in the queue is set to H. In case the error is a fatal error, a status of Error (E) is sent to the RIB and the next message in the queue is not retrieved until the error is resolved.

Message XSD

Below are the filenames that correspond with each message type. Please consult the Oracle Retail Integration Guide for each message type for the details on the composition of each message.

Message Types	Message Type Description	XML Schema Definition (XSD)
partnercre	Partner Create	PartnerDesc.xsd
partnermod	Partner Modify	PartnerDesc.xsd
partnerdel	Partner Delete	PartnerRef.xsd
partnerdtlcre	Partner Detail Create	PartnerDesc.xsd
partnerdtlmod	Partner Detail Modify	PartnerDesc.xsd
Partnerdtldel	Partner Detail Delete	PartnerRef.xsd

Receiver Unit Adjustment Publication API

This section describes the receiver unit adjustment publication API.

Functional Area

Receipts

Business Overview

When mistakes are made during the receiving process at the store or warehouse, receiver unit adjustments (RUAs) are made to correct the mistake. Merchandising publishes messages about receiver unit adjustments in cases where RUAs are initiated through Invoice Matching or Merchandising, a message is published to store and warehouse inventory systems, like Oracle Retail Store Inventory and Operations Cloud Service (SIOCS), to keep inventory in sync.

**Note:**

Oracle WMS Cloud and Oracle Retail Warehouse Management (RWMS) do not subscribe to Receiver Unit Adjustment messages from Merchandising.

Creating Receiver Unit Adjustments

When a receiver unit adjustment is created, it triggers a message to be sent to notify external systems. The full details are sent for the receiver unit adjustment as part of the create message, as shown below.

Table 2-90 Receiver Unit Adjustment Header

Message Element	Required?	Notes
Location	Always	The location (store or physical warehouse) impacted by the receiver unit adjustment.
Location Type	Always	This is the type of location in the location field. Valid values are Store (S) or Warehouse (W).

Table 2-91 Receiver Unit Adjustment Detail

Message Element	Required?	Notes
Order Number	Always	This contains the order number against which a receiver unit adjustment was performed
ASN Number	Optional	This contains the ASN number associated with the shipment for which a receiver unit adjustment was performed.
Item Number	Always	This contains item whose receipt quantity was adjusted.
Carton ID	Optional	This contains the carton that contained the item against which the receiver unit adjustment was performed, if applicable.
Adjustment Quantity	Always	This contains the quantity, either positive or negative, for the adjustment.
From Disposition	Optional	If the adjustment quantity is less than zero, then this will contain ATS. If it is a positive adjustment, then this will be NULL.
To disposition	Optional	If the adjustment quantity is greater than zero, then this will contain ATS. Otherwise, this will be NULL.
Universal identification number (UIN)	optional	Not used by Merchandising.
Status of UIN	optional	Not used by Merchandising.

Error Handling

If any errors are encountered while publishing the message, a fatal error with status E (Error) is sent to RIB. This would bring down the RIB adapter, thus preventing any further messages from being processed until this is resolved.

Message XSD

Below are the filenames that correspond with each message type. Please consult the *Oracle Retail Integration Guide* for each message type for the details on the composition of each message.

Message Types	Message Type Description	XML Schema Definition (XSD)
rcvunitadjcre	Receive unit adjust Create	RcvUnitAdjDesc.xsd
rcvunitadjcre	Receive unit adjust Create Detail	RcvUnitAdjDtl.xsd

RTV Request Publication API

This section describes the RTV request publication API.

Functional Area

Inventory Movement

Business Overview

A return to vendor (RTV) order is used to send merchandise back to the supplier. The RTV message is published by Merchandising to the store or warehouse to provide visibility to the corporately created RTV. Consequently, when the store or warehouse ships the RTV, it communicates the original RTV order number back to Merchandising to update the original RTV record.

Create RTV Request

Approval of an RTV order in Merchandising results in this message being published. The RTV Request message includes the following:

Table 2-92 RTV Request Header

Message Element	Included?	Notes
RTV Order No	Always	The number that uniquely identifies the return to vendor within the system.
Supplier	Always	The supplier site to which the merchandise is being returned.
Status Indicator	Always	The status of the return. Valid values include: 10 - Approved and 20 - Cancelled.
Location	Always	The store or virtual warehouse number from which the return will occur.
Location Type	Always	The type of location returning the items. Valid values are store (S) or warehouse (W).
Physical Location	Always	The store number or the physical warehouse from which the items are being returned.
Total Order Amount	Optional	The total amount of the return in the supplier's currency.
Ship To Address Line 1	Always	The first line of the supplier's address for returns.
Ship To Address Line 2	Optional	The second line of the supplier's address for returns.
Ship To Address Line 3	Optional	The third line of the supplier's address for returns.
Ship To City	Always	The city name of the supplier's address for returns.
State	Optional	The state code of the supplier's address for returns.
Ship To Country Code	Always	The country ID of the supplier's address for returns.
Ship To Postal Code	Optional	The postal code of the supplier's address for returns.
Return Authorization Number	Optional	The number that the supplier provides when the decision is made that an order may be returned, if required.

Table 2-92 (Cont.) RTV Request Header

Message Element	Included?	Notes
Return Courier	Optional	The courier's name if the supplier requires that returns be sent by a designated carrier.
Freight Charge	Optional	The freight cost associated with the vendor return. This field is stored in the supplier's currency.
Creation Date	Always	The date the vendor return was created.
Completed Date	Optional	The date the vendor return was completed (for updates only).
Handling Percentage	Optional	The handling (restocking) percent charged to the retailer for returns, if specified.
Handling Cost	Always	The handling (restocking) cost for the return if specified. This is specified in the supplier's currency.
External Reference No	Optional	Reference number used in an external system; only applicable if the RTV was initiated in the store or warehouse (for updates only).
Comments	Optional	Comments associated with the return.
RTV Request Details	Optional	Child Node
Not After Date	Optional	The last date that the return can be shipped to the vendor.
Flex Attributes	Optional	Child Node
Supplier Name	Always	The supplier site's name in the system's primary language.

Table 2-93 RTV Request Detail

Message Element	Included?	Notes
Sequence Number	Always	Unique number identifying the RTV Detail record.
Item	Always	Unique identifier for the item.
Shipment	Optional	Not used.
Inventory Status	Optional	The inventory status code. Only applicable if the reason field has a value of 'U' for unavailable inventory.
RTV Quantity	Always	The quantity of items requested to be returned to the supplier.
Unit Cost	Always	The cost per unit for the SKU being returned in the supplier's currency.
Reason	Always	The reason for the return. Valid values are found in code type RTVR.
From Disposition	Optional	This contains the status of the inventory that needs to be returned to the vendor. If there are multiple inventory status codes mapped to an inventory status, the values will be comma delimited. This can be comma delimited in case item is returned from multiple unavailable buckets

Update RTV Request

Updating an existing RTV request will trigger a header modification message to notify the integrating systems. After approval, limited updates can be made to an RTV that would result in a modification being published, such as updating comments or cancelling the RTV. This message will contain only the details that changed for the RTV.

Delete RTV Request

When RTV information is deleted for a closed RTV based on the history retention days defined in Merchandising's system options, the delete will trigger an RTV Request detail delete message. A header level message is not published. CFAS attributes, if associated with the RTV, are also not included in the delete publication. It is assumed the subscribing system would delete the header and CFAS attributes with the details.

The RTV Request delete message includes the following:

Table 2-94 RTV Request Delete

Message Element	Included?	Notes
RTV Order No	Always	Specifies the RTV Order number of the detail record that is being deleted.
RTV Request Detail	Optional	Child node
Location	Always	The store or warehouse number of the RTV detail record that is being deleted.
Location Type	Always	The location type of the RTV detail record that is being deleted.

Table 2-95 RTV Request Detail Delete

Message Element	Included?	Notes
Sequence Number	Always	The sequence number of the RTV detail record that is being deleted.
Item	Always	The item number of the RTV detail record that is being deleted.

Flex Attributes

If any custom flex attributes (CFAS) for the RTV Request header have been added or modified, it will trigger a header modify message. All the entity's active flex attributes from all attribute groups are published as key–value pairs based on the group set view. This CFAS object is embedded in the outbound RTV Request message.

Table 2-96 Flex Attributes Detail

Message Element	Included?	Notes
Name	Always	The flex attribute defined by the business.
Value	Optional	The value of the flex attribute defined by the business.

Table 2-96 (Cont.) Flex Attributes Detail

Message Element	Included?	Notes
Value Date	Optional	The date value of the flex attribute if flex attribute is defined as a date.

RTV Request Full Publish

When publishing a RTV Request header modification or a RTV Request detail create or modification or delete message, a second full replacement message will be published if the Publish RIB Objects system option is configured to be Deltas and Full (Y). This message will contain the header level information, flex attributes and the details of the RTV request. Based on the message type, RIB will route the full replacement message to appropriate applications. Currently this is used only by WMS Cloud.

Error Handling

When the publication encounters a non-fatal error, messages continue to be processed. For the message where the error was encountered, a status of Hospital (H) is sent to the RIB and the status of the message in the queue is set to H. In case the error is a fatal error, a status of Error (E) is sent to the RIB and the next message in the queue is not retrieved until the error is resolved.

Message XSD

Below are the filenames that correspond with each message type. Please consult the *Oracle Retail Integration Guide* for each message type for the details on the composition of each message.

Message Types	Message Type Description	XML Schema Definition (XSD)
RtvReqCre	RTV Request Create Message	RTVReqDesc.xsd
RtvReqMod	RTV Request Modify Message	RTVReqDesc.xsd
RtvReqDtIDel	RTV Request Detail Delete Message	RTVReqRef.xsd
RtvReqFulRep	RTV Request Full Replacement Message	RTVReqDesc.xsd

Season and Phase Publication API

This section describes the season phase publicatoin API.

Functional Area

Foundation

Business Overview

Merchandising publishes details about new and updated seasons and phases in order that external systems that use this information can be informed of the updates. Updates are provided synchronously in a near-real time manner.

New Seasons

Creating a new season triggers a message to be sent to notify external systems. The full details are sent for the new season as part of the create message, the season ID, description, start date and end date.

Table 2-97 Seasons Header

Message Element	Included?	Notes
Season ID	Always	This field contains the season identifier.
Season Description	Always	This field contains the description associated with the season. This is populated when creating and updating seasons.
Start Date	Always	This field contains the starting date for the season. This is populated when creating and updating seasons.
End Date	Always	This field contains the end date for the season. This is populated when creating and updating seasons.
Phases Detail	Optional	Child Node – only included when adding/updating phases

Updated Seasons

When an existing season is updated, an update message is triggered to provide the details of the update. The update message, like create, will contain the full details in the message for all fields in the message. See above for the format of an update message.

New Phases

Creating a new phase triggers a message to be sent to notify external systems. The full details are sent for the phase as part of the create message, including the ID, description, start and end date for the phase, the season in which the phase belongs.

Table 2-98 Phases Detail

Message Element	Included?	Notes
Phase ID	Always	This field contains the phase identifier.
Phase Description	Always	This field holds the description of the phase.
Start Date	Always	This field contains the starting date for the phase.
End Date	Always	This field contains the end date for the phase.

Updated Phases

When an existing phase is updated, an update message is triggered to provide the details of the update. The update message, like create, will contain the full details in the message for all fields in the message.

Deleted Seasons

When an existing season is deleted, this will also trigger a delete transaction to be sent to notify external systems that this season is no longer valid. The delete message will include only the ID of the season being deleted.

Table 2-99 Seasons Header

Message Element	Included?	Notes
Season ID	Always	This field contains the season identifier.
Phases Delete Detail	Optional	Child Node – only included when deleting phases

Deleted Phases

When an existing phase is deleted, this will also trigger a delete transaction to be sent to notify external systems that this phase is no longer valid. The delete message will include the ID of the phase being deleted and the ID of the season in which the phase belongs. See above for the full structure of the delete phase message.

Table 2-100 Phases Delete Detail

Message Element	Included?	Notes
Phase ID	Always	This field contains the phase identifier.

Error Handling

When the publication encounters a non-fatal error, messages continue to be processed. For the message where the error was encountered, a status of Hospital (H) is sent to the RIB and the status of the message in the queue is set to H. In case the error is a fatal error, a status of Error (E) is sent to the RIB and the next message in the queue is not retrieved until the error is resolved

Message XSD

Below are the filenames that correspond with each message type. Please consult the Oracle Retail Integration Guide for each message type for the details on the composition of each message.

Message Types	Message Type Description	XML Schema Definition (XSD)
seasoncre	Season Create	SeasonDesc.xsd
seasondtlcre	Phase Create	SeasonDesc.xsd
seasonmod	Season Modify	SeasonDesc.xsd

Message Types	Message Type Description	XML Schema Definition (XSD)
seasondtlmod	Phase Modify	SeasonDesc.xsd
seasondel	Season Delete	SeasonRef.xsd
seasondtldel	Phase Delete	SeasonRef.xsd

Store Publication API

This section describes the store publication API.

Functional Area

Organizational Hierarchy

Business Overview

This API publishes store and store information through RIB to external systems. Store information is published when new stores are created, updates are made to existing stores, or existing stores are deleted. Only the primary addresses and mandatory address types are published through this message, as it is assumed that integration subsystems only require one address.

Create Stores

Stores can be setup in Merchandising through the application UI or through subscription of store information from external systems. Both methods result in this message being published when a store is created. The store creation message is published from Merchandising only after the store has been completely setup, including the mandatory addresses. Once all of these criteria are met the store details are published.

Modify Stores

Updating an existing store will trigger header modification message to notify the integrating systems. The message will contain only header level information that was changed.

Create Store Addresses

New primary addresses for the mandatory types are sent in this message for a new or an existing store. This message is sent without the store header information for existing stores. The 'key_value_1' field on this message will map to store ID.

Modify Store Addresses

Any updates made to a store's primary address information for the mandatory types will trigger an address modification message. This message is sent without the store header information if no store level changes are made at the same time. The 'key_value_1' field on this message will map to store ID. When details changes are sent, only the values that changed are included in the message.

Create Store Hours

Store hours can be set up for all seven days in a week for an existing store, or when the store is created. This message is also sent along with header information.

Modify Store Hours

Any updates to the store hours are specified in this message. This message is also sent with header information.

Table 2-101 Store Create and Update

Message Element	Included?	Notes
Store	Always	This field contains the number which uniquely identifies the store.
Store Type	Always	This field indicate whether a particular store is a franchise or company store.
Store Name	Always	This field contains the name of the store which, along with the store number, identifies the store.
Store Name10	Always	This field contains a ten-character abbreviation of the store name.
Store Name3	Always	This field contains a three-character abbreviation of the store name.
Store Add1	Optional	This field contains the first line of the address of primary address type.
Store Add2	Optional	This field contains the second line of the address of primary address type.
Store City	Optional	This field contains the name of the city that is associated with the address of primary address type.
County	Optional	This field contains county name for the location associated with the address of primary address type.
State	Optional	This field contains the state abbreviation for the address of primary address type.
Country Id	Optional	This column contains the country code where the address exists of primary address type.
Store Pin code	Optional	This field contains the zip code for the address of primary address type.
Store Class	Always	This field contains the code letter indicating the class of which the store is a member. Valid values are A through E.
Store Manager Name	Always	This field contains the name of the store manager.
Store Open Date	Always	This field contains the date on which the store opened.
Store Close Date	Optional	This field contains the date on which the store closed.
Acquired Date	Optional	This field contains the date on which the store was acquired.

Table 2-101 (Cont.) Store Create and Update

Message Element	Included?	Notes
Remodel Date	Optional	This field contains the date on which the store was last remodeled.
Fax Number	Optional	This field contains the fax number for the store.
Phone Number	Optional	This field contains the phone number for the store.
Email	Optional	This field the email address for the location.
Total Square Feet	Optional	This field contains the total square footage of the store.
Selling Square Feet	Optional	This field contains the total square footage of the stores selling area.
Linear Distance	Optional	This field contains the total merchandisable space of the location.
Stockholding Indicator	Always	This field indicates whether the store can hold stock.
Channel Id	Optional	This field contains the channel with which the store is associated.
Store Format	Optional	This field contains the number indicating the format of the store. Valid values are found on the store format table.
Mall Name	Optional	This field contains the name of the mall in which the store is located.
District	Always	This field contains the number of the district in which the store is a member.
District Name	Optional	This field contains the name of the district in which the store is a member.
Promo Zone	Optional	Not used.
Promo Description	Optional	Not used.
Transfer Zone	Optional	This field contains the transfer zone ID in which the store is located.
Description	Optional	This field contains the transfer zone description in which the store is located.
Default WH	Optional	This field contains the number of the warehouse that is the primary sourcing warehouse for the store. This value will always be a virtual warehouse.
Stop Order Days	Optional	This field contains the number of days before a store closing that the store will stop accepting orders. This column will be used when the store close date is defined.
Start Order Days	Always	This field contains the number of days before the store open date that the store will begin accepting orders.
Currency Code	Always	This field contains the currency code under which the store operates.
Language	Always	This field contains the language code to be used for the given store.

Table 2-101 (Cont.) Store Create and Update

Message Element	Included?	Notes
Integrated POS	Always	This field indicates whether Sales Audit should expect files from this store for processing. Valid values are Yes (Y) or No (N).
Original Currency Code	Always	Not used.
DUNS Number	Optional	This field contains the Dun and Bradstreet number to identify the store.
DUNS Location	Optional	This field contains the Dun and Bradstreet number to identify the location.
Address Details	Always	Child Node
Pricing Location	Optional	This field contains the location from which pricing records will be copied for a new store or will be used to determine the price for this store if not otherwise defined for an item.
Pricing Location Currency	Optional	This field contains the currency code of the location from which pricing records will be copied.
Org Unit	Optional	This field contains the organizational unit ID that this store is associated with.
Time Zone Name	Always	This field contains the text value of the time zone of the store.
Flex Attributes	Optional	Child Node
Store Name Secondary	Optional	This field contains the secondary name of the store.
Store Class Description	Optional	This field contains the description of the class of which the store is a member.
VAT Region	Optional	This field contains the ID of the tax region the store is associated with.
VAT Include Indicator	Optional	This field contains whether tax will be included in the retail prices for the store. Valid values are Yes (Y) or No (N).
Channel Name	Optional	This field contains the name of the channel.
Store Format Name	Optional	This field contains the name of the Store Format.
Sister Store	Optional	This field contains store number which will be used to relate the current store to the historical data of an existing store. This is used by Allocation.
Transfer Entity	Optional	This field contains the transfer entity the store is associated with.
Auto Receive	Always	This field contains whether receipts for the store will have receipts made automatically by Merchandising. Valid Values are Y (Yes), N (No), D (System Default). Default value should be D, which means it is driven by the system option definition.
Re Merch Indicator	Always	Not used.
Franchise Customer	Optional	Indicates which franchise customer this store is associated with.

Table 2-101 (Cont.) Store Create and Update

Message Element	Included?	Notes
Customer Order Loc Indicator	Optional	This field indicates whether the location is customer order location. If the indicator is Y then the location can be used for sourcing and fulfillment orders; else it cannot be used and inventory information will not be shared with OMS.
Gift Wrapping Indicator	Optional	This field indicates if the location will support gift wrapping.
Customer Order Ship Indicator	Optional	This field indicates if the location will support shipping to a customer for customer orders.
Store Hours Details	Always	Child Node
Online Store Indicator	Optional	This field indicates that the store is an online store.

Table 2-102 Address Create and Update

Message Element	Included?	Notes
City Id	Optional	Not used.
State Name	Optional	This field contains the state abbreviation for the address.
Country Name	Always	This field contains the country where the address exists.
Address	Always	This field contains a unique number used to distinguish between different addresses.
Address type	Always	This field indicates the type for the address. Valid values are: 01 - Business, 02 - Postal, 03 - Returns, 04 - Order, 05 - Invoice, 06 - Remittance
Primary Address Type	Always	This field indicates if this address type is the primary address type for the store. Valid values are Yes (Y) or No (N).
Primary Address Indicator	Always	This field indicates whether the address is the primary address for the address type.
Address Line 1	Optional	This field contains the first line of the address.
Address Line 2	Optional	This field contains the second line of the address.
Address Line 3	Optional	This field contains the third line of the address.
City	Always	This field contains the name of the city that is associated with the address.
State	Optional	This field contains the state abbreviation for the address.
Country	Always	This field contains the country code where the address exists.
Post	Optional	This field contains the postal code for the address.
Contact Name	Optional	This field contains the name of the contact for the supplier at this address.

Table 2-102 (Cont.) Address Create and Update

Message Element	Included?	Notes
Contact Phone	Optional	This column contains the phone number of the contact person at this address.
Contact Telex	Optional	This field contains the telex number of the store representative contact.
Contact Fax	Optional	This field contains the fax number of the contact person at this address.
Contact Email	Optional	This field contains the email address of the store representative contact.
Oracle Vendor Site Id	Optional	This field contains the Oracle Vendor Site Id.
County	Optional	This field contains the county name for the location.
Jurisdiction Code	Optional	This field contains the jurisdiction code for the country-state relationship.

Table 2-103 Store Hours Create or Update

Message Element	Included?	Notes
Store	Always	This field contains the store number.
Day Number	Always	This field contains the numeric value that indicates the day that the open and close times pertain to. Valid values are 1-Sunday, 2-Monday, 3-Tuesday, 4-Wednesday, 5-Thursday, 6-Friday, or 7-Saturday.
Store Open Time	Optional	This column contains the open time for the store for each day of the week. The time format will be in HH:MM AM or HH:MM PM.
Store Close Time	Optional	This column contains the close time for the store for each day of the week. The time format will be in HH:MM AM or HH:MM PM.

Flex Attributes

If any custom flex attributes (CFAS) for the store have been added or modified, it will trigger a header modify message. All the entity's active flex attributes from all attribute groups are published as key-value pairs based on the group set view. This CFAS object is embedded in the outbound store message.

Table 2-104 Flex Attribute

Message Element	Included?	Notes
Name	Always	The flex attribute defined by the business.
Value	Optional	The value of the flex attribute defined by the business
Value Date	Optional	The date value of the flex attribute if flex attribute is defined as a date.

Delete Stores

When a store is deleted that has already published to external systems, a delete transaction will be sent to notify external systems that the store is no longer valid. If an address is deleted for a store, this also triggers a delete transaction. The message includes the store information along with address key. There is not a separate delete for store hours – it is assumed that external systems will delete the store hours when they receive the store delete message.

Table 2-105 Store Delete

Message Element	Included?	Notes
Store	Always	This field contains the number which uniquely identifies the store.
Store Type	Always	This field indicate whether a particular store is a franchise or company store.
Stock Holding Indicator	Always	This field indicates whether the store can hold stock.
Address Detail	Optional	Child Node

Table 2-106 Address Delete

Message Element	Included?	Notes
Address	Always	This field contains the address key.

Error Handling

If any errors are encountered while publishing the message, a fatal error with status E (Error) is sent to RIB. This would bring down the RIB adapter, thus preventing any further messages from being processed until this is resolved.

Message XSD

Below are the filenames that correspond with each message type. Please consult the Oracle Retail Integration Guide for each message type for the details on the composition of each message.

Message Types	Message Type Description	XML Schema Definition (XSD)
Storecre	Store Create	StoreDesc.xsd
Storedtlcre	Store Address Create	StoreDesc.xsd
Storehrcre	Store Hours Create	StoreDesc.xsd
Storemod	Store Header Modify	StoreDesc.xsd
Storedtlmod	Store Address Modify	StoreDesc.xsd
Storehrmod	Store Hours Modify	StoreDesc.xsd
Storedel	Store Delete	StoreRef.xsd
Storedtldel	Store Address Delete	StoreRef.xsd

Transfer Publication API

This section describes the transfers publication API.

Functional Area

Transfer

Business Overview

A transfer is a movement of stock on hand from one stockholding location within the company to another. Transfers consists of header level information in which source and destination locations are specified, and detail information regarding what items and the quantity of each item is to be transferred. In some cases, transfers can also be created to pass through a finisher location to have activities performed on the items being transferred, such as ticketing, re-packing, or item transformations. Transfers that have been approved are published and will be visible to external systems. Updates are provided synchronously in a near-real time manner.

New Transfer

Creating and approving a new transfer triggers a message to be sent to notify external systems of the changes. The full details in the message are sent for the new transfer as part of the create message. It includes details such as the source and destination location, the items that are to be transferred, component ticketing information (if applicable), and the quantity. If the transfer is for a customer order (`tsf_type = 'CO'`), customer related information pulled from ORDCUST table will be included in the header message and published. A transfer with header record will have at least one detail before it can be approved.

Transfer Details

Adding transfer details like item and transfer quantity to an existing transfer triggers a detail create message to be created and published after the transfer is approved. This action will create one header modify message for the status change and each detail addition will create one detail create message. The detail create message will still contain both complete header and detail information. The details of the payload used for transfer is as below.

Updated Transfer

Any updates to an existing transfer trigger a message to be sent to notify external systems based on updates made at the transfer header level and/or the detail level. The message that will be published will either contain the header information that changed in the message for header updates such as date changes, just the detail level information for detail level changes such as quantity changes or addition of new items, or both.

Header Only Updates

When updates are made at the transfer header level such as updating dates, comment description or changing the status of the transfer, only header information is published. Information specific to items and locations are not included. Changes in header

information, unapproving a transfer, or any action that may change of status of a pre-approved transfer such as closing or reinstating a transfer trigger header level publication. Creating new and updating header level custom flex attributes (CFAS) also trigger header level update publishing.

Detail Only Updates

When updates are made at the transfer detail level without changes to the header information, only detail information is published. Detail level update includes change in quantity transferred. When a transfer has already been approved, no deletes can be made at the detail level.

Full Message Updates

In cases where the solution receiving transfers cannot support just receiving the delta changes, another option is provided that can resend the full transfer header and transfer details, along with any flex attributes configured for transfers, whenever there is a change. This message payload will contain a full snapshot of the updated transfer in cases where the system option Publish Full Objects (PUB_FULL_OBJECTS_IND) is set to Deltas and Full (Y). This is used for Oracle Retail Warehouse Management (RWMS) Cloud integration.

Table 2-107 Create or Update Transfer Header

Message Element	Required?	Notes
Transfer Number	Always	This field contains the number that uniquely identifies the transfer.
Document Type	Always	This field contains the document type. For transfers it is defaulted to T.
Physical From Location	Always	This field contains the physical location that the source location belongs to. For stores or external finishers, it will be the store ID, for a virtual warehouse or internal finisher, it will be the physical warehouse ID.
From Location Type	Always	Type of location in the source location field. Valid values are store (S), virtual warehouse (which could be an internal finisher) (W), or external finisher (E).
From Store Type	Optional	If the from location is a store, this field contains the type of store. Valid values are company store (C), franchise store (F).
From Stockholding Indicator	Optional	If the from location is a store, this field indicates if it is stockholding. Valid values are yes (Y) or no (N).
From Location	Always	This field contains the source location - either a store, external finisher, or virtual warehouse number (which could be an internal finisher) based on the from location type.
Physical To Location	Always	This field contains the physical location ID for the destination location. For stores or external finishers, it will be the store ID, for a virtual warehouse or internal finisher, it will be the physical warehouse ID.
To Location Type	Always	This field contains the type of location the transfer will be shipped to. Valid values are store (S), warehouse (which could be an internal finisher) (W), or external finisher (E).

Table 2-107 (Cont.) Create or Update Transfer Header

Message Element	Required?	Notes
To Store Type	Optional	If the to location is a store, this field contains the type of store. Valid values are company store (C), franchise store (F).
To Stockholding Indicator	Optional	If the to location is a store, this field indicates if it is stockholding. Valid values are yes (Y) or no (N).
To Location	Always	This field contains the destination location - either a store, external finisher, or virtual warehouse number (which could be an internal finisher) based on the to location type.
Transfer Type	Always	This field holds the code for the type of the transfer. Valid values are: <ul style="list-style-type: none"> • SR - Store Requisition • CO - Customer Order • RV - RTV • CF - Confirmation • NS - Non-Salable • AD - Administrative • MR - Manual Requisitions • PL - PO-Linked Transfer. • IC -Intercompany
Pick Not Before Date	Always	This field holds the earliest date on which the transfer may be picked. This field is populated with the date the transfer was approved.
Pick Not After Date	Optional	The last date on which the order may be picked. This is calculated as transfer approval date + days from code_detail where code_type = 'DEFT' and code = 'DATE'
Order Type	Always	This is the order type associated with transfers. If the transfer type is customer order, then the order type is set to MANUAL, else it is defaulted from the default order type set in the Merchandising.
Priority	Optional	Not used in Merchandising.
Break By Distro	Always	This field holds the delivery type for the customer order. If it is shipped direct to the customer, this will be yes (Y), otherwise N (no).
Delivery Date	Optional	This field holds the earliest date that the transfer can be delivered to the store.
Customer Name	Optional	This field contains the name of the customer.
Delivery Address 1	Optional	This column contains the first line of the delivery address of the customer for customer order transfers. This field will be set to NULL if the Retain Customer Information system option is unchecked (N).
Delivery Address 2	Optional	This column contains the second line of the delivery address of the customer, for customer order transfers. This field will be set to NULL if the Retain Customer Information system option is unchecked (N).

Table 2-107 (Cont.) Create or Update Transfer Header

Message Element	Required?	Notes
Delivery City	Optional	This column contains the city of the delivery address, for customer order transfers. This field will be set to NULL if the Retain Customer Information system option is unchecked (N).
Delivery State	Optional	This column contains the state code of the delivery address, for customer order transfers. This field will be set to NULL if the Retain Customer Information system option is unchecked (N).
Delivery Postal	Optional	This column contains the postal code of the delivery address, for customer order transfers. This field will be set to NULL if the Retain Customer Information system option is unchecked (N).
Delivery Country	Optional	This column contains the country code of the delivery address, for customer order transfers. This field will be set to NULL if the Retain Customer Information system option is unchecked (N).
Message	Optional	This column contains any comments associated with the transfer.
Parent Transfer Number	Optional	This column contains the transfer number for the first leg (from location to finisher) for transfers with finishing.
Expected WH Date	Optional	This column contains the date the inventory is expected to arrive at the warehouse.
Approval ID	Optional	This column contains the ID of the user who approved the transfer.
Approval Date	Optional	This column contains the date the transfer was approved
From Location Transfer Entity	Optional	This column contains the transfer entity of the source location.
To Location Transfer Entity	Optional	This column contains the transfer entity of the destination location.
Inventory Type	Optional	This column will indicate whether the transfer is for available inventory or unavailable inventory. Valid Values are: <ul style="list-style-type: none"> • A: Available • U: Unavailable
Transfer Status	Optional	This field indicates the status of the transfer. Valid values are: <ul style="list-style-type: none"> • I - Input • B - Submitted • A - Approved • P - Picked • S - Shipped • N - In Progress • C - Closed • D - Deleted
Not After Date	Optional	This field contains the last day delivery of the transfer will be accepted.

Table 2-107 (Cont.) Create or Update Transfer Header

Message Element	Required?	Notes
Context Type	Optional	This field indicate the transfer reason or category. Valid values are found in code type CNTX. Note: context type Repairing (REPAIR) has specific logic tied to it and must be associated with a return to warehouse or a transfer using an external finisher.
Context Value	Optional	This field holds value relating to the context type, for example Promotion Number for context type Promotion.
Delivery Slot ID	Optional	This field indicates the delivery slot that will be used for the transfer. The valid values are defined in the DELIVERY_SLOT table.
Delivery Slot Description	Optional	This field contains the delivery slot description.
Customer Order Number	Optional	This field contains the customer order number associated with the transfer.
Fulfill Order Number	Optional	This field contains the fulfillment order number for the transfer.
Carrier Code	Optional	This field indicates the carrier the customer order is to be shipped with, if specified on the order.
Carrier Service Code	Optional	This field indicates the code for the service level for the carrier (e.g., overnight shipping) for a customer order shipment. Valid values are found in code type CSVC.
Consumer Delivery Date	Optional	This field indicates the desired date the delivery is required by the customer.
Consumer Delivery Time	Optional	This field indicates the desired time, the delivery is required by the customer.
Deliver First Name	Optional	This column contains the first name for the delivery address on the order, for a customer order transfer. This field will be set to NULL if the Retain Customer Information system option is unchecked (N).
Deliver Phonetic First	Optional	This column contains the phonetic first name for the delivery address on the order, for a customer order transfer. This field will be set to NULL if the Retain Customer Information system option is unchecked (N).
Deliver Last Name	Optional	This column contains the phonetic last name for the delivery address on the order, for a customer order transfer. This field will be set to NULL if the Retain Customer Information system option is unchecked (N).
Deliver Phonetic Last	Optional	This column contains the phonetic last name for the delivery address on the order, for a customer order transfer. This field will be set to NULL if the Retain Customer Information system option is unchecked (N).

Table 2-107 (Cont.) Create or Update Transfer Header

Message Element	Required?	Notes
Deliver Preferred Name	Optional	This column contains the preferred name for the delivery address on the order, for a customer order transfer. This field will be set to NULL if the Retain Customer Information system option is unchecked (N).
Deliver Company Name	Optional	This column contains the company name for the delivery address on the order, for a customer order transfer. This field will be set to NULL if the Retain Customer Information system option is unchecked (N).
Deliver Address 3	Optional	This column contains the third line of the delivery address of the customer, for a customer order transfer. This field will be set to NULL if the Retain Customer Information system option is unchecked (N).
Deliver County	Optional	This column contains the county of the delivery address, for a customer order transfer. This field will be set to NULL if the Retain Customer Information system option is unchecked (N).
Deliver Phone	Optional	This column contains the delivery phone number, for a customer order transfer. This field will be set to NULL if the Retain Customer Information system option is unchecked (N).
Bill First Name	Optional	This column contains the first name for the billing address, for a customer order transfer. This field will be set to NULL if the Retain Customer Information system option is unchecked (N).
Bill Phonetic First	Optional	This column contains the phonetic first name for the billing address on the order, for a customer order transfer. This field will be set to NULL if the Retain Customer Information system option is unchecked (N).
Bill Last Name	Optional	This column contains the last name for the billing address on the order, for a customer order transfer. This field will be set to NULL if the Retain Customer Information system option is unchecked (N).
Bill Phonetic Last	Optional	This column contains the phonetic last name for the billing address on the order, for a customer order transfer. This field will be set to NULL if the Retain Customer Information system option is unchecked (N).
Bill Preferred Name	Optional	This column contains the preferred name for the billing address on the order, for a customer order transfer. This field will be set to NULL if the Retain Customer Information system option is unchecked (N).
Bill Company Name	Optional	This column contains the company name for the billing address on the order, for a customer order transfer. This field will be set to NULL if the Retain Customer Information system option is unchecked (N).

Table 2-107 (Cont.) Create or Update Transfer Header

Message Element	Required?	Notes
Bill Address 1	Optional	This column contains the first line of the billing address of the customer, for a customer order transfer. This field will be set to NULL if the Retain Customer Information system option is unchecked (N).
Bill Address 2	Optional	This column contains the second line of the billing address of the customer, for a customer order transfer. This field will be set to NULL if the Retain Customer Information system option is unchecked (N).
Bill Address 3	Optional	This column contains the third line of the billing address of the customer, for a customer order transfer. This field will be set to NULL if the Retain Customer Information system option is unchecked (N).
Bill County	Optional	This column contains the county of the billing address, for a customer order transfer. This field will be set to NULL if the Retain Customer Information system option is unchecked (N).
Bill City	Optional	This column contains the city of the billing address, for a customer order transfer. This field will be set to NULL if the Retain Customer Information system option is unchecked (N).
Bill Country	Optional	This column contains the country code of the billing address, for a customer order transfer. This field will be set to NULL if the Retain Customer Information system option is unchecked (N).
Bill Post	Optional	This column contains the postal code of the billing address, for a customer order transfer. This field will be set to NULL if the Retain Customer Information system option is unchecked (N).
Bill State	Optional	This column contains the state code of the billing address, for a customer order transfer. This field will be set to NULL if the Retain Customer Information system option is unchecked (N).
Bill Phone	Optional	This column contains the billing phone number, for a customer order transfer. This field will be set to NULL if the Retain Customer Information system option is unchecked (N).
Partial Delivery Indicator	Optional	This column indicates if a customer order transfer can be picked and shipped partially or if it should be shipped only when complete. Valid values are Y or N.
Consumer Direct	Optional	This column indicates the order as being consumer direct (Y) or not (N). An order is consumer direct if it is a customer order.

Table 2-108 Create or Update Transfer Detail

Message Element	Required?	Notes
Item	Always	This field contains unique alphanumeric value that identifies the item.
Transfer Quantity	Always	This field contains the total quantity of the item reserved at the from location for this transfer. For customer order transfers the values are fetched from the customer order tables.
Price	Optional	This field contains the unit retail price in the selling unit of measure for the item/ to location combination
Selling UOM	Optional	This field contains the selling unit of measure of the item.
Priority	Optional	Not used in Merchandising.
Expedite Flag	Optional	This field will be set to yes (Y) if the freight type is Expedite else it is set to no (N).
Store Order Multiple	Optional	This field contains the multiple in which the item needs to be shipped from a warehouse to the location.
Transfer PO Link Number	Optional	If this transfer was generated out of the replenishment process using the WH/Cross Link stock category, then this field contains a reference number to link the item on the transfer to the purchase order that will fulfill part of the need.
Ticket Type	Optional	This field contains the ticket type ID associated with the item.
Inventory Status	Optional	This field contains the inventory status that should be used as the source of inventory on the transfer. If the transfer is from available inventory this field will contain -1.
Transaction UOM	Optional	This field contains the original transaction unit of measure. It is only used for customer direct orders coming from an order management system.
Item Line Number	Optional	This field contains the detail item line number for customer order transfers.
Comments	Optional	This field contains any comments associated with the transfer line item for a customer order transfer.

Table 2-109 Pack Component Detail

Message Element	Required?	Notes
Component Item	Conditional	This field contains an alphanumeric value that identifies the item within the pack.
Component item Price	Conditional	This field contains the unit retail price in the selling unit of measure for the component item/location combination
Component Selling UOM	Conditional	This field contains the selling unit of measure for the component item.

Flex Attributes

If any custom flex attributes (CFAS) for the transfer have been added or modified, it will trigger a transfer header message, as described above. The node of the integration that supports this will contain the name of the attribute as it is defined in the group set level view, the value of the custom attribute. If it is a date attribute, the date value is in a separate field. You can define group set views at the header level (TSFHEAD) levels. Flex attributes can only be added to or updated on a transfer; they cannot be deleted.

Table 2-110 Custom Flex Attribute Detail

Message Element	Required?	Notes
Name	Always	The flex attribute defined by the business.
Value	Optional	The value of the flex attribute defined by the business
Value Date	Optional	The date value of the flex attribute if flex attribute is defined as a date.

Deleted Transfer

A transfer can only be deleted when it is still in approved status or when it has been closed. Only a detail that has not been shipped may be deleted, and it cannot be deleted if it is currently being worked on by an external system. Deleting a transfer removes it from the system. External systems are notified by a published delete message that contains the number of the transfer to be deleted.

Table 2-111 Header Information

Message Element	Required?	Notes
Transfer Number	Always	This field contains the number that uniquely identifies the transfer.
Documentation Type	Always	This field contains the document type. For transfers it is defaulted to T.
Physical From Location	Optional	This field contains the physical location that the source location belongs to. For stores or external finishers, it will be the store ID, for a virtual warehouse or internal finisher, it will be the physical warehouse ID.
From Location	Optional	This field contains the source location - either a store, external finisher, or virtual warehouse number (which could be an internal finisher) based on the from location type.
From Location Type	Optional	Type of location in the source location field. Valid values are store (S), virtual warehouse (which could be an internal finisher) (W), or external finisher (E).
From Store Type	Optional	If the from location is a store, this field contains the type of store. Valid values are company store (C), franchise store (F).

Table 2-111 (Cont.) Header Information

Message Element	Required?	Notes
From Stockholding Indicator	Optional	If the from location is a store, this field indicates if it is stockholding. Valid values are yes (Y) or no (N).
Physical To Location	Always	This field contains the physical location ID for the destination location. For stores or external finishers, it will be the store ID, for a virtual warehouse or internal finisher, it will be the physical warehouse ID.
To Location	Always	This field contains the destination location - either a store, external finisher, or virtual warehouse number (which could be an internal finisher) based on the to location type.
To Location Type	Optional	This field contains the type of location the transfer will be shipped to. Valid values are store (S), warehouse (which could be an internal finisher) (W), or external finisher (E).
To Store Type	Optional	If the to location is a store, this field contains the type of store. Valid values are company store (C), franchise store (F).
To Stockholding Indicator	Optional	If the to location is a store, this field indicates if it is stockholding. Valid values are yes (Y) or no (N).
Transfer Parent Number	Optional	This column contains the transfer number for the first leg (from location to finisher) of a transfer with finishing.

Table 2-112 Detail Information

Message Element	Required?	Notes
Item	Always	This field contains unique alphanumeric value that identifies the item being deleted from the transfer.

Transfer Unapprove

When the transfer is moved back to input status from approved, the RIB processes the transfer as a delete message when publishing it to external systems. When the transfer is re-approved, the transfer is processed as a new transfer for publishing.

Error Handling

If any errors are encountered while publishing the message, a fatal error with status E (Error) is sent to RIB. This would bring down the RIB adapter thus preventing any further messages from being processed until this is resolved.

Message XSD

Below are the filenames that correspond with each message type. Please consult the *Oracle Retail Integration Guide* for each message type for the details on the composition of each message.

Message Types	Message Type Description	XML Schema Definition (XSD)
TransferCre	Transfer Create Message	TsfDesc.xsd
TransferDtlCre	Transfer Detail Create Message	TsfDesc.xsd
TransferHdrMod	Transfer Modify Message	TsfDesc.xsd
TransferDtlMod	Transfer Detail Modify Message	TsfDesc.xsd
TransferDel	Transfer Delete Message	TsfRef.xsd
TransferDtlDel	Transfer Detail Delete Message	TsfRef.xsd
Transferfulrep	Transfer Full Replacement Message	TsfDesc.xsd
TransferUnapp	Transfer Unapprove	TsfRef.xsd

UDA Publication API

This section describes the UDA publication API.

Functional Area

Foundation Data

Business Overview

Merchandising publishes details about user-defined attributes (UDAs) to the Oracle Retail Integration Bus (RIB). UDAs provide a method for defining attributes and associating the attributes with specific items, items on an item list, or items in a specific department, class, or subclass. UDAs are useful for information and reporting purposes. Unlike traits or indicators, UDAs are not interfaced with external systems. UDAs do not have any programming logic associated with them. UDA messages are specific to basic UDA identifiers and values defined in Merchandising. The UDAs can be displayed in one or more of three formats: Dates, Freeform Text, or a List of Values (LOV).

New UDAs

Creating a new UDAs triggers a message to be sent via the RIB to notify external systems. The full details are sent for the new UDA as part of the create message: the UDA ID, description, display type, data type, data length and single value indicator.

Table 2-113 UDA

Message Element	Included?	Notes
UDA ID	Always	This field contains the unique identifier of the user defined attribute.
UDA Description	Always	This field contains a description of the user defined attribute.
Module	Always	This field indicates which system module the UDA belongs to. This will always be 'ITEM'.

Table 2-113 (Cont.) UDA

Message Element	Included?	Notes
Display Type	Always	This field contains a code to indicate how the UDA will be displayed to the user. Valid values are in the UDIS code type.
Data Type	Optional	This field indicates what value types are valid for this UDA. Valid values are: <ul style="list-style-type: none"> • NUM • ALPHA • DATE
Data Length	Optional	This field indicates the length of the field.
Single Value Indicator	Optional	This field indicates whether the UDA is limited to having at most one value. Valid values are Yes (Y) and No (N).
Host Indicator	Optional	
Carton Group	Optional	
Combinability	Optional	

Updated UDAs

When an existing UDA is updated, an update message is triggered to provide the details of the update via the RIB. The update message, like create, will contain the full details in the message for all fields in the message.

Deleted UDAs

When an existing UDA is deleted, this will also trigger a delete transaction to be sent via the RIB to notify external systems that this UDA is no longer valid. The delete message will include only the UDA ID, UDA value and description of the UDA being deleted.

Table 2-114 UDA

Message Element	Included?	Notes
UDA ID	Always	This field contains the unique identifier of the user defined attribute being deleted.

New UDA Details

Creating a new UDA details triggers a message to be sent via the RIB to notify external systems. The UDA value and description are sent for the existing UDA as part of the create message.

Table 2-115 UDA Values

Message Element	Included?	Notes
UDA ID	Always	This field contains the unique identifier of the user defined attribute.

Table 2-115 (Cont.) UDA Values

Message Element	Included?	Notes
UDA Value	Always	This field contains a unique number identifying the User Defined Attribute value for the UDA. A UDA can have multiple values. For example, Color can be a UDA and it can have different values like Green, Red, Blue, etc.
UDA Value Description	Always	This field contains a description of the UDA.

Updated UDAs

When an existing UDA detail is updated, an update message is triggered to provide the details of the update via the RIB. The update message, will contain the UDA value and description in the message.

Deleted UDAs

When an existing UDA detail is deleted, this will also trigger a delete transaction to be sent via the RIB to notify external systems that this UDA detail is no longer valid. The delete message will include only the UDA ID and UDA value being deleted.

Table 2-116 UDA Values

Message Element	Included?	Notes
UDA ID	Always	This field contains the unique identifier of the user defined attribute being deleted.
UDA Value	Always	This field contains a unique number identifying the User Defined Attribute value for the UDA. A UDA can have multiple values. For example, Color can be a UDA and it can have different values like Green, Red, Blue, etc.

Error Handling

When the publication encounters a non-fatal error, messages continue to be processed.

For the message where the error was encountered, a status of Hospital (H) is sent to the RIB and the status of the message in the queue is set to H. In case the error is a fatal error, a status of Error (E) is sent to the RIB and the next message in the queue is not retrieved until the error is resolved

Message XSD

Below are the filenames that correspond with each message type. Please consult the Oracle Retail Integration Guide for each message type for the details on the composition of each message.

Message Types	Message Type Description	XML Schema Definition (XML)
UdaHdrCre	UDA Header Create	UDADesc.xsd
UDAValCre	UDA Detail Create	UDAValDesc.xsd
UDAHdrMod	UDA Header Modify	UDADesc.xsd
UDAValMod	UDA Detail Modify	UDAValDesc.xsd
UDAHdrDel	UDA Header Delete	UDAREf.xsd
UDAValDel	UDA Detail Delete	UDAValRef.xsd

Vendor Publication API

This section describes the vendor publication API.

Functional Area

Suppliers

Business Overview

This API publishes suppliers and supplier address information through the RIB to external systems. This information is further subscribed by integrating systems like Oracle Retail Store Inventory and Operations Cloud Service (SIOCS). Supplier information is published when new supplier sites are created, updates are made to existing sites, or existing supplier sites are deleted. Similarly, addresses are published when they are added, modified, or deleted. The address types that are published as part of this message are

- Returns (3)
- Order (4)
- Invoice (5)

Only supplier site level information is published. The supplier level information will not be published.

Vendor Creation

Supplier sites can be set up in Merchandising through the application UI, or through subscription of vendor information from external systems. Both methods result in this message being published when a supplier is created. The vendor creation message is published from Merchandising only after the supplier site has been completely set up, including the mandatory addresses and its org unit. Once all these criteria are met for a valid create message, the messages will be combined and sent to RIB. Both active and inactive supplier sites are published.

Vendor Header Modification

Updating an existing supplier site will trigger a header modification message to notify the integrating systems. This message will contain only the header level information that changed.

Table 2-117 Vendor Create and Update

Message Element	Included?	Notes
Vendor Header	Always	Child node
Vendor Address	Optional	Child node
Vendor Org Unit	Optional	Child node

Table 2-118 Vendor Header Create and Update

Message Element	Included?	Notes
Supplier	Always	Unique identifying number for a supplier site within the system.
Supplier Name	Always	This field contains the supplier site name.
Supplier Name Secondary	Optional	This type can hold secondary name for the supplier site with a max length of 240 characters.
Contact Name	Always	This field contains the name of the supplier representative contact for this site.
Contact Phone	Always	This field contains a telephone number for the supplier's representative contact.
Contact Fax	Optional	This field contains a fax number for the supplier's representative contact.
Contact Pager	Optional	This field contains a pager number for the supplier's representative contact.
Supplier Status	Always	This field contains the status of the supplier site. Valid values are: <ul style="list-style-type: none"> • A - Active • I - Inactive
QC Ind	Always	This field determines whether orders from this supplier will require quality control.
QC Percentage	Optional	This field contains the percentage of items per receipt that will be marked for quality checking.
QC Frequency	Optional	This field contains the frequency for which items per receipt will be marked for quality checking.
VC Ind	Always	This field determines whether orders from this supplier will require vendor control.
VC Percentage	Optional	This field contains percentage of items per receipt that will be marked for vendor checking.
VC Frequency	Optional	This field contains the frequency for which items per receipt that will be marked for vendor checking.
Currency Code	Always	This field contains code identifying the currency the supplier site uses for business transactions.
Language	Optional	This field contains the supplier's preferred language.
Terms	Always	This field contains an indicator identifying the purchase terms that will default when an order is created for the supplier site. These terms specify when payment is due and if any discounts exist for early payment.

Table 2-118 (Cont.) Vendor Header Create and Update

Message Element	Included?	Notes
Freight Terms	Always	This field contains code indicating what freight terms will default when an order is created for the supplier site.
Return Allow Indicator	Always	This field indicates whether the supplier site will accept returns. Valid values are Yes (Y) or No (N).
Return Authorization Required	Always	This field indicates if returns must be accompanied by an authorization number when sent back to the vendor.
Returns Minimum Amount	Optional	This field contains a value if the supplier site requires a minimum merchandise value to be returned to accept the return. Returns of less than this amount will not be processed by the system. This field is stored in the supplier's currency.
Return Courier	Optional	This field contains the name of the courier that should be used for returns to the supplier site.
Handling Percentage	Optional	This field contains the default percent to be multiplied by the return's total cost to determine the handling cost for the return.
EDI PO Ind	Always	This field indicates whether purchase orders will be sent to the supplier via EDI.
EDI PO Change	Always	This field indicates whether purchase order changes will be sent to the supplier via EDI.
EDI PO Confirmation	Always	This field indicates whether acknowledgements of purchase orders will be sent from the supplier via EDI.
EDI ASN	Always	This field indicates whether the supplier will send Advance Shipment Notifications electronically.
EDI Sales Report Frequency	Optional	This field contains the EDI sales report frequency for the supplier. Valid values are weekly (W) or daily (D).
EDI Supplier Availability Indicator	Always	This field indicates whether the supplier will send availability via EDI.
EDI Contract Indicator	Always	This field indicates whether the supplier site supports contract ordering sent via EDI.
EDI Invoice Indicator	Always	This field indicates whether invoices, debit memos, and credit note requests will be sent to/from the supplier via EDI.
Cost Change Percentage Variance	Optional	This field contains a percent that determines whether a cost change can be auto approve via induction. If the cost change falls within these boundaries, it will be approved when uploaded.
Cost Change Amount Variance	Optional	This field contains an amount (in supplier currency) that determines whether a cost change can be auto approve via induction. If the cost change falls within these boundaries, it will be approved when uploaded.

Table 2-118 (Cont.) Vendor Header Create and Update

Message Element	Included?	Notes
Replenishment Approval Indicator	Always	This field indicates whether contract orders created via replenishment should be created in Approved status.
Ship Method	Optional	This field contains the default method used to ship the items on the purchase order from the supplier site. Valid values are held in code type SHPM.
Payment Method	Optional	This field indicates the default method for how purchase orders for this site will be paid. Valid options are: <ul style="list-style-type: none"> • LC - Letter of Credit • WT - Wire Transfer • OA - Open Account
Contact Telex	Optional	This field contains a telex number for the supplier's representative contact.
Contact Email	Optional	This field contains an email address for the supplier's representative contact.
Settlement Code	Always	This field indicates which payment process method is used for the supplier. Valid values are N/A (N) or Evaluated Receipts Settlement (E).
Pre-Mark Indicator	Always	This field indicates whether the supplier site supports pre-marking containers for cross dock orders.
Auto Approved Invoice Indicator	Always	This field indicates whether the supplier's invoices can be automatically approved for payment.
Debit Memo Code	Optional	This field indicates when a debit memo will be sent to the supplier site to resolve a discrepancy. Valid values are: <ul style="list-style-type: none"> • Y - if debit memos are always to be sent • L - if debit memos are used only if a credit note is not sent by the invoice due date • N - if debit memos are never sent
Freight Charge Indicator	Always	This field indicates whether a supplier site can charge freight costs.
Auto Approve Debit Memo Indicator	Always	This field indicates whether debit memos sent to the supplier site can be automatically approved on creation.
Inventory Management Level	Optional	This field indicates the level for managing supplier inventory information. Valid values are: supplier (S), supplier/location (L), supplier/department (D), or supplier/department/location (A).
Backorder Indicator	Always	This field indicates if backorders or partial shipments will be accepted.
VAT Region	Optional	This field contains the unique identifying number for the VAT region applicable for this site.
Prepay Invoice Indicator	Always	This field indicates whether all invoices for the supplier can be pre-paid.

Table 2-118 (Cont.) Vendor Header Create and Update

Message Element	Included?	Notes
Service Performed Required Indicator	Always	This field indicates if the supplier's services must be confirmed as performed before paying an invoice from that supplier site.
Invoice Payment Location	Optional	This field indicates where invoices from this supplier site are paid - at the store (S) or centrally through corporate accounting (C).
Invoice Received Location	Optional	This field indicates where invoices from this supplier site are received - at the store (S) or centrally through corporate accounting (C).
Invoice At	Optional	This field indicates if the supplier site invoice lists items at gross cost (G) or net cost (N).
Delivery Policy	Always	This field contains the default delivery policy of the supplier site. Valid values come from the DLVY code type.
Comments	Optional	This field contains any miscellaneous comments associated with the supplier.
Default Item Lead Time	Optional	This field holds the default lead time for the supplier site. The lead time is the time the supplier needs between receiving an order and having the order ready to ship. This value will be defaulted to item/supplier relationships.
DUNS Number	Optional	The Dun and Bradstreet number of the supplier.
DUNS Location	Optional	The Dun and Bradstreet number of the location of the supplier.
Bracket Costing Indicator	Always	This field will determine if the supplier site supports bracket costing pricing structures.
VMI Order Status	Optional	This field determines the status in which any inbound POs from this supplier will be created. A NULL value indicates that the supplier is not a VMI supplier.
End Active Date	No	Always null
DSD Supplier Indicator	Always	This field specifies whether the vendor supports DSD ordering, where the supplier replenishes the store directly, creating the PO and receipt at the same time.
Supplier Quantity Level	Always	This field indicates the supplier site order quantity level. Valid values are cases (CA) or eaches (EA).
Supplier Parent	Optional	This is the supplier number for the supplier sites.
Store Delivery Discrepancy	No	Always null
Final Destination Indicator	Always	This field indicates whether the supplier site can ship to final destination or not. Valid values are Yes (Y) or No (N).
External Reference Indicator	Optional	This column holds the ID for the supplier used in the external financial system.

Table 2-118 (Cont.) Vendor Header Create and Update

Message Element	Included?	Notes
Deal Upload Status	No	This field indicates the status in which the deal will be created when uploaded. The valid values are: <ul style="list-style-type: none"> • W - Worksheet • S - Submitted • A - Approved
Tax ID	No	This field contains the unique tax identification number of the supplier site.
Custom Flex Attributes	Optional	Child Node

Vendor Address Creation

Adding new addresses of the types sent in this message for an existing supplier site will trigger an address detail creation message. This message is sent without the vendor header information. The 'key_value_1' field on this message will map to the supplier site.

Adding a new org unit to the supplier site will trigger an org unit create message. This message is also sent along with the header information.

Vendor Address Modification

Any updates made to the supplier site's address information of the types specified above will trigger an address modification message. This message is sent without the vendor header information. The 'key_value_1' field on this message will map to the supplier site. When detail changes are sent, only the values that changed are included in the message.

Table 2-119 Vendor Address Create or Update

Message Element	Included?	Notes
Module	Always	This field indicates the data type that the address is attached to. In this case, it will always be 'SUPP'.
Key Value 1	Always	This field holds the ID the address is attached to. In this case, it will be the supplier number.
Key Value 2	No	This is not used.
Sequence Number	Always	Number indicating the sequence that addresses within the same type were entered.
Address Type	Always	This field contains the address type. For suppliers, the following address types are sent: Returns (3) Order (4) Invoice (5)
Primary Address Indicator	Always	This field indicates whether the address is the primary address for the address type.
Address 1	Always	This field contains the first line of the address.
Address 2	Optional	This field contains the second line of the address.

Table 2-119 (Cont.) Vendor Address Create or Update

Message Element	Included?	Notes
Address 3	Optional	This field contains the third line of the address.
City	Always	This field contains the name of the city that is associated with the address.
State	Optional	This field contains the state abbreviation for the address.
Country ID	Always	This field contains the country where the address exists.
Jurisdiction Code	Optional	This field contains the ID associated to the tax jurisdiction of the country-state relationship.
Post	Optional	This field contains the zip code for the address.
Contact Name	Optional	This field contains the name of the contact for the supplier at this address.
Contact Phone	Optional	This field contains the phone number of the contact person at this address.
Contact Telex	Optional	This field contains the telex number of the contact person at this address.
Contact Fax	Optional	This field contains the fax number of the contact person at this address.
Contact Email	Optional	This field contains the email address of the supplier site's contact person.
Custom Flex Attributes	Optional	Child Node

Table 2-120 Vendor Org Unit Create or Update

Message Element	Included?	Notes
Org Unit ID	Always	This field contains org unit ID added or updated for the supplier site.
Primary Pay Site Indicator	Always	This field contains the primary pay site indicator.

Vendor Detail Deletion

Vendor addresses can be deleted only if they are not mandatory, or if they are mandatory but are not marked as the primary address for that address type. The deletion will trigger an address delete message.

Table 2-121 Vendor Address Delete

Message Element	Included?	Notes
Supplier	Always	This field contains the unique identifying number for a supplier within the system. The user determines this number when a new supplier is first added to the system.
Sequence Number	Always	This field contains the number indicating the sequence that addresses within the same type were entered.

Table 2-121 (Cont.) Vendor Address Delete

Message Element	Included?	Notes
Address Type	Always	This field contains the address type. Valid values (e.g. 01 - Business, 02 - Postal, etc.) are on the add_type table.

Deletion of an org unit will be allowed if there are no open purchase orders for the supplier site. This will trigger an org unit delete message.

Table 2-122 Vendor Org Unit Delete

Message Element	Included?	Notes
Org Unit ID	Always	This field contains org unit ID being removed for the supplier site.

Flex Attributes

If any custom flex attributes (CFAS) for the supplier have been added or modified, it will trigger a header modify message. All the entity's active flex attributes from all attribute groups are published as key-value pairs based on the group set view. This CFAS object is embedded in the outbound Vendor message.

Similarly, if any CFAS for the supplier address has been added or modified, it will trigger a vendor address modify message. All the entity's active flex attributes from all attribute groups are published as key-value pairs based on the group set view. This CFAS object is embedded in the outbound Vendor Address message.

Flex attributes can only be added to or updated; they cannot be deleted.

Table 2-123 Flex Attributes

Message Element	Required?	Notes
Name	Always	Holds the attribute name.
Value	Optional	Holds the value of the attribute for number and character type attributes
Value Date	Optional	Holds the date for date type attributes.

Full Message Updates

In cases where the integrating system is unable to receive only the changes, another option is provided that can resend the full supplier details whenever there is a change, in addition to the deltas. This will be published when the system option Publish Full Objects (PUB_FULL_OBJECTS_IND) is set to Deltas and Full (Y). The full message updates, like the vendor create message, will contain the full details in the message (that is, vendor header, its flex attributes, vendor address, its flex attributes and org unit information). See above for the format of a vendor create message. This is used for Oracle WMS Cloud integration.

Error Handling

If any errors are encountered while publishing the message, a fatal error with status E (Error) is sent to RIB. This would bring down the RIB adapter, thus preventing any further messages from being processed until this is resolved

Message XSD

Below are the filenames that correspond with each message type. Please consult the Oracle Retail Integration Guide for each message type for the details on the composition of each message.

Message Types	Message Type Description	XML Schema Definition (XSD)
VendorCre	Vendor Create	VendorDesc.xsd
VendorAddrCre	Vendor Address Create	VendorAddrDesc.xsd
VendorOUCre	Vendor Org Unit Create	VendorOUDesc.xsd
VendorHdrMod	Vendor Header Modify	VendorHdrDesc.xsd
VendorAddrMod	Vendor Address Modify	VendorAddrDesc.xsd
VendorDel	Vendor Delete	VendorRef.xsd
VendorAddrDel	Vendor Address Delete	VendorAddrRef.xsd
VendorOUDel	Vendor Org Unit Delete	VendorOURef.xsd
VendorFulRep	Full message	VendorDesc.xsd

Warehouse Publication API

This section describes the warehouse publication API.

Functional Area

Foundation Data

Business Overview

This API facilitates the transmission of warehouse data to external integrating applications. Merchandising publishes information for both virtual and physical warehouses. Those applications that do not have virtual location logic, such as Store Inventory and Operations Cloud Service (SIOCS), depend upon the RIB to transform the warehouse messages for physical warehouses only.

These messages are triggered by the insert, update, and delete of warehouses and warehouse addresses. Only the primary address of the primary address type is included in this message.

Warehouse Creation

Creating a new warehouse triggers a message to be sent to notify external systems. The warehouse details along with the addresses are sent for the new warehouse as part of

the create message. Physical warehouses are published only after the primary addresses have been added. Address information is sent only for the physical warehouses.

Warehouse Modification

Updating an existing warehouse will trigger a header modification message to notify the integrating systems. This message will contain only the header level information that changed.

Table 2-124 Warehouse Create or Update

Message Element	Included?	Notes
Warehouse	Always	The number which uniquely identifies the warehouse. This could be either a physical or virtual warehouse.
Warehouse Name	Always	This field contains the warehouse name. This is published for both physical and virtual warehouse
Address Line 1	Optional	Not used
Address Line 2	Optional	Not used
City	Optional	Not used
County	Optional	Not used
State	Optional	Not used
Country	Optional	Not used
Postal Code	Optional	Not used
E-mail	Optional	This field holds the email address for the location. This is published for only the physical warehouse
Stockholding Indicator	Optional	Indicates if the warehouse is a stock holding location. It will be N for a physical warehouse and Y for a virtual warehouse.
Channel ID	Optional	The channel for which the virtual warehouse has been assigned. This is published for the virtual warehouse
Currency Code	Optional	The code for the currency that the warehouse uses. This is published for the physical and virtual warehouse
DUNS Number	Optional	This field holds the Dun and Bradstreet number to identify the physical warehouse. This is published for the physical warehouse
DUNS Location	Optional	This field holds the Dun and Bradstreet number to identify the physical warehouse. This is published for the physical warehouse
Physical Warehouse	Optional	The number of the physical warehouse corresponding to this virtual warehouse. For physical warehouses, this field will be null.
Break Pack Indicator	Optional	Indicates whether the physical warehouse can distribute less than the supplier case quantity. Valid values are Y or N. This is published for both physical and virtual warehouse

Table 2-124 (Cont.) Warehouse Create or Update

Message Element	Included?	Notes
Redistribution Warehouse Indicator	Optional	A re-distribution warehouse is used to flag purchase orders for review closer to the time of receipt. If this is set to yes (Y) for a physical warehouse, then it indicates that this is a “dummy” location for a purchase order and that the order will have its order quantities redistributed to actual receiving locations close to expected receipt. Valid values are yes (Y) and no (N). This is published for both physical and virtual warehouse
Delivery Policy	Optional	The delivery policy of the physical warehouse. Next Day (NEXT) indicates that if a location is closed, the warehouse will deliver on the next day. Next Valid Delivery Day (NDD) indicates that the warehouse will wait until the next scheduled delivery day. This is published for both physical and virtual warehouse
Contact Person	Optional	Not used
Fax Number	Optional	Not used
Phone Number	Optional	Not used
Default Route	Optional	Not used
Default Carrier Code	Optional	Not used
Default Service Code	Optional	Not used
Expedited Route	Optional	Not used
Expedited Carrier Code	Optional	Not used
Expedited Service Code	Optional	Not used
BOL Upload Type	Optional	Not used
BOL Print Type	Optional	Not used
Lead Time	Optional	Not used. Defaulted to 0.
Distance to Destination	Optional	Not used. Defaulted to 0.
Drop Trailers Accepted	Optional	Not used
Receiving Dock Available	Optional	Not used
Container Type	Optional	Not used
MLD Default Route	Optional	Not used
Unit Pick Container Type	Optional	Not used
Destination Sequence Number	Optional	Not used. Defaulted to 0.
Owning DC	Optional	Not used
Addresses	Optional	Child Node. This is published for physical warehouse only.
Pricing Location	Optional	The location for which pricing information was copied for this virtual warehouse. This is published for virtual warehouse

Table 2-124 (Cont.) Warehouse Create or Update

Message Element	Included?	Notes
Pricing Location Currency Code	Optional	This is the currency code of the pricing location. . This is published for virtual warehouse
Org Unit ID	Optional	This field holds the org unit ID of the virtual warehouse. This is published for virtual warehouse
Flex Attributes	Optional	Child Node. This is published for physical and virtual warehouses.
Warehouse Secondary Name	Optional	This field holds the secondary name of the warehouse. This Is published for physical and virtual warehouse
VAT Region	Optional	This field holds the VAT region where the physical warehouse belongs. This Is published for physical and virtual warehouse
Organization Hierarchy Type	Optional	Contains the organization type that will be used for reporting purposes for the physical warehouse. The type comes from the organizational hierarchy. Valid values are: <ul style="list-style-type: none"> • 1 = Company • 10 = Chain • 20 = Area • 30 = Region • 40 = District • 50 = Store
Organization Hierarchy Value	Optional	This field contains the code associated with the specific organizational hierarchy type. Valid values include the company number, chain number, area number, etc.
Primary Virtual Warehouse	Optional	This field holds the virtual warehouse that will used as the basis for all transactions for which only a physical warehouse and not a virtual warehouse has not been specified.
Channel Description	Optional	The channel description for which the virtual warehouse has been assigned.
Restricted Indicator	Optional	Indicates whether the inventory for this virtual warehouse is restricted, meaning it is impacted last an inbound type of transaction occurring at the physical warehouse level. Valid values are Y and N.
Protected Indicator	Optional	Indicates whether the inventory for this virtual warehouse is protected, meaning it is affected last in outbound transactions occurring at the physical warehouse level. Valid values are Y and N.
Forecast Indicator	Optional	This indicator determines if a virtual warehouse should be forecasted. Valid values are Y and N.
Transfer Entity	Optional	Indicates the transfer entity with which this virtual warehouse is associated.
Finisher Indicator	Optional	This field indicates whether a virtual warehouse is an internal finisher or not. Valid values are Y and N.

Table 2-124 (Cont.) Warehouse Create or Update

Message Element	Included?	Notes
Inbound Handling days	Optional	This field indicated the number of days that the physical warehouse requires to receive any item and get it to the shelf so that it is ready to pick.
Virtual Warehouse Type	Optional	This attribute holds values:CS_NT - Non-traditional CS_RG - CSC XD_GS - Global Sourcing XD_RG - Cross Dock
Organization Entity Type	Optional	This field will specify if the virtual warehouse is a legal entity (Importer, Exporter) or a regular warehouse. Valid values are R - regular warehouse (including finisher); M - importer; X - exporter.
Customer Order Location Indicator	Optional	This field indicates if the virtual warehouse can source or fulfill customer orders. Valid values are Y or N.
Default Warehouse	Optional	This field contains the default sourcing warehouse linked to the virtual warehouse.
Gift wrapping Indicator	Optional	This field indicates whether the warehouse supports gift wrapping.
Customer Order Shipping Indicator	Optional	This field indicates whether the virtual warehouse supports shipping customer orders.

Warehouse Addresses

Adding a new primary address for an existing physical warehouse will trigger an address detail create message.

Any updates made to the primary address information of the warehouse will trigger an address modification message. The 'key_value_1' field on this message will map to the physical warehouse.

Table 2-125 Warehouse Address Create or Update

Message Element	Included?	Notes
Module	Always	This field indicates the data type to which the address is attached. In this case, it will always be 'WH'.
Key Value 1	Always	This field holds the warehouse ID the address is attached to.
Key Value 2	No	This is not used.
Sequence Number	Always	Number indicating the sequence that addresses within the same type were entered.
Address Type	Always	This field contains the address type. Only the primary address of the primary address type Business (01) is published for the physical warehouse.
Primary Address Indicator	Always	This field indicates whether the address is the primary address for the address type.
Address 1	Always	This field contains the first line of the address.

Table 2-125 (Cont.) Warehouse Address Create or Update

Message Element	Included?	Notes
Address 2	Optional	This field contains the second line of the address.
Address 3	Optional	This field contains the third line of the address.
City	Always	This field contains the name of the city that is associated with the address.
State	Optional	This field contains the state abbreviation for the address.
Country ID	Always	This field contains the country where the address exists.
Jurisdiction Code	Optional	This field contains the ID associated to the tax jurisdiction of the country-state relationship.
Post	Optional	This field contains the postal code for the address.
Contact Name	Optional	This field contains the name of the contact for this address.
Contact Phone	Optional	This field contains the phone number of the contact person at this address.
Contact Telex	Optional	This field contains the telex number of the contact person at this address.
Contact Fax	Optional	This field contains the fax number of the contact person at this address.
Contact Email	Optional	This field contains the email address of the contact person.
Custom Flex Attributes	Optional	Child Node

Deleting Warehouses

If a warehouse is deleted in Merchandising, a delete message will be triggered to inform dependent systems.

Table 2-126 Warehouse Delete

Message Element	Included?	Notes
Warehouse	Always	This field contains the unique warehouse number that has been deleted – either a virtual or physical warehouse.
Address	Optional	Child Node

Warehouse addresses can be deleted only if they are not mandatory or if they are mandatory but are not marked as the primary address for that address type. The deletion will trigger an address delete message.

Table 2-127 Warehouse Address Delete

Message Element	Included?	Notes
Address	Always	This field contains the unique address key that has been deleted.

Flex Attributes

If you have defined any custom flex attributes (CFAs) for warehouses or warehouse addresses, they can be integrated as part of this API. The node of the integration that supports this will accept the name of the attribute as it is defined in the group set level view and the value for the attribute.

Table 2-128 Custom Flex Attributes

Message Element	Required?	Notes
Name	Always	Indicates the name of the column defined in the group set view for flex attributes defined for transfers.
Value	Always	Indicates the value of the attribute for the transfer if the attribute is a character or number.
Value Date	Always	Indicates the value of the attribute for the transfer if the attribute is a date.

Error Handling

If any errors are encountered in the validations described above or any of the message structure validations, a status of E is returned to the external system along with the appropriate error message. If the message has been successfully persisted, a success status (S), is returned to the external system indicating that the message has been successfully received and persisted to the Merchandising database.

Message XSD

Below are the filenames that correspond with each message type. Please consult the *Oracle Retail Integration Guide* for each message type for the details on the composition of each message.

Message Types	Message Type Description	XML Schema Definition (XSD)
WHCre	WH Create Message	WHDesc.xsd
WHMod	WH Modify Message	WHDesc.xsd
WHDel	WH Delete Message	WHRef.xsd
WHDtlCre	WH Detail Create Message	WHDesc.xsd
WHDtlMod	WH Detail Modify Message	WHDesc.xsd
WHDtlDel	WH Detail Delete Message	WHRef.xsd
WHAddCre	WH Address Create	WHAddrDesc.xsd
WHAddMod	WH Address Modify	WHAddrDesc.xsd

Work Orders In Publication API

This section describes the work order in publication API.

Functional Area

Purchase Orders

Business Overview

A work order provides direction to a warehouse, such as Oracle Retail Warehouse Management (RWMS), about work that needs to be completed on items contained in a purchase order prior to shipping them on to stores or other warehouses. Merchandising publishes work orders soon after it publishes the purchase order itself. This is referred to as a Work Order In message. Work orders are defined at the physical location level. The message family manager will send to the warehouse at which the work order will be done. This is used by the RIB publication adaptor for routing messages to the appropriate warehouse.



Note:

This integration is not used in Merchandising's integration to Oracle WMS Cloud.

New Work Order

Creating a new work order for an approved order triggers a message to be sent to notify the warehouse of additional activities that must be performed on the order upon receipt. The message includes the following: work order ID, order number, warehouse that will perform the work, the locations where the items on the order will be sent onto, and details on the items and work to be done, including a sequence and WIP (work in progress) code, which is stored in Merchandising codes table, under code type WWIP. It is assumed that the WIP codes used by Merchandising are coordinated with your warehouses that will be receiving the updates.

The format for creating and updating inbound work orders is shown below.

Table 2-129 Work Order Header

Message Element	Included?	Notes
Work Order ID	Always	This field contains the unique identifier for work in process associated with an order or transfer.
Order Number	Always	This field contains the order number associated with the work order.
Work Order Detail	Always	Child node

Table 2-130 Work Order Detail

Message Element	Included ?	Notes
Warehouse	Always	The physical warehouse where the work order will be done.
Item	Always	The item on which the work order will be done.
Location Type	Always	The location type of the final destination. Valid values are: S - Store W - Warehouse
Location	Always	The final destination for the item on the order. If the item is pre-distributed this location will be the store or warehouse on the allocation. If not, it will be the same warehouse as above.
Sequence Number	Always	The sequence number that is needed to ensure each record is unique, since the same work-in-progress code can be listed more than once on the same work order.
WIP Code	Always	The WIP code for the work to be done on the item. Valid values are in the codes table under the WWIP code type.
Instructions	No	Not used

Updated Work Order

When an existing work order for an approved order is updated, an update message is triggered to provide the details of the update. The update message, like create, will contain the full details in the message for all fields in the message.

Deleted Work Order

When a work order is deleted for a purchase order, this will also trigger a delete transaction to be sent to notify external systems. The delete message will include the work order ID, order number at the header level and the warehouse, item, location type, location, sequence number and WIP code at a detail level. It is also possible to delete just a detail from the work order.

The format for deleting inbound work orders or work order details is shown below.

Table 2-131 Work Order Header

Message Element	Included?	Notes
Work Order ID	Always	This field contains the unique identifier for work in process associated with an order or transfer.
Order Number	Always	This field contains the order number associated with the work order.
Work Order Detail	Optional	Child node

Table 2-132 Work Order Detail

Message Element	Included?	Notes
Warehouse	Always	The physical warehouse where the work order will be done.
Item	Always	The item on which the work order will be done.
Location Type	Always	The location type of the final destination. Valid values are: S - Store W - Warehouse
Location	Always	The final destination for the item on the order. If the item is pre-distributed this location will be the store or warehouse on the allocation. If not, it will be the same warehouse as above.
Sequence Number	Always	The sequence number that is needed to ensure each record is unique, since the same work-in-progress code can be listed more than once on the same work order.
WIP Code	Always	The WIP code for the work to be done on the item. Valid values are in the codes table under the WWIP code type.

Error Handling

If any errors are encountered while publishing the message, a fatal error with status E (Error) is sent to RIB. This would bring down the RIB adapter thus preventing any further messages from being processed until this is resolved.

Message XSD

Below are the filenames that correspond with each message type. Please consult the Oracle Retail Integration Guide for each message type for the details on the composition of each message.

Message Types	Message Type Description	XML Schema Definition (XSD)
InBdWOCre	Work Order In Create	WOInDesc.xsd
InBdWOMod	Work Order In Modify	WOInDesc.xsd
InBdWODel	Work Order In Delete	WOInRef.xsd

Work Orders Out Publication API

This section describes the Work Orders out Publication API.

Functional Area

Transfers

Business Overview

This publication API facilitates the transmission of outbound work orders for the finisher from Merchandising to external systems. Only transfers that pass through a finisher before reaching the final location can be associated with work orders. The work order provides instructions for one or more of the following tasks to be completed at the finisher location:

- Perform an activity on an item, such as monogramming or ticketing.
- Transform an item from one thing into another, such as dyeing a white t-shirt black.
- Combine bulk items into a pack or break down a pack into its component items.

Outbound work orders are not bundled with transfer messages, because multi-legged transfers can be routed to either internal finishers (held as virtual warehouses) or external finishers (held as partners).

All activities, transformations, and packing details are contained in the same message. Because Merchandising does not allow users to modify work order activities, transformation or packing information for an approved transfer (it must be brought back to Input status), separate detail-level messages of any type (create, delete, update) are never published.

Outbound work order delete messages are published when the second leg of a multi-legged transfer is unapproved. This can be accomplished through the un-approval of an entire multi-legged transfer or the un-approval of the second leg only. A two-leg transfer that has had the first leg shipped can be set back to In Progress status to make changes to the work order activities and the final location. When the action has occurred, only the second leg is really set back to in progress. The first leg remains in shipped status.

New Work Order

A new work order is created when a transfer with finishing is approved. This triggers a message to be sent to notify the external system. The message includes the following:

- Transfer work order ID
- Finisher ID
- Transfer number
- Transfer parent number
- Inventory type (available or unavailable)
- Work order detail information - including item, destination location, item's inventory status, and the work order activity information (costs, comments)
- Transformation details - including the from and to item IDs
- Packing information - including the from items and to items

The format for creating and updating outbound work orders is shown below.

Table 2-133 Work Order Header

Message Element	Included?	Notes
Work Order ID	Always	This field contains the unique identifier for the work in process associated with a transfer.
Destination ID	Always	This field contains the source location of the transfer.

Table 2-133 (Cont.) Work Order Header

Message Element	Included?	Notes
Distro Number	Always	This field contains the first leg transfer number of the 2-legged transfer.
Work Order Detail	Always	Child node
Distro Parent Number	Optional	This field contains the second leg transfer number of the 2-legged transfer.
Work Order Transformation	Optional	Child node
Work Order Packing	Optional	Child node
Inventory Type	Optional	This field contains the code to indicate whether the inventory on the transfer is available. Valid values are Available (A) and Unavailable (U).

Table 2-134 Work Order Detail

Message Element	Included?	Notes
Destination ID	Always	This field contains to the destination location for the transfer.
Item ID	Always	This field contains the item on which the work order will be performed.
Work In Process Sequence Number	No	Not used
Work In Process Code	No	Not used
Personalization	No	Not used
Instructions	No	Not Used
Order Line Number	No	Not Used
Auto Complete	No	Not Used
Work Order Activity	Optional	Child node
Inventory Status	Optional	This field contains the inventory status of the transfer detail.

Table 2-135 Work Order Activity

Message Element	Included?	Notes
Activity ID	Always	This field contains a code number indicating the activity that will be performed on the item.
Activity Cost	Optional	This field contains the per unit cost of the finishing activity.
Comments	Optional	This field contains any comments regarding this activity.

Table 2-136 Work Order Transformation

Message Element	Included?	Notes
From Item	Always	This field contains the original item on the transfer.
To Item	Always	This field contains the item which results from the finishing activity.

Table 2-137 Work Order Packing

Message Element	Included?	Notes
To Pack	Always	Child node
From Pack	Always	Child node

Table 2-138 Work Order Pack To

Message Element	Included?	Notes
To Item	Always	This contains the item resulting from the packing process.

Table 2-139 Work Order Pack From

Message Element	Included?	Notes
From Item	Always	This contains the component item used in the packing process.

Deleted Work Order

When a transfer with finishing is deleted or unapproved, this will trigger a delete transaction message to an external system. The delete message contains the work order ID to be deleted.

Table 2-140 Work Order Header

Message Element	Included?	Notes
Work Order ID	Always	This field contains the unique identifier of the work order being deleted.
Destination ID	Always	This field contains the source location of the transfer.
Distro Number	Always	This field contains the first leg transfer number of the 2-legged transfer.
Work Order Detail	Always	Child node
Distro Parent Number	Optional	This field contains the second leg transfer number of the 2-legged transfer.

Table 2-141 Work Order Detail

Message Element	Included?	Notes
Work Order ID	Always	This field contains the unique identifier of the work order being deleted.
Destination ID	Always	This field contains to the destination location for the transfer.
Item ID	Always	This field contains the item on which the work order will be performed.
Work In Process Sequence Number	No	Not used
Order Line Number	No	Not used

Error Handling

If any errors are encountered while publishing the message, a fatal error with status E (Error) is sent to RIB. This will bring down the RIB adapter, thus preventing any further messages from being processed until this is resolved.

Message XSD

Here are the filenames that correspond to each message type. Please consult RIB documentation for each message type to get a detailed picture of the composition of each message.

Message Types	Message Type Description	XML Schema Definition (XSD)
OutBdWoCre	Work Order Create Message	WOOOutDesc.xsd
OutBdWoDel	Work Order Delete Message	WOOOutRef.xsd

3

RIB Subscription Designs

This chapter provides an overview of the RIB subscription APIs used in by Merchandising.

Allocation Subscription API

This section describes the allocation subscription API.

Functional Area

Allocation

Business Overview

This API allows an external application to create, update, and delete allocations within Merchandising. The Oracle Retail Allocation Cloud Service does **not** use this API to interface allocations to Merchandising. Allocations created or updated using this API include those that are based on warehouse inventory, as well as those using inventory from another transaction, such as a purchase order, another allocation, a transfer, or a shipment (Vendor ASN or BOL). When allocating using something other than warehouse inventory, the ID of the transaction must also be included.

Allocations only involve stockholding locations. This includes the ability to process allocations to both company and franchise stores, as well as any stockholding warehouse location, excepting internal finishers. If an allocation for a franchise store is received, Merchandising will also create a corresponding franchise order. This API supports multiple types of destination locations (warehouses as well as stores) as part of the detail section within the same message.

Allocation details can be created, edited, or deleted within the allocation message, including adding new locations to existing allocations, or modifying quantities for existing locations. If modifying an existing location, it assumes the passed-in quantity is an adjustment to the current quantity as opposed to an overwrite.

Details can be individually removed from an allocation if the detail is not in-transit, received, or in progress. An entire allocation can be deleted if none of details are in-transit or received or in progress.

Allocation Creation

When new allocations are being created, this API first validates that all required fields are present. After that, business-level validation on the input information will be performed. Detail line items must exist on an allocation header create message for an allocation to be created. New item location relationships will be created for allocation detail line items that did not previously exist.

Allocation Modification

Modifying an existing allocation will first validate the existence of the allocation header for modification. Only the allocation description and release date on the header can be modified.

Table 3-1 Allocation Create or Update

Message Element	Included?	Notes
Allocation	Always	Contains the unique identifier of the allocation. This should fall within the range of Merchandising IDs already designated for allocations.
Allocation Description	Always	Contains the user defined description of the allocation.
Order No.	Optional	Contains the purchase order with which the allocation is associated. Only used if the source for the allocation is a purchase order.
Item	Always	Contains the transaction level item that is being allocated.
From Location	Always	Contains the location that is the source of the allocation. This must be a valid stockholding virtual warehouse.
Release Date	Optional	Contains the earliest date on which the warehouse should ship the allocation.
Origin Indicator	Optional	Indicates the source application that sends the allocation. Valid values are Oracle Retail Advanced Inventory Planning (AIP) and externally generated (EG). It will be defaulted to EG, if not provided.
Allocation Details	Optional	Child Node
Doc ID	Optional	Contains identification number for a transfer, another allocation, bill of landing number (BOL), or advanced shipping notice (ASN) number for a purchase order. This field is populated according to documentation type and indicates where the inventory for the allocation should be sourced. This should be null if the source for the allocation is warehouse inventory or a purchase order.
Documentation type	Optional	Contains the type of allocation product source. Valid values are ASN, Transfer (TSF), Bill of Lading (BOL), or Allocation (ALLOC). When this is passed in as null, the source is assumed to be a PO or warehouse inventory if not PO number provided.

Allocation Details

New detail records can be added to an existing allocation. Modification of the existing details on the allocation is also supported by this API. If modifying an existing location, Merchandising assumes the passed-in quantity is an adjustment to the current quantity as opposed to an overwrite. The API verifies the allocation is not in-transit, received, nor in progress and that the quantity does not fall to zero or below.

Table 3-2 Allocation Details Create or Update

Message Element	Included?	Notes
To Location	Always	Contains the destination location of the allocation. This must be an active stockholding store or virtual warehouse.
To location Type	Always	Contains the type of the destination location. Valid values are S (store) and W (warehouse)
Quantity Allocated	Always	Contains the allocated quantity of the item for the destination location. When the allocation is being created this value must be a positive integer. If this value is being modified, it will contain the quantity adjusted (positive or negative), rather than an override value
In Store Date	Optional	Contains the date the item is to be in store. This date will be included in the Merchandising publication for communication to the warehouse.

Deleting Allocations

Allocation Header Delete

During an allocation header delete, on successfully validating the information in the message, the allocation header is updated to Cancelled (C) status.

Message Element	Included?	Notes
Allocation	Always	Contains the unique identifier of the allocation being deleted.
Allocation Detail	Optional	Child Node

Allocation Details Delete

The detail records are validated and deleted from the allocation on receipt of a detail delete message.

Message Element	Included?	Notes
To Location	Always	Contains the destination location being deleted.
To Location Type	Always	Contains the location type of the destination location being deleted.

Error Handling

If any errors are encountered in the validations described above, or in any of the message structure validations, a status of E is returned to the external system along with the appropriate error message. If the message has been successfully persisted, a success status (S), is returned to the external system indicating that the message has been successfully received and persisted to the Merchandising database.

Message XSD

Below are the filenames that correspond with each message type. Please consult the *Oracle Retail Integration Guide* for each message type for the details on the composition of each message.

Message Type	Message Type Description	XML Schema Definition (XSD)
XAllocCre	External Allocation Create via RIB	XAllocDesc.xsd
XAllocDel	External Allocation Delete via RIB	XAllocRef.xsd
XAllocDtlCre	External Allocation Detail Create via RIB	XAllocDesc.xsd
XAllocDtlDel	External Allocation Detail Delete via RIB	XAllocRef.xsd
XAllocDtlMod	External Allocation Detail Modification via RIB	XAllocDesc.xsd
XAllocMod	External Allocation Modification via RIB	XAllocDesc.xsd

Appointment Subscription API

This section describes the appointments subscription API.

Functional Area

Inventory

Business Overview

Merchandising subscribes to an appointment when merchandise arrives at a location. It also processes the appointment messages and attempts to receive against and close out the appointments. In addition, Merchandising attempts to close the document that is related to the appointment, when applicable. A document can be a purchase order, a transfer, or an allocation.

New Appointments

An appointment is created when a shipment is about to arrive at a location. After performing the business-level validation, this message will update the status to Scheduled (SC). This message contains the location-level information.

New Appointment Details

This message contains the item information associated with an appointment, including the ASN and the document number (PO, transfer or allocation).

Updated Appointments

An update message updates the status of an existing appointment if already exists. Valid values for the status column include:

- SC—Scheduled

- MS–Modified Scheduled
- AR–Arrived
- AC–Closed

Updated Appointment Details

This message updates an appointment detail record that was previously sent, such as an update to the quantity for an item. If the record doesn't already exist, it is added.

Table 3-3 Appointment Header

Message Element	Required?	Notes
Appointment Number	Always	This field contains the unique number generated by warehouse while creating an appointment.
From Location	Always	This field contains the location where the merchandise has been sent.
To Location	Optional	Not used.
Appointment Start	Optional	Not used.
Appointment Action Status	Always	This field contains the status of the appointment. Valid values include: SC - Scheduled, MS - Modified Scheduled, AR - Arrived, and AC-Closed.

Table 3-4 Appointment Detail Message

Message Element	Required?	Notes
Item	Always	This field contains the items shipped to the location.
Unit Quantity	Always	This field contains the quantity of the item slated to be sent to the location.
PO Number	Always	This field contains the purchase order, transfer or allocation corresponding to the shipped merchandise.
Document type	Always	This field indicates the type of document corresponding to the shipped merchandise. Possible choices are Purchase Order (P), Transfer (T), or Allocation (A).
ASN Number	Optional	This field contains the advance shipping notice number associated with the appointment. It is populated only when the appointment is based on an ASN.

Delete Appointments

This message deletes any existing appointment and detail records.

Delete Detail Level Appointments

This API deletes a detail record for an existing appointment.

Table 3-5 Appointment Header

Message Element	Required?	Notes
Appointment Number	Always	This field contains appointment number being deleted.
From Location	Always	This field contains the location from which the appointment is being deleted.

Table 3-6 Appointment Detail Message

Message Element	Required?	Notes
Item	Always	This field contains the item being deleted from the appointment.
PO Number	Always	This field contains the purchase order, transfer or allocation containing the item being deleted from the appointment.
ASN Number	Optional	This field contains the advance shipping notice number associated with the item/appointment being deleted. It is populated only when the appointment is based on an ASN.

Error Handling

If any errors are encountered in the validations described above or any of the message structure validations, a status of E is returned to the external system along with the appropriate error message. If the message has been successfully persisted, a success status (S), is returned to the external system indicating that the message has been successfully received and persisted to the Merchandising database.

Message XSD

Below are the filenames that correspond with each message type. Please consult the Oracle Retail Integration Guide for each message type for the details on the composition of each message.

Message Types	Message Type Description	XML Schema Definition (XSD)
Appointcre	Appointment Create Message	AppointDesc.xsd
Appointhdmod	Appointment Header Modify Message	AppointDesc.xsd
Appointdel	Appointment Delete Message	AppointRef.xsd
Appointdtlcre	Appointment Detail Create Message	AppointDesc.xsd
Appointdtlmod	Appointment Detail Modify Message	AppointDesc.xsd
Appointdtldel	Appointment Detail Delete Message	AppointRef.xsd

ASN Outbound Subscription API

This section describes the ASN outbound subscription API.

Functional Area

Inventory

Business Overview

Merchandising receives advanced shipping notifications (ASNs), also known as a bill of lading (BOL) messages, from a warehouse management system (WMS), like Oracle Warehouse Management Cloud, or a store inventory system like Oracle Retail Store Inventory and Operations Cloud Service (SIOCS).

These ASNs are notifications to Merchandising that inventory is moving from one location to another. These notification messages contain data that is used by Merchandising to create or modify a shipment record. ASNs are received for:

- Pre-existing allocations.
- Pre-existing transfers.
- Externally generated transfers, created in the store or warehouse (created as transfer type of EG within Merchandising).

An ASN message may contain multiple transfers or allocations, and as a result, the shipment record in Merchandising will reflect these multiple movements of merchandise. A BOL number on the shipment record is a means of tracking one or more transfers and allocations back through the respective stock order records. Shipments for customer orders, franchise orders, and franchise returns are also managed through this API. If the receiving location is a non-stockholding location, like in the case of a warehouse shipment to a non-stockholding franchise store, or a warehouse shipment direct to a customer (that is processed through a non-stockholding store) then the shipment will be auto-received when processed by Merchandising.



Note:

ASNs related to a purchase order from a supplier are classified as an Inbound ASNs. Details for those types of expected shipments are found in the ASN In Subscription API section.

Other Notes

- For customer order fulfillment, SIOCS will send an ASN Out message that does not include a ship-to location. Merchandising ignores these messages.
- Store to customer fulfillment request will not have associated transfer in Merchandising. When Oracle Retail Store Inventory and Operations Cloud Service (SIOCS) ships the customer order, SIOCS will generate an Outbound ASN message with an empty To Location or with Location Type as Customer (C). Since there are no associated transfers in Merchandising, Merchandising will not process these Outbound ASN messages. The reserved inventory will be backed out in Merchandising when Merchandising processes a SALES transaction backend.
- Messages consumed through this API can create new shipments or update existing shipments. A new shipment record will be created in Merchandising with Input status if the BOL number is not yet associated to any shipment record. If the BOL number is

already associated to a shipment record, the shipment record will be updated accordingly.

- The Universal Identification Number (UIN) child node may be included in the message, but this information is not used by Merchandising. This is used by SIOCS.
- A Shipment and Carton Custom Flex Attribute child nodes can be included in the message, but this information is not used by Merchandising. This is used by SIOCS.

Shipment Message Details

When inventory is shipped from one location to another, Merchandising will be notified and will then create a shipment record based on the content of the message received. If the shipment already exists, the details of the existing shipment will be updated. The message includes the following:

ASNOUT Details

Message Element	Required?	Notes
Schedule Number	Optional	Not used by Merchandising.
Auto Receive Indicator	Optional	Not used by Merchandising.
To Location	Optional	This field contains the location where the shipment will be delivered.
To Location Type	Optional	This field contains the location type of the location where the shipment will be delivered to. Valid values are store (S), warehouse (W), or finisher (E).
To Store Type	Optional	Not used by Merchandising.
To Stockholding Indicator	Optional	Not used by Merchandising.
From Location	Always	This field contains the location from which the shipment was sourced. This applies to transfer and allocation shipments.
From Location Type	Optional	This field contains the location type of the location from which the shipment was sourced. Valid values are store (S) or warehouse (W).
From Store Type	Optional	Not used by Merchandising.
From Stockholding Indicator	Optional	Not used by Merchandising.
ASN Number	Optional	This field contains the bill of lading number associated with the shipment.
ASN Type	Optional	Not used by Merchandising.
Container Quantity	Optional	This field contains the number of boxes associated with the shipment.

Message Element	Required?	Notes
BOL Number	Always	This field contains the transaction sequence number from the transfer shipment confirmation process. This is the same as the ASN Number.
Shipment Date	Optional	This field contains the date the transfer and/or allocation was shipped.
Estimated Arrival Date	Optional	This field contains the estimated arrival date when the shipment is expected to arrive at the destination.
Shipment Address 1	Optional	Not used by Merchandising.
Shipment Address 2	Optional	Not used by Merchandising.
Shipment Address 3	Optional	Not used by Merchandising.
Shipment Address 4	Optional	Not used by Merchandising.
Shipment Address 5	Optional	Not used by Merchandising.
Shipment City	Optional	Not used by Merchandising.
Shipment State	Optional	Not used by Merchandising.
Shipment Zip Code	Optional	Not used by Merchandising.
Shipment Country ID	Optional	Not used by Merchandising.
Trailer Number	Optional	Not used by Merchandising.
Seal Number	Optional	Not used by Merchandising.
Transshipment Number	Optional	Not used by Merchandising.
Comments	Optional	This field contains any miscellaneous comments about the shipment.
Carrier Code	Optional	This field contains the courier that will deliver the shipment.
Carrier Service Code	Optional	This field contains the service level code for the courier that will deliver the shipment. Valid values are found in code type CSVG. Not used by Merchandising.
System Code	Optional	Not used by Merchandising.
From Location Virtual Warehouse	Optional	Not used by Merchandising.

Child Nodes

- ASNOUT Distro Details

ASNOUT Distro Details

This level of the message contains the details about the individual stock orders contained in the shipment.

Message Element	Required?	Notes
Distro Number	Always	This field contains the transfer number or allocation number associated with the shipment.
Distro Doc Type	Always	This field determines if the Distro Number specified is a Transfer (T) or an Allocation (A).
Customer Order Number	Optional	This contains the customer order number associated with the transfer on the shipment.
Fulfill Order Number	Optional	This contains the fulfillment order number associated with the customer order number for the transfer on the shipment.
Consumer Direct	Optional	Not used by Merchandising.
Comments	Optional	This field contains any comments about the stock orders contained in the shipment.

Child Nodes

- ASNOUT Carton Details

ASNOUT Carton Details

This section of the message contains details about the cartons for a distro on a shipment.

Message Element	Required?	Notes
Final Location	Optional	Not used by Merchandising.
Container ID	Always	This field contains the carton number for shipments originating from the ASN process as carton shipments. This field will be zero for all shipments that are not at a carton level.
Container Weight	Optional	Not used by Merchandising.
Container Length	Optional	Not used by Merchandising.
Container Width	Optional	Not used by Merchandising.
Container Height	Optional	Not used by Merchandising.
Container Cube	Optional	Not used by Merchandising.
Expedite Flag	Optional	Not used by Merchandising.
In Store Date	Optional	Not used by Merchandising.

Message Element	Required?	Notes
Tracking Number	Optional	This field contains a unique tracking number that is used to track containers through a carrier's system. Not used by Merchandising.
Freight Charge	Optional	Not used by Merchandising.
Master Container ID	Optional	Not used by Merchandising.
Comments	Optional	This field contains any comments about the shipment container.
Weight	Optional	This field contains the actual weight shipped for the container.
Weight UOM	Optional	This field contains the unit of measurement for weight (for example, pounds, kilograms) that was shipped.
Carrier Shipment Number	Optional	Not used by Merchandising.
Original Item ID	Optional	Not used by Merchandising.

Child Nodes

ASNOUT Item Details

ASNOUT Item Details

This section outlines details about the items in the carton.

Message Element	Required?	Notes
Item ID	Always	This field contains the unique identifier for the item.
Unit Quantity	Always	This field contains the quantity of the item shipped in the carton for this shipment in the standard unit of measure.
Gross Cost	Optional	Not used by Merchandising.
Priority Level	Optional	Not used by Merchandising.
Order Line Number	Optional	This field is used to carry the customer order line number value for customer orders. This field is not used by Merchandising for non-customer orders.
Lot Number	Optional	Not used by Merchandising.
Final Location	Optional	Not used by Merchandising.

Message Element	Required?	Notes
From Disposition	Optional	This value is used to determine if the quantity shipped is available or unavailable. Valid values for this field are inventory status codes (INV_STATUS_CODE) in the INV_STATUS_CODES table in Merchandising.
To Disposition	Optional	Not used by Merchandising.
Voucher Number	Optional	Not used by Merchandising.
Voucher Expiration Date	Optional	Not used by Merchandising.
Container Quantity	Optional	Not used by Merchandising.
Comments	Optional	Not used by Merchandising.
Unit Cost	Optional	This field contains the unit cost of the item in the shipment. This is used only for the Brazil Localization setup to calculate transaction code 74 (Recoverable Tax for Destination Location). In all other cases this should be NULL.
Base Cost	Optional	This value will be used for the Brazil Localization setup only to get the base cost (BC) from Fiscal Management for a transfer, which will flow into Merchandising. In all other cases this should be NULL.
Weight	Optional	This field contains the actual weight shipped. This may be included for catch weight items. If not included Merchandising will use the average weight or nominal weight for the item at the shipping location.
Weight UOM	Optional	This field contains the unit of measurement for weight (for example, pounds, kilograms) shipped. Required if weight is included in the message.

Error Handling

If any errors are encountered in the validations described above or any of the message structure validations, a status of E is returned to the external system along with the appropriate error message. If the message has been successfully persisted, a success status (S), is returned to the external system indicating that the message has been successfully received and persisted to the Merchandising database.

Message XSD

Below are the filenames that correspond with each message type. Please consult the *Oracle Retail Integration Guide* for each message type for the details on the composition of each message.

Message Types	Message Type Description	XML Schema Definition (XSD)
asnoutcre	ASN Outbound Create Message	ASNOutDesc.xsd

Cost Change Subscription API

This section describes the cost change subscriptions.

Functional Area

Price and Cost

Business Overview

Merchandising exposes an API that will allow external systems to update unit cost within Merchandising. Cost changes can be performed at the item level, or at the following levels of the organization hierarchy: chain, area, region, district, and store. Unit costs are updated for all stores within the location group. Because warehouses are not part of the organization hierarchy, they are only impacted by cost changes applied at the warehouse level.

All cost changes that are sent through this API are executed immediately. The cost change subscription creates both the cost change events with an effective date of the current date, as well as updates unit costs for item/locations that already exist in Merchandising. It does not create or delete item/locations in Merchandising.

In addition to RIB, Merchandising also exposes this API as a web service. The web service takes in a collection of cost changes and will return success and failure through the service response object. See the "[Cost Change Service](#)" section of this document in the "[Provider Services](#)" section of "[SOAP Web Services](#)" more information.

This API checks that the required fields are provided and checks the supplier's currency and the item status. If differentiator IDs are passed in, it verifies that they are valid for the passed in item. The API also retrieves the following:

- Transaction level items, if the passed in item is an item parent
- All locations based on the passed in hierarchy type and value, if provided.
- All item/location combinations where the passed in supplier/country is the primary supplier/country at an item location.
- All orderable buyer packs that the passed-in item or its children, if above transaction level
- All item/locations on approved (and worksheet) order, if the recalculate order indicator is set to Yes.

This API will perform the following actions:

- Create a cost change event in Executed status, with an effective date of the current date.

- Update the unit cost in Merchandising for all item/supplier/country and item/supplier/country/locations based on the information provided.
- Create price history for all item/locations that got updated as part of the cost change.
- If the recalculate order indicator is Yes, update all relevant order/item/locations unit cost in merchandising.

It is important to note that cost changes sent through this API do not include estimated landed costs. The cost updated here is the default purchase cost, before any deals, that will be used for purchase orders created in Merchandising, similar to cost changes initiated in Merchandising.

The format for creating and updating cost changes is shown below.

Table 3-7 Cost Change

Message Element	Required?	Notes
Item	Yes	This field contains the item to which the cost change applies. It can be a parent item, but its item level cannot be greater than its transaction level. This cannot be a buyer pack.
Supplier	Yes	This field contains the number of the supplier. The supplier must be a valid supplier in merchandising. This can be a primary or non-primary supplier.
Origin Country ID	Yes	This field contains the identifier of the origin country of the item/supplier to which the cost change will apply. This value must be a valid country in merchandising. This can be the primary or a non-primary country.
Diff ID	No	This field contains the identifier for a differentiator. This can be used with a parent or grandparent item. This value must be a valid differentiator ID in merchandising.
Unit Cost	Yes	This field contains the new unit cost of the item in the currency specified on the message.
Recalculate Order Indicator	Yes	This field indicates if orders in approved status for items on the cost change will be recalculated with the new cost. Valid values are Yes (Y) and No (N).
Currency Code	Yes	This field contains the currency code of the unit cost. This must be a valid currency code in merchandising.
Hierarchy Level	No	This field indicates the level of the organizational hierarchy to which the cost change applies. Valid values are: <ul style="list-style-type: none"> • CH - chain • AR - area • RE - region • DI - district • S - store • W - warehouse
Cost Change Hierarchy	No	Child node
Custom Flex Attributes	No	Child node

Table 3-7 (Cont.) Cost Change

Message Element	Required?	Notes
Purchase Rate	No	This field contains the percentage of the retail price which will determine the cost paid to the supplier for a consignment or concession item.

Table 3-8 Cost Change Hierarchy

Message Element	Required?	Notes
Hierarchy Value	Yes	This field contains the hierarchy value at the specified level of the hierarchy which encompasses the locations affected by the cost change. This value must exist as a valid chain, area, region, district, store or warehouse in Merchandising.

Custom Flex Attributes

If any custom flex attributes (CFAS) for the cost change have been added or modified, it will trigger an update message. The node of the integration that supports this will contain the name of the attribute as it is defined in the group set level view, the value of the custom attribute. If it is a date attribute, the date value is in a separate field. Flex attributes can only be added to or updated; they cannot be deleted.

Table 3-9 Flex Attributes

Message Element	Required?	Notes
Name	Yes	Holds the attribute name.
Value	Conditional	Holds the value of the attribute for number and character type attributes. Either the value or the value date field should be provided.
Value Date	Conditional	Holds the date for date type attributes. Either the value or the value date field should be provided.

Error Handling

This API ensures that the correct message type is passed in for cost change messages. If the message type is invalid, an error status is returned to the external system, along with the appropriate error message. This is to inform the external system that the message type is invalid.

The standard error handling functions of Merchandising are in place in this API and return messages as appropriate to the outcome.

Message XSD

Below are the filenames that correspond with each message type. Please consult the Oracle Retail Integration Guide for each message type for the details on the composition of each message.

Message Type	Message Type Description	XML Schema Definition (XSD)
xcostchgmod	External Cost Change Modify	XCostChgDesc.xsd

Currency Exchange Rate Subscription API

This section describes the currency exchange rates subscription API.

Functional Area

Foundation

Business Overview

Currency exchange rates constitute financial information that is subscribed to by Merchandising. A currency exchange rate is the price of one country's currency expressed in another country's currency. This API assumes the currency codes are already present in Merchandising. This API supports creating new rates by date and updating existing rates for the same conversion date. Deleting previously created rates is not supported.

Table 3-10 Currency Exchange Rates

Message Element	Required?	Notes
From Currency	Yes	The source currency code of the currency exchange rate.
To Currency	Yes	The resultant currency code of the currency exchange rate.
Conversion Date	Yes	Contains the date on which the currency rate became or will become active.
Conversion Rate	Yes	The exchange rate between the two currency codes for the type and effective date.
Conversion Type	Yes	This field identifies the type of exchange rate the history exists for. Valid values are defined under code type EXTP: <ul style="list-style-type: none"> • Consolidation – C • Operational – O • Letter of Credit/Bank – L • Purchase Order – P • Customs Entry – U • Transportation - T

Error Handling

If any errors are encountered in the validations described above or any of the message structure validations, a status of E is returned to the external system along with the appropriate error message. If the message has been successfully persisted, a success status (S), is returned to the external system indicating that the message has been successfully received and persisted to the Merchandising database.

Message XSD

Here are the filenames that correspond with each message type. Please consult the RIB documentation for each message type to get a detailed picture of the composition of each message.

Message Types	Message Type Description	XML Schema Definition (XSD)
CurrRateCre	Currency Rate Create Message	CurrRateDesc.xsd
CurrRateCre	Currency Rate Modify Message	CurrRateDesc.xsd

Customer Order Fulfillment Subscription API

Functional Area

Customer Order Fulfillment

Business Overview

Merchandising provides an interface to process Customer Order Fulfillment requests from an external order management system (OMS). If the system option OMS_IND = 'Y', then Merchandising expects to receive customer orders via this API. If the system option PERSIST_CUSTOMER_DATA_IND = 'N', personal information will not be stored in the customer order table in Merchandising.

Merchandising supports two integration methods for processing Customer Order Fulfillment messages from OMS - either through RIB or Web service. At implementation time, clients should decide on either one or the other integration method, but not both. The same core logic is used to validate and persist customer orders to Merchandising tables.

- In a RIB implementation, Merchandising subscribes to Customer Order Fulfillment messages. When a customer order is created, or partially or fully cancelled, the customer order information is sent from the Order Management System (OMS) to the RIB. Merchandising subscribes to the customer order information as published from the RIB and places the information onto Merchandising tables.
- In a web service implementation, Merchandising exposes a FulfillOrder Web service to create or cancel a customer order in Merchandising. OMS will invoke the service with customer order details to place the information on Merchandising tables. See [Customer Order Fulfillment Service](#) in the "SOAP Web Services" chapter of this document for more details on this method.

The Customer Order Fulfillment message staged will go through a process of validation. Records that pass validation will create new customer order records. If any validation error occurs, transaction will be rolled back, and no customer orders will be created.

There are two scenarios where a customer order fulfillment request cannot be created in Merchandising:

1. Due to data validation errors (for example, invalid item).

2. Due to 'No Inventory' - There is not enough inventory available at the source location or item is not ranged or inactive at the source location, or item is not supplied by the supplier (in a PO scenario).

Other Notes

1. Non-stockholding franchise stores cannot part of a fulfillment order, either as a sourcing location or as a fulfillment location.
2. Catch weight and transformable sellable items are not supported in this integration. To sell items that can vary by weight, like bananas through online channels, setup should be done as a regular (non-catch weight) item with a unit cost and standard UOM defined in items of eaches.
3. It is assumed that customer orders will be captured in the selling UOM in OMS, but that all transactions will be communicated to Merchandising in standard UOM.
4. If the same customer order fulfillment request is sent for a different item or for an existing item but with a different item line number, the existing PO or transfer will be updated.

The Customer Order Fulfillment messages contain information such as delivery type, source type and destination type. Based on these, the system should proceed to create a Purchase Order, Transfer or Inventory Reservation. The table below shows the customer order scenarios for the combination of delivery type, source type and destination type.

Scenario #	Source Location	Fulfillment Location	Delivery Type	Transaction created
1	Warehouse	Store	Pickup in Store	Virtual WH to Physical Store Transfer + Reservation. FulfilOrdDesc will contain: 1st leg: source_loc_type = 'WH', fulfill_loc_type = 'S' 2nd leg: source_loc_type = NULL, fulfill_loc_type = 'S'.
2	Warehouse	Store	Ship to Customer	Virtual WH to Physical Store Transfer + Reservation. FulfilOrdDesc will contain: 1st leg: source_loc_type = 'WH', fulfill_loc_type = 'S' 2nd leg: source_loc_type = NULL, fulfill_loc_type = 'S'.
3	Store A	Store B	Pickup in Store	Physical Store to Physical Store Transfer + Reservation. FulfilOrdDesc will contain: 1st leg: source_loc_type = 'ST', fulfill_loc_type = 'S' 2nd leg: source_loc_type = NULL, fulfill_loc_type = 'S'.

Scenario #	Source Location	Fulfillment Location	Delivery Type	Transaction created
4	Store A	Store B	Ship to Customer	Physical Store to Physical Store Transfer + Reservation. FulfilOrdDesc will contain: 1st leg: source_loc_type = 'ST', fulfill_loc_type = 'S' 2nd leg: source_loc_type = NULL, fulfill_loc_type = 'S'.
5	NULL	Store	Pickup in Store	Reservation. FulfilOrdDesc will contain: Single-leg: source_loc_type = NULL, fulfill_loc_type = 'S'.
6	NULL	Store	Ship to Customer	Reservation. FulfilOrdDesc will contain: Single-leg: source_loc_type = NULL, fulfill_loc_type = 'S'.
7	NULL	Warehouse	Ship to Customer	Virtual WH to Virtual Store Transfer. FulfilOrdDesc will contain: Single-leg: source_loc_type = 'WH', fulfill_loc_type = 'V'.
8	Vendor	Store	Pickup in Store	Purchase Order to Physical Store + Reservation. FulfilOrdDesc will contain: 1st leg: source_loc_type = 'SU', fulfill_loc_type = 'S' 2nd leg: source_loc_type = NULL, fulfill_loc_type = 'S'.
9	Vendor	Store	Ship to Customer	Purchase Order to Physical Store+ Reservation. FulfilOrdDesc will contain: 1st leg: source_loc_type = 'SU', fulfill_loc_type = 'S' 2nd leg: source_loc_type = NULL, fulfill_loc_type = 'S'.
10	NULL	Vendor	Ship to Customer	Purchase Order to Virtual Store FulfilOrdDesc will contain: Single-leg: source_loc_type = 'SU', fulfill_loc_type = 'V'.

The customer order subscription API supports create and cancel operations using the following message types belonging to the 'fulfilord' message family:

- **fulfilordapprdel** - used by Merchandising to cancel customer orders.
- **fulfilordreqdel** - used by SIOCS to request a customer order cancellation. This message type is used only by SIOCS and is ignored by Merchandising.
- **fulfilordpocre** - used to create purchase orders because of customer order fulfillment requests.

- **fulfilordtsfcre** - used to create transfers because of customer order fulfillment requests.
- **fulfilordstdlvcre** - used to perform inventory reservation because of customer order fulfillment requests.

The format for creating customer order fulfillment requests is shown below.

Table 3-11 Create Customer Order Fulfillment

Message Element	Required?	Notes
Customer Order Number	Yes	This field contains the master customer order number from OMS.
Fulfillment Order Number	Yes	This field contains the unique number from OMS related to the fulfillment details. A single customer order can have one or more fulfillment orders.
Source Location Type	No	This field contains the location type where the fulfillment order will be sourced. This could be either SU for supplier, ST for store, or WH for warehouse. This would only be populated for vendor, warehouse, or multi-site fulfillment orders. Both source location type and source location ID must be populated, or both should be NULL.
Source Location ID	No	This field contains the store, supplier or warehouse number associated with sourcing the fulfillment order. This would only be populated for vendor, warehouse, or multi-site fulfillment orders. Both source location type and source location ID must be populated, or both should be NULL. If the source location type is supplier, this must be a valid supplier site in Merchandising. If the source location type is store, this should be a valid stockholding customer orderable company or franchise store. If the source location type is warehouse, it can be a valid physical or virtual warehouse.
Fulfillment Location Type	No	This field indicates the location type associated with fulfilling the fulfillment order. This would be either S (for physical store) or V (for virtual store).
Fulfillment Location ID	Yes	This field indicates the store number associated with fulfilling the fulfillment order. This should always be populated with a virtual or physical store number. The fulfillment location ID should be different from the source location ID.
Partial Delivery Indicator	Yes	This field indicates if the fulfillment order can be picked and shipped partially (N) or if it should be shipped only when complete (Y).
Delivery Type	No	This field indicates the fulfillment method - ship to customer or store pickup. Valid values are S (ship direct) and C (customer pickup).
Carrier Code	No	This field indicates the carrier the order is to be shipped with, if specified on the fulfillment order.

Table 3-11 (Cont.) Create Customer Order Fulfillment

Message Element	Required?	Notes
Carrier Service Code	No	This field indicates the method that was selected for shipping by the customer placing the order (for example, Standard Shipping, Overnight, and so on). Valid values are from code type CSVG.
Consumer Delivery Date	Yes	This field indicates the desired date the delivery is required by the customer. This will be in GMT time.
Consumer Delivery Time	No	This field indicates the desired time the delivery is required by the customer. This will be in GMT time. Both Delivery Date and Delivery Time should be populated, or both should be NULL.
Delivery Charges	No	This field contains the delivery charges on drop ship. Used for Brazil implementations only. This value should be greater than 0.
Delivery Charges Currency	No	This field contains the currency of the delivery charges. It must be a valid currency code in Merchandising. Delivery Charges and Delivery Charges Currency must both be populated, or both should be NULL.
Comments	No	This field contains any comments sent by OMS about the order.
Customer Order Fulfillment Details	Yes	Child Node
Fulfillment Order Customer Details	Conditional	Child node
Flex Attributes	No	Child node Not used in Merchandising
Order Placed Store	No	This field indicates the store number associated with the location that the customer order was placed. For on-line orders this would contain the virtual store number associated with the on-line store. For orders captured in a physical store this would contain the store number for the physical store.

Table 3-12 Create Customer Order Fulfillment Details

Message Element	Required?	Notes
Item	Yes	This field indicates the item ordered by the customer.
Reference Item	No	This field indicates the reference item (barcode) ordered by the customer. This is supported for vendor drop-ships orders only.
Order Quantity SUOM	Yes	This field contains the quantity that was ordered by the customer in item's standard unit of measure.
Standard UOM	No	This field contains the item's standard unit of measure.
Transaction UOM	No	This field indicates the original transaction unit of measure the order was placed in.

Table 3-12 (Cont.) Create Customer Order Fulfillment Details

Message Element	Required?	Notes
Substitute Indicator	Yes	This field indicates if substitutes are allowed on a fulfillment order. This will only be used by orders passed to SIOCS.
Unit Retail	No	This field contains the unit sales retail of item on the fulfillment order. This will only be used by Brazil orders in case of warehouse fulfillment or vendor sourced POs shipped directly to the customer. It is needed for sales nota fiscal generation.
Retail Currency	No	This field contains the currency for the unit retail. This will only be used by Brazil orders in case of warehouse fulfillment or vendor sourced POs shipped directly to the customer. It is needed for sales nota fiscal generation.
Comments	No	This field is used to indicate any special instructions for the item, such as services (monograms, engrave, etc.).
Item Line Number	No	This field contains the detail item line number on the fulfillment order.

Only one customer order detail record should exist for each fulfillment order. If the system options value Retain Customer Data is set to Y and the order is fulfilled from the supplier or warehouse with the fulfillment location type as virtual store, then this node is required. If Retain Customer Data is N, then only the customer number will be retained.

Table 3-13 Create Fulfillment Order Customer Details

Message Element	Required?	Notes
Customer Number	Yes	This field indicates the number that uniquely identifies the customer in OMS.
Delivery First Name	No	This field contains the first name of the contact person at the delivery address on the fulfillment order.
Delivery Phonetic First	No	This field contains the phonetic first name of the contact person at the delivery address on the fulfillment order.
Delivery Last Name	No	This field contains the last name of the contact person at the delivery address on the fulfillment order.
Delivery Phonetic Last	No	This field contains the phonetic last name of the contact person at the delivery address on the fulfillment order.
Delivery Preferred Name	No	This field contains the preferred name of the contact person at the delivery address on the order.
Deliver Company Name	No	This field contains the company name of the contact person at the delivery address on the fulfillment order.
Deliver Address 1	No	This field contains the first line of the delivery address of the customer.

Table 3-13 (Cont.) Create Fulfillment Order Customer Details

Message Element	Required?	Notes
Delivery Address 2	No	This field contains the second line of the delivery address of the customer.
Delivery Address 3	No	This field contains the third line of the delivery address of the customer.
Delivery Country	No	This field contains the county portion of the delivery address.
Delivery City	No	This field contains the city portion of the delivery address.
Delivery State	No	This field contains the state portion of the delivery address.
Delivery Country ID	No	This field contains the country portion of the delivery address.
Delivery Post	No	This field contains the postal code portion of the delivery address.
Delivery Jurisdiction	No	This field identifies the jurisdiction code of the delivery country-state relationship.
Delivery Phone	No	This field contains the delivery phone number.
Delivery E-mail	No	This field contains the delivery email.
Bill First Name	No	This field contains the first name of the customer to be billed for this fulfillment order.
Bill Phonetic First	No	This field contains the phonetic first name of the customer to be billed for this fulfillment order.
Bill Last Name	No	This field contains the last name of the customer to be billed for this fulfillment order.
Bill Preferred Name	No	This field contains the preferred name of the customer to be billed for this fulfillment order.
Bill Company Name	No	This field contains the company name of the customer to be billed for this fulfillment order.
Bill Address 1	No	This field contains the first line of the billing address of the customer.
Bill Address 2	No	This field contains the second line of the billing address of the customer.
Bill Address 3	No	This field contains the third line of the billing address of the customer.
Bill Country	No	This field contains the county portion of the billing address.
Bill City	No	This field contains the city portion of the billing address.
Bill State	No	This field contains the state portion of the billing address.
Bill Country ID	No	This field contains the country portion of the billing address.
Bill Post	No	This field contains the postal code portion of the billing address.

Table 3-13 (Cont.) Create Fulfillment Order Customer Details

Message Element	Required?	Notes
Bill Jurisdiction	No	This field identifies the jurisdiction code for the billing country-state relationship.
Bill Phone	No	This field contains the billing phone number.
Bill E-mail	No	This field contains the billing e-mail address.

The format for cancelling customer order fulfillment requests is shown below.

Table 3-14 Cancel Customer Order Fulfillment

Message Element	Required?	Notes
Customer Order Number	Yes	Holds the master customer order number from OMS.
Fulfillment Order Number	Yes	This field contains the unique number from OMS related to the fulfillment details. A single customer order can have one or more fulfillment orders.
Source Location Type	No	This field contains the original sourcing location type for the fulfillment order being cancelled. This would be either SU for supplier, ST for store, or WH for warehouse. This would only be populated for vendor, warehouse, or multi-site fulfillment orders.
Source Location ID	No	This field contains the store, supplier or warehouse number associated with sourcing the fulfillment order being cancelled. This would only be populated for vendor, warehouse, or multi-site fulfillment orders.
Fulfillment Location Type	No	This field indicates the location type associated with fulfilling the fulfillment order being cancelled. This would be either S (for physical store) or V (for virtual store).
Fulfillment Location ID	Yes	This field indicates the store number associated with fulfilling the fulfillment order being cancelled.
Fulfillment Order Detail	Yes	Child node

Table 3-15 Cancel Customer Order Fulfillment Detail

Message Element	Required?	Notes
Item	Yes	This field contains the item cancelled by the customer.
Reference Item	No	This field contains the reference item cancelled by the customer. This is supported for vendor drop-ships orders only.
Cancel Quantity SUOM	Yes	This field contains the quantity that should be cancelled from the order in item's standard unit of measure.
Standard UOM	No	This field contains the item's standard unit of measure.

Table 3-15 (Cont.) Cancel Customer Order Fulfillment Detail

Message Element	Required?	Notes
Transaction UOM	No	This field indicates the original transaction unit of measure the order is placed in.
Item Line Number	No	This field indicates the detail item line number on the fulfillment order that is being cancelled.

In a RIB implementation, once fulfillment create messages are processed in Merchandising, Merchandising will publish to the RIB a customer order fulfillment confirmation message with a message type of 'fulfilordcfmcre' via the customer order fulfillment confirmation publishing API. Confirmation messages will only be sent for customer order fulfillment creates requests that result in creating purchase orders and transfers in Merchandising. It will not be sent for cancel requests, or for customer order fulfillment requests that result in inventory reservation.

- If a customer order is partially fulfilled, a confirmation message with status 'P' will be sent with details of fulfilled order quantity.
- If a customer order is not fulfilled at all due to unavailable inventory, a confirmation message with status 'X' will be sent without any details.
- If a customer order is fulfilled completely due to available inventory, a confirmation message with status 'C' will be sent with details for the fulfilled order quantity.

See [Customer Order Fulfillment Confirmation Publication API](#) for more details on the confirmation message sent.

Error Handling

If any errors are encountered in the validations described above or any of the message structure validations, a status of E is returned to the external system along with the appropriate error message. If the message has been successfully persisted, a success status (S), is returned to the external system indicating that the message has been successfully received and persisted to the Merchandising database.

Message XSD

Here are the filenames that correspond with each message type. Please consult the RIB documentation for each message type in order to get a detailed picture of the composition of each message.

Message Types	Message Type Description	XML Schema Definition (XSD)
Fulfilordapprdel	Fulfilment Cancel Message	FulfilOrdRef.xsd
Fulfilordreqdel	Fulfilment Cancel Request Message	FulfilOrdRef.xsd
Fulfilordpocre	Fulfilment PO Create Message	FulfilOrdDesc.xsd
Fulfilordtsfcre	Fulfilment Transfer Create Message	FulfilOrdDesc.xsd
Fulfilordstdlvcre	Fulfilment Store Delivery Create Message	FulfilOrdDesc.xsd

Differentiator Group Subscription API

This section describes the Differentiator group subscription API.

Functional Area

Items

Business Overview

This API allows external systems to create, edit, and delete differentiator groups within Merchandising. Diff ID details can be added, edited, or deleted within the diff group message. When creating a new diff group, diff ID must be included, but they can also be passed in with their own specific message type. Diff ID detail create and modify messages must also include the diff group record.

Creating Diff Groups

When a new differentiator group is created, this API will first validate that all required fields are present in the message. When creating a new differentiator group at least one detail line must also be included in the message. After that, business level validation on the input information will be performed. The tables below summarize the validation.

Table 3-16 Diff Group Header

Message Element	Required?	Notes
Differentiator Group Identifier	Yes	This field contains the unique differentiator group identifier.
Differentiator Type	Yes	This field contains the differentiator type, such as C for color. Must exist as a valid diff type in Merchandising.
Differentiator Group Description	Yes	This field contains the description of the differentiator group.
Create Date Time	No	This field contains the date and time the differentiator group was created. If it is not populated on the subscription message it will be defaulted to the time of creation in Merchandising.
Differentiator Group Detail	Conditional	Child node This is required in a diff group detail create and modify message.

Table 3-17 Diff Group Detail

Message Element	Required?	Notes
Differentiator ID	Yes	This field contains the identifier of the differentiator contained within the differentiator group. This id must be unique within the diff group and must already exist in Merchandising.
Display Seq	No	This field contains the order in which the diff ID should appear within the differentiator group, when displayed on-line.
Create Date Time	No	This field contains the date/time the differentiator ID was added to the differentiator group. If it is not populated on the subscription message it will be defaulted to the time of creation in Merchandising.

Updating Diff Groups

When updating a differentiator group, the group ID must already be present in the Merchandising. Changes can be sent for header level updates or detail level updates. If the changes are at the header level, then all of the required header level information needs to be included in the update, similar to that described above for creating a new differentiator. However, the diff details should not be included in a header only update. Fields that can be updated at the header level using this API include:

- Differentiator type
- Differentiator group description

For updating the record, the diff group ID is required in the header level and diff ID is required at the detail level. Fields that can be updated at the detail level using this API include:

- Display seq

Deleting Diff Groups

If you are deleting a differentiator detail in the differentiator group or deleting the whole differentiator group, then the API will validate that the differentiator group is valid and that it is not associated with any items or diff ranges. If you are deleting the whole differentiator group, then no details should be included in the message. If you are deleting a detail record on the differentiator, then validation will be done to ensure that the diff ID exists on the differentiator group.

Table 3-18 Diff Group Header

Message Element	Required?	Notes
Differentiator Group Identifier	Yes	This field contains the diff group to be deleted.
Differentiator Group Detail	Conditional	Child node This is required in a diff group detail delete message.

Table 3-19 Diff Group Detail

Message Element	Required?	Notes
Differentiator ID	Yes	This field contains the identifier of the differentiator that will be deleted, that is contained within the differentiator group. This id must be unique within the diff group and must already exist in Merchandising.

Error Handling

If any errors are encountered in the validations described above or any of the message structure validations, a status of E is returned to the external system along with the appropriate error message. If the message has been successfully persisted, a success status (S), is returned to the external system indicating that the message has been successfully received and persisted to the Merchandising database.

Message XSD

Below are the filenames that correspond with each message type. Please consult the Oracle Retail Integration Guide for each message type for the details on the composition of each message.

Message Types	Message Type Description	XML Schema Definition (XSD)
XDiffGrpDes	Diff Group Create and Modify	XDiffGrpDesc.xsd
XDiffGrpDtl	Diff Group Detail Create and Modify	XDiffGrpDesc.xsd
XDiffGrpRef	Diff Group Delete	XDiffGrpRef.xsd
XDiffGrpDtlRef	Diff Group Detail Delete	XDiffGrpRef

Differentiator Subscription API

This section describes the Diff ID subscription API.

Functional Area

Items

Business Overview

This API subscribes to differentiators from external systems to create, update or delete differentiators in Merchandising. This subscription API provides a means to keep Merchandising in sync with an external system. These transactions are performed immediately upon message receipt so success or failure can be communicated to the calling application.

Creating Differentiators

When a new differentiator is created, this API will first validate that all required fields are present in the message. After that, business level validation on the input information will be performed. The business validation:

- verifies the diff id does not contain white space or underscores
- verifies if diff id is not already present as a diff id or diff group id
- verifies the diff type is a valid value on the code detail table under code type DIFF

If all the validations are met, the differentiator in the message data is created in Merchandising.

The format for creating and updating differentiators is shown below.

Table 3-20 Differentiator

Message Element	Required?	Notes
Differentiator ID	Yes	This field contains the unique identifier of the differentiator. This must be a valid value under the code type DIFF.
Differentiator Type	Yes	This field contains the identifier of the differentiator type. This value must be a valid diff_type in merchandising.
Differentiator Description	Yes	This field contains the description of the differentiator.
Industry Code	No	This field contains the unique reference number which represents all possible combinations of sizes according to the National Retail Federation.
Industry Subgroup	No	This field contains a sub-grouping code used by industry standards to further identify the differentiator.
Create Date Time	No	This contains the date and time the differentiator was created. If this field is not populated on the message it will default to the time of creation in merchandising.

Updating Differentiators

When a differentiator is updated, this API will first validate that all required fields are present in the message. After that, business level validation on the input information will be performed. If all the validations are met, the differentiator in the message data is updated in Merchandising. The message and validation is similar to that of the creating differentiators.

Deleting Differentiators

When a differentiator is deleted, this API will first validate that all required fields are present in the message. After that, business level validation on the input information will be performed. The business validation:

- verifies if diff id to be deleted exists in Merchandising

If all the validations are met, the differentiator in the message data is deleted from Merchandising.

Table 3-21 Differentiator

Message Element	Required?	Notes
Differentiator ID	Yes	This field contains the unique identifier of the differentiator being deleted.

Error Handling

If any errors are encountered in the validations described above or any of the message structure validations, a status of E is returned to the external system along with the appropriate error message. If the message has been successfully persisted, a success status (S), is returned to the external system indicating that the message has been successfully received and persisted to the Merchandising database.

Message XSD

Here are the filenames that correspond with each message type. Please consult RIB documentation for each message type in order to get a detailed picture of the composition of each message.

Message Type	Message Type Description	XML Schema Definition (XSD)
xdiffidcre	Differentiator Create	XDiffIDDesc.xsd
xdiffidmod	Differentiator Modify	XDiffIDDesc.xsd
xdiffiddel	Differentiator Delete	XDiffIDRef.xsd

DSD Deals Subscription API

This section describes the DSD deals subscription API.

Functional Area

Deals

Business Overview

Direct store delivery (DSD) is the delivery of merchandise and/or services to a store without the benefit of a pre-approved purchase order, such as when the supplier drops off merchandise directly in the retailer's store. This process is common in convenience and grocery stores, where suppliers routinely come to restock merchandise. In these cases, the invoice may be given to the store (as opposed to sent to corporate), and the invoice may or may not be paid for out of the register.

Merchandising subscribes to DSD messages from the RIB. These messages notify Merchandising of a direct store delivery transaction at a location so that it may record the purchase order and account for it in the store's inventory. Merchandising also subscribes to DSD deals messages for deals applicable to any DSD order and performs the following functionalities as necessary:

- Applies any deals to a DSD purchase order if the deals indicator in the message is set to Y
- Creates a shipment
- Receives a shipment.
- Creates an invoice

Table 3-22 DSD Deals

Message Element	Required?	Notes
Order Number	Yes	This field contains the order number.
Supplier	Yes	This field contains the supplier number for the deal being created for this order, not the supplier site.
Store	Yes	This field contains the location the shipment will be delivered to.
Department	No	The department in which all the items on the order belong.
Currency Code	Yes	This field contains a code identifying the currency the supplier uses for business transactions.
Paid Indicator	Yes	This field indicates if the invoice has already been paid. Valid values are Y (invoice has already been paid) or N (invoice should be paid in accounts payable system).
External Reference Number	No	If the invoice indicator is Y (invoice has been created), the external reference number, proof of delivery number, or payment reference number must be provided.
Proof of Delivery Number	No	This field contains the proof of delivery or service number given at the time of receipt at the store. This field will also be included when the invoice is interfaced through Sales Audit.
Payment Reference Number	No	This field contains the reference number attached to the invoice payment, used when the invoice is paid from the POS system and interfaced through Sales Audit.
Payment Date	No	This field contains the date when the invoice was paid from the POS system. This field will be populated when the invoice is interfaced through Sales Audit.
Deals Indicator	Yes	This field indicates whether deals need to be applied to the order or not. Valid values are Yes (Y) and No (N).
Shipment	Yes	This field contains the corresponding shipment for the order that was applied by the deal.
Invoice Id	Yes	This field contains the invoice number for the purchase order of this deal.
Invoice Indicator	Yes	This field indicates whether an invoice was created for this receipt by the supplier. Valid values are Yes (Y) and No (N).
Receipt Date	Yes	This field contains the date of the receipt.
Quantity Sum	Yes	This field contains the total quantity for the invoice.
Cost Sum	Yes	This field contains the total merchandise cost for the invoice. This field will be held in the invoice currency.

Table 3-22 (Cont.) DSD Deals

Message Element	Required?	Notes
External Receipt Number	Yes	This field holds the external transaction sequence number for the receipt.

**Note:**

Invoices are not created if Invoice Matching is not running, if the invoice indicator or paid indicator from the message is N, or if paid indicator on the message is Y and Sales Audit is not running.

Error Handling

If any errors are encountered in the validations described above or any of the message structure validations, a status of E is returned to the external system along with the appropriate error message. If the message has been successfully persisted, a success status (S), is returned to the external system indicating that the message has been successfully received and persisted to the Merchandising database.

Message XSD

Here are the filenames that correspond with each message type. Please see RIB documentation for each message type in order to get a detailed picture of the composition of each message.

Message Types	Message Type Description	XML Schema Definition (XSD)
dsddealscre	DSD Deals Create Message	DSDDealsDesc.xsd

DSD Receipt Subscription API

This section describes the DSD receipt subscription API.

Functional Area

Purchase Orders

Business Overview

Direct store delivery (DSD) is the delivery of merchandise to and/or the performance of services in a store without the benefit of a pre-approved purchase order. When the delivery or service occurs, the store inventory system (for example, SIOCS) informs Merchandising of the receipt, which also generates the purchase order at the same time. The receipt can include both the merchandise item as well as the non-merchandise information associated with the order. This information works in

conjunction with payment details sent via Sales Audit if payment was made a part of the delivery or service at the store.

Table 3-23 DSD Receipt Header

Message Element	Required?	Notes
Supplier	Yes	This field contains the unique identifying number for a supplier for the receipt.
Origin Country ID	Yes	This field contains the identifier of the country from which the item is being sourced.
Store	Yes	This field contains the location where the items were delivered.
Department	Conditional	The department in which all the items on the order belong.
Currency Code	Yes	This field contains a code identifying the currency the supplier uses for business transactions.
Paid Indicator	Yes	This field indicates if the invoice has already been paid. Valid values are Y (invoice has already been paid) or N (invoice should be paid in accounts payable system).
External Reference Number	No	If the invoice indicator is Y (invoice has been created), the external reference number, proof of delivery number, or payment reference number must be provided.
Proof of Delivery Number	No	This field contains the proof of delivery or service number given at the time of receipt at the store. This field will also be included when the invoice is interfaced through Sales Audit.
Payment Reference Number	No	This field contains the reference number attached to the invoice payment, used when the invoice is paid from the POS system and interfaced through Sales Audit.
Payment Date	No	This field contains the date when the invoice was paid from the POS system. This field will be populated when the invoice is interfaced through Sales Audit.
Invoice Indicator	Yes	This field indicates whether an invoice was created for this receipt by the supplier. Valid values are Yes (Y) and No (N).
Deals Indicator	Yes	This field indicates whether deals need to be applied to the order or not. Valid values are Yes (Y) and No (N).
DSD Detail	No	Child node
DSD Non-Merchandise	No	Child node
External Receipt Number	No	This field holds the external transaction sequence number for the receipt.
Receipt Date	No	This field contains the date of the receipt.

Table 3-24 DSD Receipt Detail

Message Element	Required?	Notes
Item	Yes	This field contains the item in the receipt. The item must be an approved item that is at transaction level or above.

Table 3-24 (Cont.) DSD Receipt Detail

Message Element	Required?	Notes
Quantity Received	Yes	This field contains the number of items received for the item/shipment combination. The value must be greater than 0.
Unit Cost	No	This field contains the cost of the item from the supplier/origin country in the supplier's currency.
Weight	No	This field contains the weight of the item in the receipt, if it is a catch weight item.
Weight UOM	No	This field contains the unit of measure of the received weight.
DSD Detail UIN	No	Child node; not used in Merchandising

Table 3-25 DSD Non-Merchandise

Message Element	Required?	Notes
Non-Merchandise Code	Yes	This field contains the non-merchandising code that will be added to an invoice for services or other non-merchandise costs associated with the order.
Non-Merchandise Amount	Yes	This field contains the amount of the non-merchandise cost that was invoiced. This field will be held in the invoice currency.
VAT Code	No	This field contains the code identifying the VAT rate that should be applied to the non-merchandise amount entered.
Service Performed Indicator	Yes	This field indicates if a service was performed for the non-merchandise cost. Valid values are yes (Y) or no (N).

Error Handling

If any errors are encountered in the validations described above or any of the message structure validations, a status of E is returned to the external system along with the appropriate error message. If the message has been successfully persisted, a success status (S), is returned to the external system indicating that the message has been successfully received and persisted to the Merchandising database.

Message XSD

Here are the filenames that correspond with each message type. Please consult the RIB documentation for each message type to get a detailed picture of the composition of each message.

Message Types	Message Type Description	XML Schema Definition (XSD)
dsdreceiptcre	DSD Receipt Create Message	DSDReceiptDesc.xsd
dsdreceiptmod	DSD Receipt Modify Message	DSDReceiptDesc.xsd

Freight Term Subscription API

This section describes the freight terms subscription API.

Functional Area

Foundation Data

Business Overview

Freight terms are financial arrangement information related to shipping that can be subscribed to by Merchandising from a financial system. Freight terms are the terms for shipping - for example, the freight terms could be a certain percentage of the total cost, a flat fee per order, and so on. After confirming the validity of the records enclosed within the message, Merchandising is updated with the information.

Creating/Updating Freight Terms

When a new freight term is created, this API will first validate that all required fields are present in the message. Required fields are terms, description, enabled flag, and start and end dates. After required field validation, the freight term record in the message will be inserted if the term does not exist. If the freight term exists, then the dates and enabled flag will be updated.

Message Element	Required?	Notes
Freight Terms	Yes	Contains a number that uniquely identifies the freight terms.
Terms Description	Yes	Contains a description of the freight terms used in the system.
Enabled Flag	Yes	Indicates whether the freight terms are valid or invalid within the respective application. Valid values are Y and N.
Active Start Date	Yes	Indicates the date for assigning an active date to the Freight Terms.
Active End Date	Yes	Indicates the date for assigning an inactive date to the Freight Terms.

Error Handling

If an error occurs in this procedure, a call will be placed to a function to build a complete error message. This message together with a status of E is returned to the external system. If the message has been successfully persisted, a success status (S), is returned to the external system indicating that the message has been successfully received and persisted to the Merchandising database.

Message XSD

Below are the filenames that correspond with each message type. Please consult the RIB documentation for each message type in order to get a detailed picture of the composition of each message.

Message Types	Message Type Description	XML Schema Definition (XSD)
FrtTermCre	Freight Term Create Message	FrtTermDesc.xsd

General Ledger Chart of Accounts Subscription API

This section describes the GL chart of accounts subscription API.

Functional Area

Financial Integration

Business Overview

This API is used when Retail Financial Integration (RFI) is not being used and integration with a financial system is a custom or to a non-Oracle financials system. When using RFI with an Oracle Financial system the Chart of Accounts Information in Merchandising and Sales Audit is created through a mapping and COA validation process.

Before Merchandising can publish stock ledger data to an external financial application, it must receive the General Ledger chart of accounts (GLCOA) structure. The chart of accounts is the financial application's debit and credit account segments (for example, company, cost center, account, and so on). These are mapped to the transactions, locations, and product hierarchy in Merchandising when stock ledger data and Sales Audit totals are sent to the General Ledger. In some financial applications, these are known as code combination chart fields. There is also a primary account, in some systems known as a CCID that uniquely identifies the combination of segment or chart field values. Upon receipt of GLCOA message data, Merchandising populates the data to the FIF_GL_ACCT table.

Create and Update Chart of Accounts

This message is used to create new chart of account entries, as well as update existing entries. The message payload details are shown below.

Table 3-26 Chart of Accounts Create

Message Element	Required?	Notes
Primary Account	Always	This field denotes the primary account for a chart of accounts.
Attribute 1	Optional	Secondary account information. A value is required if description 1 is supplied.

Table 3-26 (Cont.) Chart of Accounts Create

Message Element	Required?	Notes
Attribute 2	Optional	Secondary account information. A value is required if description 2 is supplied.
Attribute 3	Optional	Secondary account information. A value is required if description 3 is supplied.
Attribute 4	Optional	Secondary account information. A value is required if description 4 is supplied.
Attribute 5	Optional	Secondary account information. A value is required if description 5 is supplied.
Attribute 6	Optional	Secondary account information. A value is required if description 6 is supplied.
Attribute 7	Optional	Secondary account information. A value is required if description 7 is supplied.
Attribute 8	Optional	Secondary account information. A value is required if description 8 is supplied.
Attribute 9	Optional	Secondary account information. A value is required if description 9 is supplied.
Attribute 10	Optional	Secondary account information. A value is required if description 10 is supplied.
Attribute 11	Optional	Secondary account information. A value is required if description 11 is supplied.
Attribute 12	Optional	Secondary account information. A value is required if description 12 is supplied.
Attribute 13	Optional	Secondary account information. A value is required if description 13 is supplied.
Attribute 14	Optional	Secondary account information. A value is required if description 14 is supplied.
Attribute 15	Optional	Secondary account information. A value is required if description 15 is supplied.
Description 1	Optional	Description of the attribute 1 field. Required if attribute 1 is supplied.
Description 2	Optional	Description of the attribute 2 field. Required if attribute 2 is supplied.
Description 3	Optional	Description of the attribute 3 field. Required if attribute 3 is supplied.
Description 4	Optional	Description of the attribute 4 field. Required if attribute 4 is supplied.
Description 5	Optional	Description of the attribute 5 field. Required if attribute 5 is supplied.
Description 6	Optional	Description of the attribute 6 field. Required if attribute 6 is supplied.
Description 7	Optional	Description of the attribute 7 field. Required if attribute 7 is supplied.
Description 8	Optional	Description of the attribute 8 field. Required if attribute 8 is supplied.

Table 3-26 (Cont.) Chart of Accounts Create

Message Element	Required?	Notes
Description 9	Optional	Description of the attribute 9 field. Required if attribute 9 is supplied.
Description 10	Optional	Description of the attribute 10 field. Required if attribute 10 is supplied.
Description 11	Optional	Description of the attribute 11 field. Required if attribute 11 is supplied.
Description 12	Optional	Description of the attribute 12 field. Required if attribute 12 is supplied.
Description 13	Optional	Description of the attribute 13 field. Required if attribute 13 is supplied.
Description 14	Optional	Description of the attribute 14 field. Required if attribute 14 is supplied.
Description 15	Optional	Description of the attribute 15 field. Required if attribute 15 is supplied.
Set of Books	Always	Indicates the set of books that these accounts apply to.

Error Handling

If any errors are encountered in the validations described above or any of the message structure validations, a status of E is returned to the external system along with the appropriate error message. If the message has been successfully persisted, a success status (S), is returned to the external system indicating that the message has been successfully received and persisted to the Merchandising database.

Message XSD

Below are the filenames that correspond with each message type. Please consult the *Oracle Retail Integration Guide* for each message type for the details on the composition of each message.

Message Types	Message Type Description	XML Schema Definition (XSD)
Glcoacre	GL COA Create Message	GLCOADesc.xsd

Inbound ASN Subscription API

This section describes the ASNIN subscription API.

Functional Area

Shipping and Receiving

Business Overview

A supplier or consolidator will send an advanced shipping notice (ASN) to Merchandising. Merchandising subscribes to the ASN information and places the information onto Merchandising tables depending upon the validity of the records enclosed within the ASN message.

The ASN message will consist of a header record, a series of order records, carton records, and item records. For each message, header, order, and item records will be required. The carton portion of the record is optional. If a carton record is present, however, then that carton record must contain items.

The header record will contain information about the shipment, such as where it is being shipped and when it is estimated to arrive. The order records will identify which purchase orders are associated with the shipment. If the shipment is packed in cartons, carton records will identify which items are in which cartons. The item records will contain the items on the shipments, along with the quantity shipped.

The location that is contained on the ASN will represent the expected receiving location for the order. If the location is a non-stockholding store in Merchandising, then the shipment will also be automatically received when the ASN is processed. Two types of non-stockholding stores orders are supported in this integration – franchise stores and drop ship customer orders.

Once the ship quantity is matched, an invoice is generated for Invoice Matching.



Note:

This message can also be used by stores and warehouses for inbound transfer and allocation shipments. However, for Merchandising, those shipments are all processed as an Outbound ASN.

Creating/Updating ASN

When a new ASN is created or updated, this API will first validate that all required fields are present in the message. Additionally, when creating or updating an ASN at least one detail line must also be included in the message. After that, business level validation on the input information will be performed. The tables below summarize these two types of validation.

ASN Header

Message Element	Required?	Notes
Schedule Number	Optional	Not used in Merchandising.
Auto Receive	Optional	Not used in Merchandising.
Destination Location	Optional	Contains the location that the shipment will be delivered to. For purchase orders this will always be either a store or a physical warehouse.
Destination Location Type	Optional	This column contains the destination location type of the destination location field. Valid values are 'S' = store, and 'W' = warehouse.
Destination Store Type	Optional	Not used in Merchandising

Message Element	Required?	Notes
To Stockholding Indicator	Optional	Not used in Merchandising
Source Location	Optional	Not used in Merchandising
Source Location Type	Optional	Not used in Merchandising
Source Location Store Type	Optional	Not used in Merchandising
From Stockholding Indicator	Optional	Not used in Merchandising
Advance Shipment Number	Always	This column contains the advance shipping notice number associated with the shipment.
Shipment Type	Always	This column is used to determine the ship origin. If it is C, that means it is a carton shipment and the shipment origin in Merchandising is set to 6 (ASN UCC-128). Otherwise, the Merchandising shipment origin will be defaulted to 0 (ASN Shipment).
Container Quantity	Optional	Not used in Merchandising.
Bill of Lading Number	Optional	This column holds the bill of lading number associated with a shipment from the PO receiving process.
Shipment Date	Always	This column contains the date the PO was shipped.
Estimated Arrival Date	Optional	This field contains the estimated arrival date of a vendor PO shipment.
Ship Address 1	Optional	Not used in Merchandising.
Ship Address 2	Optional	Not used in Merchandising.
Ship Address 3	Optional	Not used in Merchandising.
Ship Address 4	Optional	Not used in Merchandising.
Ship Address 5	Optional	Not used in Merchandising.
Ship City	Optional	Not used in Merchandising.
Ship State	Optional	Not used in Merchandising.
Ship Postal Code	Optional	Not used in Merchandising.
Ship Country	Optional	Not used in Merchandising.
Trailer Number	Optional	Not used in Merchandising.
Seal Number	Optional	Not used in Merchandising.
Carrier Code	Optional	This column contains a code that indicates the carrier that is involved in the shipment.
Carrier Service Code	Optional	Not used in Merchandising.
Vendor Number	Always	This column contains the supplier who will provide the merchandise specified in the order.
Ship payment method	Optional	This column indicates the payment terms for freight charges associated with the order. Valid values are found in code type SHMT.
Comments	Optional	Contains any comments about the shipment.

ASN PO

Message Element	Required?	Notes
Order Number	Always	Identifies the order number which relates to the goods delivered in the shipment.

Message Element	Required?	Notes
Document Type	Optional	Not used in Merchandising.
Not After Date	Optional	This column contains the last date that delivery of the order will be accepted.
Comments	Optional	This column contains any comments about the shipment.

ASN Carton Detail

This is an optional node inside the PO details. If the shipment type is C then the carton details and the items within the carton are validated.

Message Element	Required?	Notes
Final Location	Always	This will be the final destination of the carton. For a cross-dock order this will be the allocation location, otherwise it will be the direct to order location.
Container ID	Always	This column holds the UCC-128 carton number.
Container Weight	Optional	Not used in Merchandising.
Container Length	Optional	Not used in Merchandising.
Container Width	Optional	Not used in Merchandising.
Container Height	Optional	Not used in Merchandising.
Container Cube	Optional	Not used in Merchandising.
Expedite Flag	Optional	Not used in Merchandising.
In Store Date	Optional	Not used in Merchandising.
Carrier Shipment Number	Optional	Not used in Merchandising.
Tracking Number	Optional	Not used in Merchandising.
Freight Charge	Optional	Not used in Merchandising.
Master Container ID	Optional	Not used in Merchandising.

ASN Item Details

This is a mandatory node inside the PO and the optional Carton Node. For each message, header, order, and item records will be required. The carton portion of the record is optional. If a carton record is present, then that carton record must contain items in it.

Message Element	Required?	Notes
Final Location	Conditional	This column contains the final location for the order. If it is a pre-allocated order that has been pre-marked, then this is required and would contain the allocation location.
Item Number	Optional	This column contains unique identifier for the item. Either VPN, item number, or reference item must be specified in the message.
Unit Quantity	Always	This column contains the quantity of the item that is expected to be received.
Priority Level	Optional	Not used in Merchandising.

Message Element	Required?	Notes
VPN	Optional	This column contains the vendor product number used to find the item number. Either VPN, item number, or reference item must be specified in the message.
Order Line Number	Optional	Not used in Merchandising
Lot Number	Optional	Not used in Merchandising.
Reference Item Number	Optional	The column contains a bar code or reference item. Either VPN, item number, or reference item must be specified in the message.
Distro Number	Optional	Not used in Merchandising.
Consumer Direct Order	Optional	Not used in Merchandising.
Customer Order Number	Optional	Not used in Merchandising.
Fulfill Order Number	Optional	Not used in Merchandising.
Distro Document Type	Optional	Not used in Merchandising.
Container Quantity	Optional	Not used in Merchandising.
Comments	Optional	Not used in Merchandising.



Note:

Universal Identification Number (UIN) Details and Flex Attribute is not used by Merchandising.

Delete ASN

When deleting an ASN, Merchandising will validate that the shipment is not in received status. If in a valid status, the shipment will be deleted. When deleting the ASN records, if Invoice Matching is integrated, it will populate the staging table for Invoice Matching to know which shipments have been purged.

Message Element	Required?	Notes
Destination Location	Optional	Contains the location that the shipment will be delivered to. For purchase orders this will always be either a store or a physical warehouse.
Destination Location Type	Optional	This column contains the destination location type of the destination location field. Valid values are 'S' = store, and 'W' = warehouse.
Destination Store Type	Optional	Not used in Merchandising
To Stockholding Indicator	Optional	Not used in Merchandising
Source Location	Optional	Not used in Merchandising
Source Location Type	Optional	Not used in Merchandising
Source Store Type	Optional	Not used in Merchandising
Source Stockholding Indicator	Optional	Not used in Merchandising

Message Element	Required?	Notes
ASN Number	Always	This column contains the advance shipping notice number associated with the shipment. The delete happens based on the ASN number where the origin of shipping is through ASN shipment or through ASN UCC-128.

Error Handling

If any errors are encountered in the validations described above or any of the message structure validations, a status of E is returned to the external system along with the appropriate error message. If the message has been successfully persisted, a success status (S), is returned to the external system indicating that the message has been successfully received and persisted to the Merchandising database.

Message XSD

Below are the filenames that correspond with each message type. Please consult the Oracle Retail Integration Guide for each message type for the details on the composition of each message.

Message Types	Message Type Description	XML Schema Definition (XSD)
asnincre	ASN in Create	ASNInDesc.xsd
asninmod	ASN in Modification	ASNInDesc.xsd
asnindel	ASN in delete	ASNInRef.xsd

Inventory Adjustment Subscription API

This section describes the Inventory Adjustment Subscription API.

Functional Area

Inventory

Business Overview

Merchandising receives requests for inventory adjustments from an integration subsystem through the Inventory Adjustment Subscription API. The requests contain information about the item and location whose inventory is being adjusted, the quantity to adjust, a from and to disposition code, and the reason for the adjustment. Merchandising uses information in these requests to:

- Adjust overall quantities of stock on hand for an item at a location.
- Adjust the availability of item-location quantities based on status.

After initial processing and validation, Merchandising performs the following tasks:

- The item/location is ranged if it does not already exist.

- For total stock on hand adjustments:
 - Stock on hand is updated for the item at the location, for total stock on hand adjustments.
 - Stock adjustment is recorded to the Merchandising transaction level stock ledger
- For status-based adjustments:
 - Quantities by inventory status are adjusted for the item/location combination.
 - Non-sellable quantity is updated for the item/location.
- For both types, an audit trail is created for the inventory adjustment by item, location, inventory status and reason.

**Note:**

An adjustment can impact both total stock on hand and inventory status at the same time.

Inventory Adjustment Transaction Codes

Whenever the status or quantity of inventory changes, Merchandising writes transaction codes to adjust inventory values in the stock ledger. The types of inventory adjustment transaction codes are:

- Tran code 22 - Adjustments where positive and negative adjustments are made to total stock on hand using a reason code with the COGS indicator = N. In this case, a transaction is inserted to the transaction level stock ledger for both the retail and cost value of the adjustment.
- Tran code 23 - Adjustments where positive and negative adjustments are made to total stock on hand using a reason code with the COGS indicator = Y. In this case, a transaction is inserted to the transaction level stock ledger for both the retail and cost value of the adjustment.
- Tran code 25 - Adjustments to inventory status, where inventory is moved to or from an unavailable or non-sellable status.

Other Notes:

- One or both of the to disposition and from disposition fields must have values. Both cannot be empty.
- The item must be inventoried and approved.
- If the item is a simple pack catch weight item, then the weight and weight UOM are either both defined or both NULL. Weight UOM must be of the type Mass.
- The item is a transaction-level or a reference item. When a reference item is passed in, its parent item (the transaction level item) has its inventory adjusted.
- If adjusting a pack at a warehouse, the pack item must have its inventory tracked at the pack level (receive as type = Pack for the item/warehouse).
- If the location is a warehouse, then either a virtual or physical warehouse can be supported. If it is a virtual warehouse, it must be a stockholding warehouse. If it is

a physical warehouse, then the adjusted quantity is distributed among the virtual locations of the physical location.

The table below contains the details of the message as well as the validations.

Table 3-27 Inventory Adjustment Header

Message Element	Required?	Notes
Destination ID	Yes	This field contains the location where the inventory adjustment is being made.
Inventory Adjustment Detail	Yes	Child node

Table 3-28 Inventory Adjustment Detail

Message Element	Required?	Notes
Item	Yes	This contains the item for which stock is being adjusted. The item should be an approved inventory item.
Adjustment Reason Code	No	This field contains the reason for inventory adjustment. This field will only have a value for sellable inventory types.
Unit Quantity	Yes	This field contains the number of units to be added or reduced. The value should not be 0 and should be a whole number if the standard UOM of the item is EA.
Transshipment Number	No	Not used
From Disposition	Conditional	From Disposition and To Disposition should both have values or both should be NULL.
To Disposition	Conditional	From Disposition and To Disposition should both have values or both should be NULL.
From Trouble Code	No	Not used
To Trouble Code	No	Not used
From WIP Code	No	Not used
To WIP Code	No	Not used
Transaction Code	No	Not used
User ID	Yes	This field contains the name of the user who created the inventory adjustment.
Create Date Time	Yes	This field contains the date and time the inventory adjustment was made.
PO Number	No	This field contains either a PO, Allocation, BOL or Transfer number associated to the inventory adjustment based on the doc_type. The value should exist in Merchandising.
Document Type	No	This field indicates the type of document where the inventory adjustment originated from. Valid values are: <ul style="list-style-type: none"> • P - Purchase Order • T - Transfer • A - Allocation
Auxiliary Reason Code	No	Not used
Weight	Conditional	This contains the weight of the item. Weight and Weight UOM should both have values or both should be NULL.

Table 3-28 (Cont.) Inventory Adjustment Detail

Message Element	Required?	Notes
Weight UOM	Conditional	This is the unit of measurement for weight. Weight and Weight UOM should both have values or both should be NULL. This value should belong to the uom_class, MASS.
Unit Cost	No	This field contains the unit cost of the item.
Inventory Adjustment UIN	No	Child node

Table 3-29 Inventory Adjustment UIN

Message Element	Required?	Notes
Unique Identification Number	Yes	This field contains the Universal Identification Number of the item at the location.
Status	Yes	This field contains the status code of the UIN.

Error Handling

If any errors are encountered in the validations described above or any of the message structure validations, a status of E is returned to the external system along with the appropriate error message. If the message has been successfully persisted, a success status (S), is returned to the external system indicating that the message has been successfully received and persisted to the Merchandising database.

Message XSD

Below are the filenames that correspond with each message type. Please consult the Oracle Retail Integration Guide for each message type for the details on the composition of each message.

Message Types	Message Type Description	XML Schema Definition (XSD)
invadjustcre	Inventory Adjustment Create Message	InvAdjustDesc.xsd

Inventory Request Subscription API

This section describes the inventory request subscription API.

Functional Area

Inventory Request Subscription

Business Overview

Merchandising receives requests for inventory using the Inventory Request API, which allows for items to be ordered by the store and fulfilled by the Merchandising. Unlike

store order replenishment, Merchandising fulfills inventory requests from the store regardless of replenishment review cycles, delivery dates, and any other factors that may restrict a request from being fulfilled.

For item/store combinations that are on the Store Order type of replenishment in Merchandising, orders will be placed using this API and then the replenishment process builds the recommended order quantity (ROQ) based on the store's requests. Requests that will not be reviewed prior to the date requested by the store are fulfilled through a one-off process (executed real-time through this API) that creates warehouse transfers and/or purchase orders to fulfill the requested quantities.

For item/location combinations that are currently using other methods of replenishment in Merchandising, the store requested quantities will be added on top of the calculated recommended order quantities to increase the overall replenishment. It can also be used for item/store combinations not on replenishment in Merchandising. In these cases, the one-off process described above will be used to create a POs or transfers utilizing attributes defined for the item/location.

Other validation notes:

- Order quantities will be rounded using the store order multiple when an order is created for a warehouse or to the case size if ordering from the supplier.
- Upcharges will always be applied to a transfer, when they can be defaulted.
- Merchandising will validate that all items belong to the same department when department level ordering (supplier) or department level transfers (warehouse) are being used.
- The store must be open for ordering.

Creating Inventory Requests

The table below summarizes these validations applicable for this API.

Message Element	Required?	Notes
Item	Always	The item must be approved, orderable, and inventoried item; it must also be ranged to the location in the inventory request and must be active at that location.
Unit of purchase	Always	Unit of purchase must either be eaches (EA), case (CA), or pallet (PA).
Need date	Always	This is the date that the store needs the item by.
Need quantity	Always	This is the quantity being requested in standard UOM.
Delivery slot	Optional	Valid delivery slots are in the delivery_slot table.
Store	Always	The store must exist as a valid stockholding store in Merchandising.
Request type	Optional	Store order (SO) or Inventory Request (IR) If the request type is SO or blank, then replenishment method should be store order. If the request type is IR, delivery slot should be provided.

Error Handling

If an error occurs in this procedure, a call will be placed to a function to build a complete error message. This message together with a status of E is returned to the external system. If the message has been successfully persisted, a success status (S), is returned to the external system indicating that the message has been successfully received and persisted to the Merchandising database. This API supports non-fatal error processing. If an error is encountered in one inventory request detail, it will log and return the error to the RIB.

Message XSD

Here are the filenames that correspond with each message type. Please consult the RIB documentation for each message type in order to get a detailed picture of the composition of each message.

Message Types	Message Type Description	XML Schema Definition (XSD)
InvReqCre	Inventory Request Create Message	InvReqDesc.xsd

Item Location Subscription API

This section describes the item location subscription API.

Functional Area

Items

Business Overview

This API subscribes to item location from external systems to create or modify item location combinations in Merchandising. Item/location relationships can be created for an item and a single location or using one of the levels of the organizational hierarchy.

Creating Locations

When a new item location is created, this API will first validate that all required fields are present in the message. Additionally, when creating a new item location at least one detail line must also be included in the message. After that, business level validation on the input information will be performed. The tables below summarize these two types of validation.

Updating Item Location

When updating an item location, this API will first validate that all required fields are present in the message. Additionally, when updating an item location at least one detail line must also be included in the message. After that, business level validation on the input information will be performed. The tables below summarize these two types of validation.

Table 3-30 Item Location Create and Update

Message Element	Required?	Notes
Item	Yes	Must be an existing item in Merchandising.
Hier Level	Yes	Must be a valid organization hierarchy level. Valid values are chain (CH), area (AR), region (RE), district (DI), store (S) or warehouse (W).

Table 3-31 Item Location Detail Create and Update

Message Element	Required?	Notes
Hier Value	Yes	Must be valid ID for a chain, area, region, district, store or warehouse given the Hier Level.
Status	Yes	For new item/location relationships, this must be active (A), inactive (I), discontinued (C). If it is an update to an existing item/location, valid values are active (A), inactive (I), discontinued (C), or delete (D).
Store Ord Mult	Yes	This contains the multiple in which the item needs to be shipped from a warehouse to the location. Valid values are found in ORML.
Source Method	Conditional	This field determines the primary sourcing for this item/location. Valid values are Supplier (S) or Warehouse (W). Must either be supplier site or virtual warehouse. Additionally, valid value is dependent on the Hier Level: For Hier Level of store (S): <ul style="list-style-type: none"> Source Method is required to be warehouse (W) for wholesale or franchise stores. For Hier Level of warehouse (W) <ul style="list-style-type: none"> Source Method is required to be supplier (S) Source Method must be W if the Source Warehouse is populated.
Source Warehouse	Conditional	This determines which warehouse is the sourcing location for this item/location. This value is required if the sourcing method is Warehouse. Must be a valid virtual warehouse. Additionally, item should already be ranged to the Source Warehouse.
Receive as Type	Conditional	This determines whether the stock on hand for a pack component item or the buyer pack itself will be updated when a buyer pack is received at a warehouse. Valid values are Each (E) or Pack (P). Must be blank if Item is not a pack or Hier Level is not warehouse. Must be eaches (E), when Hier Level is warehouse and Hier Value is an internal finisher. Must be pack (P), when Item is a vendor pack. Must be pack (P), when Item order as type is pack and warehouse break pack indicator is N.
Taxable Indicator	No	This field determines if the item is taxable at the location. Valid values are Yes (Y) or No (N).
UIN Type	No	This contains the type of unique identification number (UIN) used to identify the instances of the item at the location. Valid values are found in code type UINT.

Table 3-31 (Cont.) Item Location Detail Create and Update

Message Element	Required?	Notes
UIN Label	Conditional	This contains the label for the unique identification number (UIN) when displayed. Valid values are found in code type ULBL. UIN Label is required if UIN Type is populated.
Ext UIN Indicator	No	This indicates if unique identification number (UIN) is being generated in the external system. Valid values are Yes (Y) or No (N).
Primary Supplier	No	This contains the primary supplier site for the item/location.
Primary Country	No	This contains the primary country of sourcing for the item/location.
Local Item Description	No	Could be used to have an alternate description for the item at this language, such as in another language.
Ti	No	This determines the number of shipping units (cases) that make up one tier of a pallet. Multiply TI x HI to get total number of cases for a pallet.
Hi	No	This determines the number of tiers that make up a complete pallet (height). Multiply TI x HI to get total number of cases for a pallet.
Daily Waste Percent	No	This defines the average percentage lost from inventory on a daily basis due to natural wastage.
Local Short Description	No	Could be used to provide an alternative short description for the item at this location, such as in another language.
UIN Type	No	This contains the type unique identification number (UIN) used to identify the instances of the item at the location. Valid values are found in code type UINT.
UIN Label	No	This contains the label for the unique identification number (UIN) when displayed. Valid values are found in code type ULBL.
Capture Time	No	This determines when the unique identification number (UIN) should be captured for an item during transaction processing. Valid values are found in code type CPTM.
Unit Cost	No	This contains the current unit cost of the item based on the primary supplier/country for the location in local currency.
Cost UOM	No	This is used to allow costs to be managed in a different UOM than the standard UOM.
Purchase Type	No	This defines whether the item is owned, consignment stock, or a concession item at the location. Valid values are: <ul style="list-style-type: none"> • 0 - Owned • 1 - Consignment • 2 - Concession

Table 3-31 (Cont.) Item Location Detail Create and Update

Message Element	Required?	Notes
Calculation Basis	No	This determines if the cost for the consignment/ concession item will be managed either based on cost per unit or as a percentage of retail. Valid values are: <ul style="list-style-type: none"> • C - Cost per Unit • P - Purchase Rate
Purchase Rate	No	This contains the percentage of the retail price which will determine the cost paid to the supplier for a consignment or concession item, if the calculation basis for the item/location is Purchase Rate.
Promotable Indicator	No	This determines whether the retailer is allowed to specify if the item is promotable or not. Valid values are Yes (Y) or No (N).

Table 3-32 Item Location Trait Create or Update

Message Element	Required?	Notes
Launch Date	No	This holds the date when the item is initially sold at the location.
Quantity Key Options	No	This field determines whether the quantity key on a POS can be used for this item at the location. Valid values are found in code type RPO.
Manual Price Entry	No	This field determines whether the price for the item/ location can be entered manually on POS. Valid values are found in code type RPO.
Deposit Code	No	This determines whether a deposit is associated with this item at the location. Valid values are found in code DEPO.
Food Stamp	No	This determines whether the item is approved for food stamps at the location. Valid values are Yes (Y) or No (N).
WIC	No	This determines whether the item is approved for WIC at the location. Valid values are Yes (Y) or No (N).
Proportional Tare Percent	No	This is the proportion of the total weight of a unit of an item that is the packaging. For example, if the tare item is bulk candy, this is the proportional of the total weight of one piece of candy that is the candy wrapper.
Fixed Tare Value	No	This is the tare of the packaging. For example, if the tare item is bulk candy, this is weight of the bag and twist tie.
Fixed Tare UOM	No	This contains the unit of measure value associated with the tare value.
Reward Eligible	No	This determines whether the item is legally valid for various types of bonus point/award programs at the location. Valid values are Yes (Y) or No (N).
National Brand Competitor Item	No	This contains the nationally branded item to which it will be compared to. Will contain a valid item ID.

Table 3-32 (Cont.) Item Location Trait Create or Update

Message Element	Required?	Notes
Return Policy	No	This determines the return policy for the item at the location. Valid values are found in code type RETP.
Stop Sale	No	This defines if the sale of the item should be stopped immediately at the location (i.e., in case of recall etc.). Valid values are Yes (Y) or No (N).
Electronic Market Clubs	No	This contains the code that represents the marketing clubs to which the item belongs to at the location. Valid values are found in code type MKTC.
Report Code	No	This determines to which reports the location should run. Valid values are found in code type REPC.
Shelf Life on Selection	No	This contains the required shelf life for an item on selection in days.
Shelf Life on Receipt	No	This contains the required shelf life for an item on receipt in days.
Investment Buy Shelf Life	No	Indicates the shelf life in days for this item at this location, which is used by the Investment Buy replenishment processing.
Store Reorderable	No	This determines whether the store may re-order the item. Valid values are Yes (Y) or No (N).
Rack Size	No	This determines the rack size that should be used for the item.
Full Pallet Item	No	This determines whether a store must reorder an item in full pallets only. Valid values are Yes (Y) or No (N).
In Store Market Basket	No	This contains the in-store market basket code for the item/location combination. Valid values are found in code type STMB.
Storage Location	No	This contains the current storage location or bin number for the item at the location.
Alternate Storage Location	No	This contains the preferred alternate storage location or bin number for the item at the location.
Returnable	No	This determines if the item can be returned at this location. Valid values are Y or N.
Refundable	No	This determines if the item is refundable at the location or not. Valid values are Yes (Y) or No (N).
Back Order	No	This determines if the item can be back ordered to the location. Valid values are Yes (Y) or No (N).

Flex Attributes

If you have defined any custom flex attributes (CFAS) for item locations, then they can be integrated as part of this API. The node of the integration that supports this will accept the name of the attribute as it is defined in the group set level view and the value for the attribute. For date type CFAS attributes, there is a separate value column to pass the value.

Table 3-33 Flex Attributes

Message Element	Required?	Notes
Name	Yes	Holds the attribute name.
Value	No	Holds the value of the attribute for number and character type attributes
Value Date	No	Holds the date for date type attributes.

Error Handling

If any errors are encountered in the validations described above or any of the message structure validations, a status of E is returned to the external system along with the appropriate error message. If the message has been successfully persisted, a success status (S) is returned to the external system, indicating that the message has been successfully received and persisted to the Merchandising database.

Message XSD

Below are the filenames that correspond with each message type. Consult the RIB documentation for each message type in order to get a detailed picture of the composition of each message.

Message Type	Message Type Description	XML Schema Definition (XSD)
xitemloccre	External item locations create	XItemlocDesc.xsd
xitemlocmod	External item locations modification	XItemlocDesc.xsd

Item Reclassification Subscription API

This section describes the item reclassification subscription API.

Functional Area

Merchandise Hierarchy

Business Overview

Merchandising subscribes to item reclassification messages, which update the department, class, and/or subclass for the item, that are published by an external system. This subscription is necessary in order to keep Merchandising in sync with the external system. This API allows external systems to create and delete item reclassification events within Merchandising.

Only the following item types can be interfaced using this API:

- Transaction level items without a parent
- Parent items, whose child items are the transaction level, such as with a fashion style (parent) and its SKUs (children)

- Complex pack items - but the reclassification cannot include the component items in the pack

The following item types cannot be reclassified:

- Child items with a parent - these are reclassified when the parent is updated
- Reference items - these below transaction level items are automatically reclassified with the transaction level item or its parent, whichever applies
- Simple Packs - these are reclassified when the component item is reclassified

This API allows a reclassification event to be created for a department/class/subclass combination that does not yet exist. This is valid as long as the merchandise hierarchy is scheduled to be created on or prior to the reclassification taking effect.

New/Updated Reclassifications

When a reclassification is created, both a reclassification header and detail are required. For an update, either to update the header or add a detail, all fields in both header and detail nodes are required.

Reclassification Header

Message Element	Required?	Notes
Reclass Number	Always	This is the unique number which identifies the reclassification event. For a detail create message, the reclassification number should already exist in Merchandising.
Reclass Description	Always	This is the description of the reclassification event.
Reclass Date	Always	The date on which the reclassification event is scheduled to take place. The date must be at least a day after the current business day.
To Department	Always	The department to which the item will belong after the reclassification event. The dept/class/subclass combination must already exist or will be created before the reclassification date.
To Class	Always	The class to which the item will belong after the reclassification event. The dept/class/subclass combination must already exist or will be created before the reclassification date.

Message Element	Required?	Notes
To Subclass	Always	The subclass to which the item will belong after the reclassification event. The dept/class/subclass combination must already exist or will be created before the reclassification date.
Item Reclassification Detail	Always	Child node

Reclassification Detail

Message Element	Required?	Notes
Item	Always	This is the item that will be reclassified. Following are the validations that will be done on upload: Must be a level 1 item If the item is a pack, it should not be a simple pack. Must not be on any approved order Must not be an orderable buyer pack that can be received as component items. Must not be on an existing reclassification

Deleting Reclassifications

Reclassifications can be deleted by:

- Deleting a single reclassification event
- Deleting specific items on a reclassification event
- Deleting all reclassification events on a particular event date
- Deleting all reclassification events

Reclassification Header

Message Element	Required?	Notes
Reclass Number	Conditional	The reclassification event that will be deleted. If this is populated in a header delete message, then the specific reclassification event will be deleted. This is required in a header delete message if Reclass Date is NULL and the Purge All indicator is N or NULL. If Reclass Number is provided, Reclass Date should be NULL.
Reclass Date	Conditional	The date of the reclassification events to be deleted. If this is populated in a header delete message, then all reclassification events occurring on the given date will be deleted. This is required in a header delete message if the Reclass Number is NULL and the Purge All indicator is N or NULL. If Reclass Date is provided, Reclass Number should be NULL.
Purge All	Conditional	If this field is Y in a header delete message, then all item reclassification events will be deleted. When the purge indicator is Y, both Reclass Number and Reclass date should be NULL. If the Purge All indicator is NULL or N, only the Reclass Number or the Reclass date should contain a value.
Reclassification Detail	Optional	Child Node

Reclassification Detail

Message Element	Required?	Notes
Item	Conditional	This is the item within a reclassification event that will be deleted. This is the only field required for detail delete messages as an item can only exist in one reclassification event. The entire reclassification event will be deleted when no item detail exists after detail deletion.

Error Handling

If any errors are encountered in the validations described above or any of the message structure validations, a status of E is returned to the external system along with the appropriate error message. If the message has been successfully persisted, a success status (S), is returned to the external system indicating that the message has been successfully received and persisted to the Merchandising database.

Message XSD

Here are the filenames that correspond with each message type. Please consult the RIB documentation for each message type in order to get a detailed picture of the composition of each message.

Message Type	Message Type Description	XML Schema Definition (XSD)
xitemrclscre	External item reclassification create	XItemRclsDesc.xsd
xitemrclsdtlcre	External item reclassification detail create	XItemRclsDesc.xsd
Xitemrclsdel	External item reclassification delete	XitemRclsRef.xsd
Xitemrclsdtldel	External item reclassification detail delete	XItemRclsRef.xsd

Item Subscription API

This section describes the item subscription API.

Functional Area

Items

Business Overview

This API subscribes to items from external systems to create, update or delete items in Merchandising. This subscription API provides a means to not only create, update and delete items but also create, update, and delete other details of the item structure.

Item messages include the detail nodes for the supplier and supplier/country. If the item is not a non-sellable pack, the item/zone/price node is also required. Optional nodes can be

included in the message for supplier/country, supplier/country/location, pack components, item/VAT, dimension, UDA, season, image, translations, HTS, assessments, expenses, and upcharges.

This API supports the insertion of data into the Item Induction staging tables based on the data sent through Oracle Retail Integration Bus (RIB), without requiring that the data sent to be functionally complete. If a complete set of item data is sent by the external system through this API, then it is possible to automatically trigger the upload of the data directly into Merchandising by setting a flag in the RIB message, such that item information is uploaded without further user interaction.

Items must be created and maintained following a logical hierarchy as outlined by the referential integrity of the item database tables: Item parents before child items; item components before items that are packs; items before item-suppliers; item/suppliers before item/supplier/countries; items before item/locations (a separate API), and so on. Failing to do so results in message failure.

The auto-creation of item children using differentiator records attached to an item parent, as currently occurs using Merchandising online processes, is not supported in this API.

These transactions are performed immediately upon message receipt so success or failure can be communicated to the calling application.

Creating Items

When a new item is created, this API will first validate that all required fields are present in the message. After that, business level validation on the input information will be performed. If all the validations are met, the item in the message data is created in Merchandising. The tables below summarize these two types of validation.

Header Level Validation

Message Element	Required?	Notes
Item	Always	The unique identifier of the main item on the message.
Item parent	Optional	Must be an existing item in Merchandising.
Item grandparent	Optional	Must be an existing item in Merchandising.
Pack indicator	Optional	Valid values are Y and N. The value Y indicates this item is a pack item.
Item level	Optional	Valid values are 1 (single level or pack item), 2 (child item), and 3 (grandchild item).
Transaction level	Optional	Valid values are 1, 2, and 3.
Differentiator 1	Optional	Must be an existing differentiator or differentiator group in Merchandising. If the item on the message is a parent, this field may contain a differentiator group ID.
Differentiator 2	Optional	same as Differentiator 1.
Differentiator 3	Optional	same as Differentiator 1.
Differentiator 4	Optional	same as Differentiator 1.
Department	Always	Must be an existing department in Merchandising.

Message Element	Required?	Notes
Class	Optional	Must be an existing class in the dept in Merchandising.
Subclass	Optional	Must be an existing subclass in the dept/class in Merchandising.
Item description	Always	Contains the complete, long description for the item being created.
Item supplier country location hierarchy level	Optional	Must be a valid organization hierarchy level. Valid values are: A - Area AI - All Internal Finishers AS - All Stores AW - All Warehouses D - District DW - Default Warehouse I - Internal Finisher L - Location Trait PW - Physical Warehouse R - Region S - Store T - Transfer Zone W - Warehouse
Item zone price hierarchy level	Optional	Must be a valid organization hierarchy level. Valid values are chain (C), area (A), region (R), district (D), store (S) or warehouse (W). Must be null if the item is a non-sellable pack.
Short Description	Optional	Contains a shortened description of the item being created. Generally, this description would be used by other systems where it is problematic to display the full long description (for example, a POS system, or customer receipt).
Cost zone group id	Optional	Must be an existing cost zone in Merchandising. Must be null if landed cost is not being used in the system or the item is an orderable buyer pack.
Standard UOM	Optional	Must be a valid UOM. Defaulted to Eaches (EA), if not populated in the message.
Store order multiple	Optional	Valid values are Cases (C), Inners (I), and Eaches (E).
Forecast indicator	Optional	Indicates if this item will have a sales forecast created for it by a forecasting module. Valid values are Y and N. Must be N if the item is a pack.
Simple pack indicator	Conditional	Valid values are Y and N. This field is required if the item is a pack.

Message Element	Required?	Notes
Contains inner indicator	Always	Valid values are Y and N. This should be Y if the item is a pack and at least one component item is a pack. Otherwise, it should be N.
Sellable indicator	Optional	Indicates if this item will be made available for sale to end consumers. Valid values are Y and N. It should be Y for a non-pack item.
Orderable indicator	Optional	Indicates if this item will be orderable. Valid values are Y and N. It should be Y for a non-pack item.
Pack type	Optional	Valid values are buyer (B) and vendor (V). Must be Null for a non-pack item.
Order as type	Optional	Must be a valid code from code type PARC if populated: E - Eaches P - Pack Must be Null for a non-pack item.
Comments	Optional	Free form text
Create datetime	Optional	
Status	Optional	Valid values are worksheet (W), submitted (S), approved (A) and deleted (D).
UOM conversion factor	Optional	
Package size	Optional	
Handling temperature	Optional	Must be a valid code from code type HTMP if populated.
Handling sensitivity	Optional	Must be a valid code from code type HSEN if populated.
Manufacturer's recommended retail	Optional	
Waste type	Optional	Must contain either: SL - Sales Wastage SP - Spoilage Wastage
Waste percentage	Optional	Must be greater than 0 or less than 100 if populated.

Message Element	Required?	Notes
Item number type	Optional	<p>Must be a valid code from code type UPCT if populated (note: new codes for this code type are not supported):</p> <p>EAN13 - EAN/UCC-13 EAN13S - EAN/UCC-13 with Supplement EAN8 - EAN/UCC-8 ISBN10 - ISBN-10 ISBN13 - ISBN-13 ITEM - Oracle Retail Item Number MANL - Manual NDC - NDC/NHRIC - National Drug Code PLU - PLU SSCC - SSCC Shipper Carton UCC14 - EAN/UCC-14 UPC-A - UCC12 UPC-AS - UCC12 with Supplement UPC-E - UCC8 UPC-ES - UCC8 with Supplement VPLU - Variable Weight PLU</p>
Catch weight indicator	Optional	Indicates if this item should be weighed upon receipt. Valid values are Y and N.
Constant dimension indicator	Optional	Valid values are Y and N.
Gift wrap indicator	Optional	Valid values are Y and N.
Ship alone indicator	Optional	Valid values are Y and N.
External source system	Optional	Not used by Merchandising.
Size group 1	Optional	Not used by Merchandising.
Size group 2	Optional	Not used by Merchandising.
Size 1	Optional	Not used by Merchandising.
Size 2	Optional	Not used by Merchandising.
Color	Optional	Not used by Merchandising.
System indicator	Optional	Not used by Merchandising.
UPC supplement	Optional	Not used by Merchandising.
UPC type	Optional	Not used by Merchandising.
Primary UPC indicator	Optional	Not used by Merchandising.
Primary replenishment indicator	Optional	Not used by Merchandising.
Item aggregate indicator	Optional	Used only for a parent level item. Used in conjunction with the Diff 1-4 aggregate indicators to define which diff values should be used in aggregations along with the parent item, for example, to specify a style (parent item) /color (diff) aggregation. Valid values are Y and N.
Differentiator 1 aggregate indicator	Optional	Valid values are Y and N.

Message Element	Required?	Notes
Differentiator 2 aggregate indicator	Optional	Valid values are Y and N.
Differentiator 3 aggregate indicator	Optional	Valid values are Y and N.
Differentiator 4 aggregate indicator	Optional	Valid values are Y and N.
Perishable indicator	Optional	Valid values are Y and N.
Notional pack indicator	Optional	Valid values are Y and N.
Stock on Hand inquiry at pack indicator	Optional	Valid values are Y and N.
AIP case type	Optional	Must be one of the following if populated: F - Formal I - Informal
Order type	Optional	Must be one of the following if populated: F - Fixed Weight V - Variable Weight
Sale type	Optional	Must be one of the following if populated: L - Loose Weight V - Variable Weight Each
Catch weight UOM	Optional	
Deposit item type	Optional	Valid values are: E - Contents A - Container Z - Crate T - Returned Item (Empty bottle)
Inventory indicator	Optional	Indicates if inventory will be managed for this item. Valid values are Y and N.
Item transformation indicator	Optional	Valid values are Y and N.
Container item	Conditional	Required if the Deposit item type is E.
Package UOM	Optional	
Format ID	Optional	Required for Variable PLU item types.
Prefix	Optional	
Brand	Optional	Must be an existing brand in Merchandising if populated.
Product classification	Optional	Must be a valid code from code type PCLA if populated.
Item description secondary	Optional	
Description uppercase	Optional	
Merchandise indicator	Optional	Valid values are Y and N.
Original retail	Optional	
Retail label type	Optional	Must be a valid code from code type RTLT if populated.
Retail label value	Optional	

Message Element	Required?	Notes
Default waste percentage	Always	Must be NULL for below transactional level items. Must be greater than 0 and less than 100, if populated. For transactional level or above items, this field must be populated if wastage type is SP.
Item service level	Optional	
Check UDA indicator	Optional	
Deposit in price per UOM	Optional	Required if deposit item type is E. Valid values are: E - Exclusive of Deposit Amount I - Inclusive of Deposit Amount
Attempt RMS load	Always	Valid values are STG and Merchandising. The value STG indicates the item will be loaded into induction staging environment as an incomplete item. The value Merchandising indicates the item is complete and an attempt to create the item in Merchandising will be made. In the case of Merchandising, the item will be subject to all validations for a completed item.

A pack item may be created by providing the component item information while creating the item. This detail is required if item is a pack item. Below are the validations:

Message Element	Required?	Notes
Component item	Always	Must be an existing item in Merchandising.
Pack quantity	Always	Must be an integer if standard UOM is 'EA'. Must be greater than 0.

Additionally, item zone prices may be updated together with the creation of a new item. Below are the validations:

Message Element	Required?	Notes
Hierarchy ID	Always	
Base retail Indicator	Optional	Not used by Merchandising.
Selling unit retail	Optional	
Selling UOM	Optional	Unit of Measure for the selling unit retail. UOM must be from either the QTY, MASS, VOL, AREA or DIMEN class.
Multi selling UOM	Optional	Unit of Measure for the multi-selling units and retail. UOM must be from either the QTY, MASS, VOL, AREA or DIMEN class.
Country ID	Optional	
Currency code	Optional	Not used by Merchandising.

Message Element	Required?	Notes
Multi units	Optional	Must be populated only if multi-unit retail is populated. Must be greater than 1 if populated.
Multi-unit retail	Optional	Must be greater than 0.

If the above validations pass, then the item will be created with the status defined in the message. If the status in the message is approved, then the item will also be subjected to a series of approval checks. For an item to be successfully approved, mandatory information such as supplier, supplier country, component item information (if item is a pack) is required to be passed as part of the item message. If the item cannot be approved, it will not be created.

You can also include following information on the item - expenses, HTS and assessments, tickets, UDA, VAT (for SVAT tax type), Upcharges, Images, Seasons, Item supplier country locations. If included, these will be created simultaneously with the creation of the item.

Updating Items

Updates can be made to the items that are in Worksheet, Submitted or Approved. For the update messages, the API will validate that the item number included in the message already exists in Merchandising.

Header Level Updates

Only header level fields need to be provided for header level updates. Any item details included in the message will be ignored for a header level update message. There are certain fields that are not allowed to be updated at header level depending on the status, and if these are still provided in the message, appropriate error message will be returned.

Deleting Items

If you are deleting an item, the API will first validate that the item number is valid. The item number is the only required field for a header delete message. If the item does not exist in Merchandising a reject message will be returned.

Optionally the item detail records can be deleted by integrating as part of the item delete message.

Table 3-34 Item Delete

Message Element	Required?	Notes
Item	Always	Must be an existing item in Merchandising.
Hierarchy Level	Optional	Not used
Item Country	Optional	Child Node
Item Supplier	Optional	Child Node
Item VAT	Optional	Child Node

Table 3-34 (Cont.) Item Delete

Message Element	Required?	Notes
Item Image	Optional	Child Node
Item Season	Optional	Child Node
Item UDA	Optional	Child Node
Item Pack	Optional	Not used.
System Indicator	Optional	Not used.
UPC Supplement	Optional	Not used.
Item Header Translation	Optional	Child Node
Item HTS	Optional	Child Node
Item Expenses	Optional	Child Node
Item Ticket	Optional	Child Node
Item Up-Charges	Optional	Child Node

Creating an Item Supplier

An item supplier may be created together with the creation of a new item or added to an existing item. Below are the message details and validations:

Message Element	Required?	Notes
Supplier	Always	Must be an existing supplier in Merchandising.
Primary supp indicator	Always	Indicates if this supplier is the primary supplier for the item. An item can only have one primary supplier. Valid values are Y and N.
VPN	Optional	Vendor Part/Product Number. Must not contain the new line, pipe, linefeed or semicolon characters.
Supplier label	Optional	
Consignment rate	Optional	Used only for consignment items, to specify the rate of consignment (the percentage of the retail price that will be remitted to the supplier).
Supp discontinue date	Optional	Must be greater than or equal vdate.
Direct ship indicator	Optional	Indicates if this item can be directly shipped to end consumers from this supplier. Valid values are Y and N.
Pallet name	Optional	Must be a valid code from code type PALN if populated: FLA - Flat PAL - Pallet Additional codes can be added to the code type.

Message Element	Required?	Notes
Case name	Optional	Must be a valid code from code type CASN if populated: BA - Barrel BBL - Barrel BE - Bundle BG - Bag BI - Bin BJ - Bucket BK - Basket BX - Box CA - Can CON - Container CR - Crate CS - Case CT - Carton PACK - Pack PO – Pot Additional codes can be added to the code type.
Inner name	Always	Must be a valid code from code type INRN if populated: EA - Eaches INR - Inner SCS - Sub-Case SPACK - Sub-Pack Additional codes can be added to the code type.
Primary case size	Conditional	Required for an orderable item that has been configured for Informal (I) case types.
Supplier differentiator 1	Optional	
Supplier differentiator 2	Optional	
Supplier differentiator 3	Optional	
Supplier differentiator 4	Optional	
Concession rate	Always	Required for items defined as concession items.
Default expense profiles indicator	Optional	

Updating an Item Supplier

In order to update an item's supplier information, the supplier must exist for the item in Merchandising, otherwise, an error will be returned. All fields identified in the create section above are updateable and will go through the same validation as in the creation of an item supplier.

Deleting an Item Supplier

In order to delete an item supplier, the supplier must exist for the item in Merchandising, otherwise, an error will be returned. Optionally the item supplier detail records can be deleted by integrating as part of the item supplier delete message.

Table 3-35 Item Supplier Delete

Message Element	Required?	Notes
Supplier	Always	Must be an existing supplier for the item in Merchandising.
Delete Children Indicator	Always	Indicates whether the deletion should be applied to all associated item/supplier records below the item level of the item specified on the deletion message. Valid values are Y and N.
Item Supplier Country	Optional	Child node.
Item Supplier Country of Manufacture	Optional	Child node.
Item Supplier translation	Optional	Child node.

Creating an Item Supplier Country

An item supplier country may be created together with the creation of a new item or added to an existing item supplier. Below are the validations:

Message Element	Required?	Notes
Origin country ID	Always	Must be an existing country in Merchandising.
Primary country indicator	Always	Indicates the primary origin country for the item / supplier. An item/supplier can only have one primary origin country. Valid values are Y and N.
Unit cost	Optional	Contains the unit cost for the item/supplier/ origin country. This must be provided if the item is orderable.
Lead time	Optional	Must be between 0 and 9999.
Pickup lead time	Optional	Must be between 0 and 9999.
Minimum order quantity	Optional	Must be greater than 0.
Maximum order quantity	Optional	Must be greater than 0.
Supplier hierarchy level 1	Optional	Can be used to identify the first level of a supplier hierarchy, such as the manufacturer.
Supplier hierarchy level 2	Optional	Can be used to identify the second level of a supplier hierarchy, such as the distributor.
Supplier hierarchy level 3	Optional	Can be used to identify the third level of a supplier hierarchy, such as the wholesaler.

Message Element	Required?	Notes
Default UOP	Always	Contains the default Unit of Purchase for the item at the supplier/origin country.
Supplier pack size	Always	Must be greater than 0. Must not be a decimal if Cost UOM class is Qty
Inner pack size	Always	Must be less than supplier pack size and supplier pack size must be a multiple of inner pack size if populated. Must not be decimal if Cost UOM class is Qty
Ti	Always	Contain the number of cased in a tier on a pallet. Must be greater than 0. The pallet size for an item is the supplier pack size x Ti x Hi.
Hi	Always	Contains the number of tiers high for a pallet. Must be greater than 0. The pallet size for an item is the supplier pack size x Ti x Hi.
Cost UOM	Always	Must be an existing Unit of measure in Merchandising. If the UOM Class is QTY, inner pack size and supp pack size cannot be a decimal.
Tolerance type	Conditional	Must be one of the following: A - Actual P - Percentage Required for a variable weight simple pack catchweight item.
Minimum tolerance	Conditional	Required if the item is a variable weight simple pack catchweight item. Must be greater than 0. Must be less than 100 for Percentage tolerance type. Must be less than net weight for Actual tolerance type. Must be less than max tolerance.
Maximum tolerance	Optional	Required if the item is a variable weight simple pack catchweight item. Must be greater than 0. Must be less than 100 for Percentage tolerance type. Must be greater than net weight for Actual tolerance type. Must be greater than max tolerance.
Supplier hierarchy type 1	Optional	Used in conjunction with the Supplier Hierarchy Level 1 to define the type, for example Manufacturer.
Supplier hierarchy type 2	Optional	Used in conjunction with the Supplier Hierarchy Level 1 to define the type, for example Distributor.

Message Element	Required?	Notes
Supplier hierarchy type 3	Optional	Used in conjunction with the Supplier Hierarchy Level 1 to define the type, for example Wholesaler.
Round level	Optional	Must be one of the following: C - Case CL - Case/Layer CLP - Case/Layer/Pallet L - Layer LP - Layer/Pallet P – Pallet Used in conjunction with the rounding percentages to determine when to round up or down to the nearest inner, case, layer, or pallet.
Round to inner percentage	Always	Must be between 0 and 100.
Round to case percentage	Always	Must be between 0 and 100.
Round to layer percentage	Always	Must be between 0 and 100.
Round to pallet percentage	Always	Must be between 0 and 100.
Packing method	Optional	Must be a valid code from code type PKMT if populated: FLAT - Flat HANG - Hanging
Default expense profiles indicator	Optional	
Purchase type	Conditional	Required if consignment concession flag is Y. 0 – Owned (default) 1 - Consignment 2 - Concession
Calculation basis	Conditional	Required if consignment concession flag is Y and purchase type is either 1 or 2. C - Cost per Unit P - Purchase Rate
Purchase rate	Conditional	Required if purchase type is either 1 or 2 and calculation basis is P. Must be between 0 and 100.

Updating an Item Supplier Country

In order to update an item supplier country information, the supplier country must exist for the item in Merchandising, otherwise, an error will be returned. All the updateable fields will go through the same validation as in the creation of an item supplier country.

Deleting an Item Supplier Country

In order to delete an item supplier country, the supplier country must exist for the item in Merchandising, otherwise, an error will be returned.. Optionally, the item supplier country

detail records can be deleted by integrating as part of the item supplier country delete message.

Table 3-36 Item Supplier Country Delete

Message Element	Required?	Notes
Origin Country Id	Always	Must be an existing supplier country for the item in Merchandising.
Item Supplier Country Location	Optional	Child Node
Item Supplier Country Dimension	Optional	Child Node

Creating Item Supplier Manufacturing Country

An item supplier manufacturing country may be created together with the creation of a new item or added to an existing item supplier. Below are the validations:

Message Element	Required?	Notes
Manufacturer country ID	Always	Must be an existing country in Merchandising.
Primary manufacturer country indicator	Optional	Valid values are Y and N.

Updating Item Supplier Manufacturing Country

In order to update an item supplier country information, the supplier manufacturing country must exist for the item in Merchandising, otherwise, an error will be returned. All the updateable fields will go through the same validation as in the creation of an item supplier manufacturing country.

Deleting Item Supplier Manufacturing Country

In order to delete an item supplier manufacturing country, the supplier manufacturing country must exist for the item in Merchandising, otherwise, an error will be returned.

Table 3-37 Item Supplier Country of Manufacture Delete

Message Element	Required?	Notes
Manufacturer Country Id	Always	Must be an existing supplier manufacturer country for the item in Merchandising.

Creating Item Supplier Country Location

Item supplier country location can be created together with the creation of a new item or added to an existing item supplier country. Records are not required at this level for an item. If provided the values override those defined at the item/supplier/country level for the specified location(s). Below are the validations:

Message Element	Required?	Notes
Hierarchy ID	Always	<p>Must be NULL when item-supplier-country-location hierarchy level is either All Stores (AS), All Warehouses (AW) or All Internal Finishers (AI).</p> <p>Required if item-supplier-country-location hierarchy level is either Store (S), Default Warehouse (DW), Warehouse (W), Physical Warehouse (PW), Internal Finisher (I), District (D), Region (R), Area (A), Transfer Zone (T) or Location Trait (L). Contains the identifier for the specified location hierarchy level. For example, if the location hierarchy level is S this will contain a store ID.</p>
Unit cost	Optional	<p>Required if item is an owned item or a consignment/concession item with calculation basis as C, when consignment concession flag is set to Y.</p> <p>Required if consignment concession flag is set to either N or D.</p> <p>Must be blank for buyer packs.</p> <p>Must be greater than 0 and less than 9999999999999.9999 if populated.</p>
Negotiated item cost	Optional	
Primary location indicator	Optional	Indicates if the location is the primary for the item/supplier/origin country. Valid values are Y and N.
Pickup lead time	Optional	Must be between 0 and 9999.
Round level	Optional	<p>Must be one of the following:</p> <p>C - Case</p> <p>CL - Case/Layer</p> <p>CLP - Case/Layer/Pallet</p> <p>L - LayerLP - Layer/Pallet</p> <p>P – Pallet</p> <p>Used in conjunction with the rounding percentages to determine when to round up or down to the nearest inner, case, layer, or pallet.</p>
Round to case percentage	Optional	Must be between 0 and 100.
Round to layer percentage	Optional	Must be between 0 and 100.
Round to pallet percentage	Optional	Must be between 0 and 100.
Round to inner percentage	Optional	Must be between 0 and 100.
Supplier hierarchy level 1	Optional	Can be used to identify the first level of a supplier hierarchy, such as the manufacturer.
Supplier hierarchy level 2	Optional	Can be used to identify the second level of a supplier hierarchy, such as the distributor.

Message Element	Required?	Notes
Supplier hierarchy level 3	Optional	Can be used to identify the third level of a supplier hierarchy, such as the wholesaler.
Cost UOM	Always	Must be an existing Unit of measure in Merchandising.
Purchase type	Conditional	Must be populated when consignment concession flag is Y. 0 - Owned 1 - Consignment 2 - Concession
Calculation basis	Conditional	Must be provided when consignment concession flag is Y and purchase type is either 1 or 2. Must be one of the following: C - Cost per Unit P - Purchase Rate
Purchase rate	Conditional	Must be provided if purchase type is either 1 or 2 and calculation basis is P. Must be between 0 and 100.

Updating Item Supplier Country Locations

In order to update an item supplier country location information, the supplier country location must exist for the item in Merchandising, otherwise, an error will be returned. All the updateable fields will go through the same validation as in the creation of an item supplier country location.

Deleting Item Supplier Country Locations

In order to delete an item supplier country location, the supplier country location must exist for the item in Merchandising, otherwise, an error will be returned.

Table 3-38 Item Supplier Country Location Delete

Message Element	Required?	Notes
Hierarchy Id	Always	Must be an existing location for the item supplier country in Merchandising.

Creating Item Supplier Country Dimensions

Item supplier country dimensions can be created together with the creation of a new item or added to an existing item supplier country. Dimensions are not required for an item. However, if non-standard units of measure will be used for the item there must be a case-type dimension provided. For example, for a selling unit retail to be defined in ounces there must be a dimension defined containing the liquid and volume to be used when converting between a unit and an ounce. Below are the validations:

Message Element	Required?	Notes
Dimension object	Always	Must be one of the following: CA - Case EA - Each IN - Inner PA - Pallet
Tare weight	Optional	Must be greater than zero and less than weight (gross weight).
Tare type	Optional	Must be one of the following: D - Dry W - Wet
Length width and height UOM	Conditional	Must be a valid UOM of the UOM Class DIMEN, if populated.
Length	Optional	Must be provided if dimension object is populated and UOM class is DIMEN. Must be greater than 0.
Width	Optional	Must be provided if dimension object is populated and UOM class is DIMEN. Must be greater than 0.
Dim height	Optional	
Liquid volume	Optional	Required if dimension object is populated and it's UOM class is LVOL. Must be greater than 0, if populated.
Liquid volume UOM	Optional	Must be a valid UOM of the UOM Class LVOL, if populated.
Statistical cube	Optional	Must be greater than 0, if populated.
Weight UOM	Optional	Must be a valid UOM of the UOM Class MASS, if populated.
Weight	Optional	Required if tare type and net weight are populated. Must be less than tare weight, if populated. Must be greater than 0, if populated.
Net weight	Optional	Minimum tolerance value must be less than net weight for an actual tolerance type. Maximum tolerance value must be greater than net weight. Must be greater than 0, if populated. Must be less than or equal to weight (gross weight).
Presentation method	Optional	Must be a valid code from code type PCKT, if populated.

Updating Item Supplier Country Dimensions

In order to update an item supplier country dimension information, the supplier country dimension must exist for the item in Merchandising, otherwise, an error will be returned. All

the updateable fields will go through the same validation as in the creation of an item supplier country dimension.

Deleting Item Supplier Country Dimensions

In order to delete an item supplier country dimension, the supplier country dimension must exist for the item in Merchandising, otherwise, an error will be returned.

Table 3-39 Item Supplier Country Dimension Delete

Message Element	Required?	Notes
Dimension Object	Always	Must be an existing supplier country dimension for the item in Merchandising.

Creating Item VAT

Item VAT (value added tax) may be created together with the creation of a new item or added to an existing item when default tax type for the system is SVAT (Simple VAT). If not populated, item vat is defaulted from the corresponding department's vat information. Below are the validations:

Message Element	Required?	Notes
VAT Type	Always	Valid values for are Cost (C), Retail (R) and Both (B).
VAT Region	Always	Must be an existing VAT Region in Merchandising, if populated
VAT Code	Always	Must be an existing VAT code in Merchandising, if populated
Active Date	Always	Must not be earlier than or equal to vdate.
Reverse VAT Indicator	Always	Valid values are Y and N.

Deleting Item VAT

In order to delete an item VAT, the vat information must exist for the item in Merchandising, otherwise, an error will be returned.

Table 3-40 Item VAT Delete

Message Element	Required?	Notes
VAT Type	Always	Must be an existing VAT type for the VAT Item in Merchandising.
VAT Region	Always	Must be an existing VAT region for the VAT Item in Merchandising.
VAT Code	Always	Must be an existing VAT code for the VAT Item in Merchandising.
Active Date	Always	Must be an existing active date for the VAT Item in Merchandising.

Creating Item UDA

Item UDA (user defined attribute) of type date/freeform text/list of value can be created together with the creation of a new item or added to an existing item. Below are the validations:

Message Element	Required?	Notes
UDA ID	Always	Must be an existing UDA in Merchandising for the given display type.
Display type	Always	Must contain one of the following to indicate the type of UDA: <ul style="list-style-type: none"> • DT – Date • LV – List of Values • FF – Free Form
UDA date	Optional	Must be populated for 'DT' display type.
UDA value	Optional	Must be populated for 'LV' display type.
UDA text	Optional	Must be populated for 'FF' display type.
New UDA date	Optional	Must not already be present in Merchandising for the given item UDA combination. Must not be same as the existing value for the given item UDA combination.
New UDA value	Optional	Must not already be present in Merchandising for the given item UDA combination. Must not be same as the existing value for the given item UDA combination.
New UDA text	Optional	Must not already be present in Merchandising for the given item UDA combination. Must not be same as the existing value for the given item UDA combination.

Updating Item UDA

In order to update an item UDA of type date/freeform text/list of values, the UDA must exist for the item in Merchandising, otherwise, an error will be returned. All the updateable fields will go through the same validation as in the creation of an item UDA.

Deleting Item UDA

In order to delete an item UDA, the UDA must exist for the item in Merchandising, otherwise, an error will be returned.

Table 3-41 Item UDA Delete

Message Element	Required?	Notes
UDA Id	Always	Must be an existing UDA for the Item in Merchandising.

Table 3-41 (Cont.) Item UDA Delete

Message Element	Required?	Notes
Display Type	Always	Display Type of the UDA. Valid values are: LV - List of Values FF - Freeform DT - Date
UDA Value	Optional	This can either be a UDA value (for LV display type) or a UDA date (for DT display type) or a freeform text (for FF display type). This must be an existing value for the Item UDA in Merchandising.

Creating Item Season

Item Seasons can be created together with the creation of a new item or added to an existing item. Below are the validations:

Message Element	Required?	Notes
Season ID	Always	Must be an existing season in Merchandising.
Phase ID	Always	Must be an existing phase for the given season in Merchandising.
Item season sequence number	Optional	Used to ensure uniqueness. Each season/phase combination for an item should have a unique sequence number.
Diff ID	Optional	Can be used to differentiate an item/season by a diff, for example if the season/phase applies to only a specific color for a style.
Create datetime	Optional	Holds the date time stamp the item/season/phase was created.
Last update datetime	Optional	Holds the date time stamp of the most recent update by the Last update ID.
Last update ID	Optional	Holds the ID of the user who most recently updated this record.
color	Optional	Not used by Merchandising.

Deleting Item Season

In order to delete an item season, the season-phase ID must exist for the item in Merchandising, otherwise, an error will be returned.

Table 3-42 Item Season Delete

Message Element	Required?	Notes
Season ID	Always	Must be an existing season for the Item in Merchandising.
Phase ID	Always	This must be an existing phase for the item season in Merchandising.
Differentiator ID	Optional	This must be a valid group/differentiator ID for the item in Merchandising.
Color	Optional	Not used.

Creating Item Images

Item images can be created together with the creation of a new item or added to an existing item. Below are the validations:

Message Element	Required?	Notes
Image name	Always	Contains the image file name (for example, image.jpg).
Image address	Optional	Contains the image URL path. If not provided the system default will be used.
Image description	Optional	Contains a name for the image that is logical for a user to be able to identify the image.
Create datetime	Optional	May contain the date timestamp the image was added for the item.
Last update datetime	Optional	May contain the date timestamp for when the image was last updated.
Last update ID	Optional	May contain the ID for the user that made the last update.
Image type	Always	Must be a valid code from code type = IITD if populated: H - High L - Low M - Medium T - Thumbnail
Primary indicator	Always	Valid values are Y and N.
Display priority	Always	Must be greater than 0.

Updating Item Images

In order to update an item image, the image name must exist for the item in Merchandising, otherwise, an error will be returned. All the updateable fields will go through the same validation as in the creation of an item image.

Deleting Item Images

In order to delete an item image, the image must exist for the item in Merchandising, otherwise, an error will be returned. Optionally, the item image translation can be deleted by integrating as part of the item image delete message.

Table 3-43 Item Image Delete

Message Element	Required?	Notes
Image Name	Always	Must be an existing image for the item in Merchandising.
Image Translations	Optional	Child Node

Creating HTS and Assessments

HTS and Assessment may be created together with the creation of a new item or added to an existing item. Below are the validations:

Table 3-44 HTS Validation

Message Element	Required?	Notes
HTS	Always	Contains the HTS code. The specified code must exist in Merchandising.
Import country ID	Always	Contains the country ID where the item is being imported into and the HTS code applies.
Origin country ID	Always	Contains the country ID where the item originated from.
Effect from Date	Always	Contains the first date the HTS code is effective for the item.
Effect to Date	Always	Contains the last date the HTS code is effective for the item.
Clearing zone ID	Optional	Must be an existing clearing zone in Merchandising and the clearing zone must be associated with the import country.
Status	Optional	Valid values are worksheet (W) and approved (A).

Table 3-45 HTS Assessment Validation

Message Element	Required?	Notes
Component ID	Always	Component ID must exist on the ELC components table.
CVB code	Optional	Required if the component rate calculation basis is value (V), otherwise this will be defaulted to NULL. Must be a valid CVB code in Merchandising.
Component rate	Optional	Must be greater than or equal to 0 is populated. Required if CVB Code is populated.

Table 3-45 (Cont.) HTS Assessment Validation

Message Element	Required?	Notes
Per count	Optional	Required if CVB Code is populated. Defaulted to NULL if calculation basis is V, else it must be greater than or equal to 0.0001.
Per count UOM	Optional	Required if calculation basis is V - Value. Must be an existing UOM in Merchandising.
Estimated assessment Value	Optional	
Nominal flag 1	Optional	Valid values are '+', '-' and 'N', if populated. Defaulted to N if not populated.
Nominal flag 2	Optional	Valid values are '+', '-' and 'N', if populated. Defaulted to N if not populated.
Nominal flag 3	Optional	Valid values are '+', '-' and 'N', if populated. Defaulted to N if not populated.
Nominal flag 4	Optional	Valid values are '+', '-' and 'N', if populated. Defaulted to N if not populated.
Nominal flag 5	Optional	Valid values are '+', '-' and 'N', if populated. Defaulted to N if not populated.

Updating HTS and Assessments

In order to update HTS and assessments, the record to be updated must exist in Merchandising, otherwise, an error will be returned. Status and origin country ID can be updated at the HTS level. For assessments, all fields identified in the create section above except for component ID are updateable and will go through the same validation as in the creation of assessments.

Deleting HTS and Assessments

In order to delete HTS and assessments, the record to be deleted must exist in Merchandising, otherwise, an error will be returned.

Table 3-46 Item HTS Delete

Message Element	Required?	Notes
HTS	Always	The specified HTS code must exist for the item in Merchandising.
Import country ID	Always	The specified import country Id must exist for the item/HTS in Merchandising.
Origin country ID	Always	The specified HTS import Id must exist for the item/HTS in Merchandising.
Effect from Date	Always	The specified effective from date must exist for the item/HTS in Merchandising.
Effect to Date	Always	The specified effective to date must exist for the item/HTS in Merchandising.
Item HTS Assessments	Optional	Child Node

Table 3-47 Item HTS Assessments Delete

Message Element	Required?	Notes
Component Id	Always	The specified component must exist for the item HTS assessment in Merchandising.

Creating Expenses

Expenses may be created together with the creation of a new item or added to an existing item that has suppliers defined. Below are the validations:

Message Element	Required?	Notes
Supplier	Always	Item supplier relationship must exist in Merchandising.
Component ID	Always	Component ID must exist on the ELC components table of type Expense (E).
Discharge port	Always	Must be an existing outside location of type 'DP' in Merchandising, if populated.
Origin country ID	Optional	Must be an existing country in Merchandising, if populated.
Lading port	Optional	Must be an existing outside location of type 'LP' in Merchandising, if populated.
Zone ID	Optional	Must be an existing cost zone in Merchandising, if populated.
Zone group ID	Optional	Must be an existing cost zone group in Merchandising, if populated.
Base Expense Indicator	Optional	Valid values are Y and N.
CVB code	Optional	Must be a valid CVB code in Merchandising.
Component rate	Optional	Required if component ID is populated. Must be greater than or equal to 0 if populated.
Per count	Optional	Required if calculation basis is S (Specific). Must be greater than 0, if populated. Defaulted to NULL for Value calculation basis.
Per count UOM	Optional	Required if calculation basis is Specific (S). Must be an existing UOM in Merchandising, if populated. Defaulted to NULL for Value (V) calculation basis.
Component currency	Optional	Must be an existing currency in Merchandising if populated.
Update orders Indicator	Optional	Valid values are Y and N, if populated.
Nominal flag 1	Optional	Valid values are '+', '-' and 'N', if populated.
Nominal flag 2	Optional	Valid values are '+', '-' and 'N', if populated.

Message Element	Required?	Notes
Nominal flag 3	Optional	Valid values are '+', '-' and 'N', if populated.
Nominal flag 4	Optional	Valid values are '+', '-' and 'N', if populated.
Nominal flag 5	Optional	Valid values are '+', '-' and 'N', if populated.

Updating Expenses

In order to update expenses, the item/supplier/component ID must exist in Merchandising, otherwise, an error will be returned. All fields identified in the create section above except for supplier/component ID are updateable and will go through the same validation as in the creation of expenses.

Deleting Expenses

In order to delete expenses, the item/supplier/component ID/discharge must exist for the item in Merchandising, otherwise, an error will be returned.

Table 3-48 Item Expense Delete

Message Element	Required?	Notes
Supplier	Always	Must be an existing supplier expense for the item in Merchandising.
Component Id	Always	Must be an existing component Id for the item supplier expense in Merchandising.
Discharge port	Always	Must be an existing discharge port location for the item supplier expense in Merchandising.
Origin country ID	Optional	Must be an existing country for item supplier expense in Merchandising, if populated.
Lading port	Optional	Must be an existing lading port location for the item supplier expense in Merchandising, if populated.
Zone ID	Optional	Must be an existing cost zone for item supplier expense in Merchandising, if populated.

Creating Tickets

Tickets may be created together with the creation of a new item or added to an existing item. Below are the validations:

Message Element	Required?	Notes
Ticket type ID	Always	Must be an existing ticket type in Merchandising.

Message Element	Required?	Notes
PO print type	Optional	Must be one of the following: A - On Approval R - On Receipt This is used to indicate when ticket requests will be created for the item.
Print on price change Indicator	Always	Valid values are Y and N.
Ticket over percentage	Optional	Must be greater than 0, if populated.

Updating Tickets

In order to update tickets, the item/ticket type ID must exist in Merchandising, otherwise, an error will be returned. All fields identified in the create section above except for ticket type ID are updateable and will go through the same validation as in the creation of item tickets.

Deleting Tickets

In order to delete item tickets, the item/ticket type ID must exist in Merchandising, otherwise, an error will be returned.

Table 3-49 Item Ticket Delete

Message Element	Required?	Notes
Ticket type ID	Always	Must be an existing ticket type for the item in Merchandising.

Creating Up-charges

Upcharges may be created together with the creation of a new item or added to an existing item. Below are the validations:

Table 3-50 Header Level Validation

Message Element	Required?	Notes
From location type	Always	Valid values are store (S), warehouse (W), physical warehouse (PW), area (A), region (R), country (C).
From location	Optional	Valid location or hierarchy ID.
To location type	Always	Valid values are store (S), warehouse (W), physical warehouse (PW), area (A), region (R), country (C).
To location	Optional	Valid location or hierarchy ID.

Table 3-51 Detail Level Validation

Message Element	Required?	Notes
Component ID	Always	Component ID must exist on the ELC components table.
Component rate	Always	
Per count	Optional	Must be greater than 0.
Per count UOM	Optional	Must be an existing UOM in Merchandising.
Up charge group	Always	Must be an existing Upcharge Group in Merchandising.
Component currency	Always	Must be an existing currency in Merchandising.
Transfer allocation default indicator	Optional	

Deleting Up-charges

In order to delete item upcharges, the upcharge or upcharge/component ID must exist for the item in Merchandising, otherwise, an error will be returned.

Table 3-52 Item Up-charges Header Delete

Message Element	Required?	Notes
From location type	Always	Valid values are store (S), warehouse (W), physical warehouse (PW), area (A), region (R), country (C).
From location	Optional	Location or hierarchy ID from which upcharges are being deleted.
To location type	Always	Valid values are store (S), warehouse (W), physical warehouse (PW), area (A), region (R), country (C).
To location	Optional	Location or hierarchy ID to which upcharges are being deleted.
Detail Up-charges	Optional	Child node

Table 3-53 Item Up-charges Header Delete

Message Element	Required?	Notes
Component ID	Always	Component ID must exist on the item up-charges in Merchandising.
Transfer allocation default indicator	Optional	

Updating Up-charges Detail

In order to update an item upcharge detail, the upcharge/component ID must exist for the item in Merchandising, otherwise, an error will be returned.

Creating Item Country

For GTAX tax type, item country can be created together with the creation of a new item or added to an existing item. If not provided, country is defaulted. Below are the validations:

Message Element	Required?	Notes
Country ID	Always	Must be an existing country in Merchandising.

Deleting Item Country

In order to delete an item country, the item country must exist in Merchandising, otherwise, an error will be returned.

Table 3-54 Item Country Delete

Message Element	Required?	Notes
Country ID	Always	Must be an existing country for the item in Merchandising.

Flex Attributes

If custom flex attributes (CFAS) have been defined for items, at item or item/supplier or item/supplier/country or item/supplier/country/location level, then they can be integrated as part of this API. The node of the integration that supports this will accept the name of the attribute as it is defined in the group set level view and the value for the attribute. Flex attributes can only be added to or updated on an item at header and detail levels but cannot be deleted.

Message Element	Required?	Notes
Name	Always	Holds the name of the attribute.
Value	Optional	Holds the value of the attribute for non-date attributes.
Value Date	Optional	Holds the value of the attribute for date attributes.

Translated Language

If translations have been defined for items, at item or item/image or item/supplier level, then they can be integrated as part of this API. The node of the integration that supports this will accept the language and the description specified in the language. Translations can be added, updated or deleted.

Table 3-55 Item Header Translation Create or Update

Message Element	Required?	Notes
Language	Always	Contains the language in which the translated text is maintained. Must be a valid language in Merchandising.
Short Description	Optional	Contains the translated text for the item short description.
Item Description	Always	Contains the translated text for the item description.
Item Description Secondary	Optional	Contains the translated text for the secondary description of the item.

Table 3-56 Item Header Translation Delete

Message Element	Required?	Notes
Language	Always	Contains the language of the translated text that needs to be deleted.

Table 3-57 Item Supplier Translation Create or Update

Message Element	Required?	Notes
Language	Always	Contains the language in which the translated text is maintained. Must be a valid language in Merchandising.
Supplier Differentiator 1	Optional	Contains the translated text for supplier differentiator 1.
Supplier Differentiator 2	Optional	Contains the translated text for supplier differentiator 2.
Supplier Differentiator 3	Optional	Contains the translated text for supplier differentiator 3.
Supplier Differentiator 4	Optional	Contains the translated text for supplier differentiator 4.
Supplier Label	Optional	Contains the translated text for the supplier label.

Table 3-58 Item Supplier Translation Delete

Message Element	Required?	Notes
Language	Always	Contains the language of the translated text that needs to be deleted.

Table 3-59 Item Image Translation Create or Update

Message Element	Required?	Notes
Language	Always	Contains the language in which the translated text is maintained. Must be a valid language in Merchandising.
Image Description	Always	Contains the translated text for the item image description.

Table 3-60 Item Image Translation Delete

Message Element	Required?	Notes
Language	Always	Contains the language of the translated text that needs to be deleted.

Error Handling

If any errors are encountered in the validations described above or any of the message structure validations, a status of E is returned to the external system along with the appropriate error message. If the message has been successfully persisted, a success status (S), is returned to the external system indicating that the message has been successfully received and persisted to the Merchandising database.

Message XSD

Below are the filenames that correspond with each message type. Please consult the *Oracle Retail Integration Guide* for each message type for the details on the composition of each message.

Message Types	Message Type Description	XML Schema Definition (XSD)
xitemcre	Item Create Message	XItemDesc.xsd
xitemmod	Item Modify Message	XItemDesc.xsd
xitemdel	Item Delete Message	XItemRef.xsd
xitemsupcre	Item/Supplier Create Message	XItemDesc.xsd
xitemsupmod	Item/Supplier Modify Message	XItemDesc.xsd
xitemsupdel	Item/Supplier Delete Message	XItemRef.xsd
xitemsupctycre	Item/Supplier/Country Create Message	XItemDesc.xsd
xitemsupctymod	Item/Supplier/Country Modify Message	XItemDesc.xsd
xitemsupctydel	Item/Supplier/Country Delete Message	XItemRef.xsd
xiscmfrcr	Item/Supplier/Country of Manufacture Create Message	XItemDesc.xsd
xiscmfrrmod	Item/Supplier/ Country of Manufacture Modify Message	XItemDesc.xsd
xiscmfrrdel	Item/Supplier/ Country of Manufacture Delete Message	XItemRef.xsd

Message Types	Message Type Description	XML Schema Definition (XSD)
xiscdimcre	Item/Supplier/Country/Dimension Create Message	XItemDesc.xsd
xiscdimmod	Item/Supplier/Country/Dimension Modify Message	XItemDesc.xsd
xiscdimdel	Item/Supplier/Country/Dimension Delete Message	XItemRef.xsd
xitemvatcre	Item/Vat Create Message	XItemDesc.xsd
xitemvatdel	Item/Vat Delete Message	XItemRef.xsd
xitemctrycre	Item/Country Create Message	XItemCtryDesc.xsd
xitemctrydel	Item/Country Delete Message	XItemCtryRef.xsd
xitemudacre	Item/UDA Create Message	XItemDesc.xsd
xitemudadel	Item/UDA Delete Message	XItemRef.xsd
xitemimagecre	Item/Image Create Message	XItemDesc.xsd
xitemimagemod	Item/Image Modify Message	XItemDesc.xsd
xitemimagedel	Item/Image Delete Message	XItemRef.xsd
xitemtlcre	Item Master translated language Create Message	XItemDesc.xsd
xitemtlmod	Item Master translated language Modify Message	XItemDesc.xsd
xitemtlidel	Item Master translated language Delete Message	XItemRef.xsd
xitemsuplcre	Item/Supplier translated language Create Message	XItemSupDesc.xsd
xitemsuplmod	Item/Supplier translated language Modify Message	XItemSupDesc.xsd
xitemsuplidel	Item/Supplier translated language Delete Message	XItemSupRef.xsd
xitemimagetlcre	Item/Image translated language Create Message	XItemImageDesc.xsd
xitemimagetlmod	Item/Image translated language Modify Message	XItemImageDesc.xsd
xitemimagetlidel	Item/Image translated language Delete Message	XItemImageRef.xsd
xitemhtscre	Item/HTS create message	XItemDesc.xsd
xitemhtsmod	Item/HTS modify message	XItemDesc.xsd
xitemhtsdel	Item/HTS delete message	XItemRef.xsd
xitemhtsassesscre	Item/HTS assess create message	XItemDesc.xsd
xitemhtsassessmod	Item/HTS assess modify message	XItemDesc.xsd
xitemhtsassessdel	Item/HTS assess delete message	XItemRef.xsd
xitemexpensescre	Item/Expenses create message	XItemDesc.xsd
xitemexpensesmod	Item/Expenses modify message	XItemDesc.xsd

Message Types	Message Type Description	XML Schema Definition (XSD)
xitemexpensesdel	Item/Expenses delete message	XItemRef.xsd
xitemticketcre	Item/Ticket create message	XItemDesc.xsd
xitemticketmod	Item/Ticket modify message	XItemDesc.xsd
xitemticketdel	Item/Ticket delete message	XItemRef.xsd
xitemseasoncre	Item/Seasons create message	XItemDesc.xsd
xitemseasondel	Item/Seasons delete message	XItemRef.xsd
xitemchgcre	Item up charge create message	XItemDesc.xsd
xitemchgdtlmod	Item up charge detail modify message	XItemDesc.xsd
xitemchgdel	Item up charge delete message	XItemRef.xsd

Location Trait Subscription API

This section describes the location trait subscription API.

Functional Area

Foundation Data

Business Overview

The Location Trait Subscription API processes incoming data from an external system to create, edit and delete location traits in Merchandising. This data is processed immediately upon message receipt so success or failure can be communicated to the external application.

The table below contains the details of the message as well as the validations.

Table 3-61 Location Traits

Message Element	Required?	Notes
Trait ID	Yes	This field contains the unique id number of the location trait.
Trait Description	Yes	This field contains the description of the location trait.

Error Handling

If any errors are encountered in the validations described above or any of the message structure validations, a status of E is returned to the external system along with the appropriate error message. If the message has been successfully persisted, a success status (S), is returned to the external system indicating that the message has been successfully received and persisted to the Merchandising database.

Message XSD

Here are the filenames that correspond with each message type. Please consult RIB documentation for each message type in order to get a detailed picture of the composition of each message.

Message Type	Message Type Description	XML Schema Definition (XSD)
xlocrtcre	External Location Trait Create	XLocTrtDesc.xsd
xlocrttdel	External Location Trait Delete	XLocTrtRef.xsd
xlocrtmod	External Location Trait Modification	XLocTrtDesc.xsd

Merchandise Hierarchy Reclassification Subscription API

This section describes the merchandise hierarchy reclassification subscription API.

Functional Area

Merchandise Hierarchy

Business Overview

This API allows Merchandising to subscribe merchandise hierarchy reclassification messages, that are published by an external system. It is intended to be used by retailers who manage their hierarchies in a system outside Merchandising. This API allows for pending merchandise hierarchy reclassification events to be created, modified or deleted. A separate batch process will read the information off the pending merchandise hierarchy table and creates or modifies the merchandise hierarchy information in Merchandising once the effective date arrives.

Creating Merchandise Hierarchy Reclassifications

When a new merchandise hierarchy reclassification is created, the API will first validate that all required fields are present in the message. Certain of the fields are required regardless of hierarchy level, while others are dependent on other hierarchy configurations. After that, business level validation on the input information will be performed. The tables below summarizes the validation.

Table 3-62 Header Level Validation

Message Element	Required?	Notes
Merchandise Hierarchy Level	Always	Indicates the level of merchandise hierarchy. Valid values are V (division), G (group), D (department), C (class), and S (subclass).
Merchandise Hierarchy ID	Always	Holds the merchandise hierarchy ID, of the merchandise hierarchy component being created or updated.

Table 3-62 (Cont.) Header Level Validation

Message Element	Required?	Notes
Merchandise Hierarchy parent ID	Always	This field will hold the parent of the hierarchy identified in the Merchandise Hierarchy ID field. This column will only be populated if the Merchandise Hierarchy level is class or subclass.
Merchandise Hierarchy Grandparent ID	Conditional	This field will hold the grandparent ID of the hierarchy identified in the Merchandise Hierarchy ID field. This column will only be populated if the Merchandise Hierarchy Level is subclass.
Merchandise Hierarchy Name	Always	The name of the hierarchy value.
Effective Date	Always	The date the hierarchy change will become effective. The effective date must be greater than or equal to the current date.
Action Type	Conditional	Indicates if this field is an addition (A) or modification (M). It is required on a create message and should not be populated on a modify message.
Buyer	Conditional, Optional	The number of the buyer associated with the entity. This value must be predefined in Merchandising. This field should only hold a value if the hierarchy level indicates division, group, or department.
Purchase Type	Conditional	The code indicates whether items in the department will be created by default as normal merchandise (0), consignment (1), or concession (2). This field is required if the hierarchy level indicates department, otherwise it should be null. Additionally, if the Consignment/Concession system option is set to N, this should always be 0.
Total Market Amount	Optional	This field stores total market amount that is expected for the entity. This field will only be used if the hierarchy value indicates division or department.
Merchandiser	Conditional, Optional	This field indicates the number of the merchandiser associated with the entity. This value must be predefined in Merchandising. This field should hold a value only if the hierarchy level indicates division, group, or department.
Budgeted Markup Percentage	Conditional	This field stores the markup percent of cost. Budgeted Markup Percentage or Budgeted Intake Percentage cannot be both null or both have values. This field is required if the hierarchy level indicates department, otherwise it should be null.
Profit Calculation Type	Conditional	Indicates whether profit will be calculated by direct cost (1) or retail inventory (2). This field is required for a new department, otherwise it should be null. A Department cannot be set up as profit calculation type of Direct Cost and purchase type of Consignment Stock.
Markup Calculation Type	Conditional	Indicates how markup is calculated in the department. Valid values are for a new department, otherwise it should be null.

Table 3-62 (Cont.) Header Level Validation

Message Element	Required?	Notes
OTB Calculation Type	Conditional	Indicates how open to buy is calculated in the department. Valid values are cost (C) and retail (R). This field is required for a new department, otherwise it should be null.
Maximum Average Counter	Conditional	The maximum count of days with acceptable data to include in an average for items with the department. This field is required for a new department, otherwise it should be null. The value cannot be a negative.
Average Tolerance Percentage	Conditional	The tolerance percentage value used in averaging for items within a department. This field will only be used for a new department. The value cannot be a negative.
Budgeted Intake Percentage	Conditional	Indicates the markup percent of retail to use as a default for the department. Budgeted Markup Percentage or Budgeted Intake Percentage cannot be both null or both have values. This field is required for a new department, otherwise it should be null.
Department VAT Inclusive Indicator	Conditional	Indicates the default value for the class VAT indicator. When classes are initially set up, they will inherit this value. This field will only be populated when the hierarchy level indicates department.
Class VAT Indicator	Conditional	Indicates if retail is displayed and held with or without VAT for items within a class. Valid values are Y (yes) and N (no). This field is required if the hierarchy level indicates department and you are configured for Simple VAT or Global Tax and must be set to Y in those cases. If you are configured for US Sales Tax, then it must be N.

Updating Merchandise Hierarchy Reclassifications

For updating a previously created reclassification, the hierarchy type must be already present in Merchandising. For updates, the validation is similar to that described above for creating a new reclassification. If updating the effective date of a reclassification that has an Add action type, there should not be any child hierarchy with earlier effective date. For example, if you are adding a department, there cannot be a reclassification for adding a class in the department with an earlier effective date.

Deleting Merchandise Hierarchy Reclassifications

To delete a previously created reclassification event, the below validations will be executed to ensure there are no conflicts, along with checks for the existence of child reclassification records, before deleting the record.

Message Element	Required?	Notes
Merchandise Hierarchy Level	Always	Indicates the level of merchandise hierarchy. Valid values are V (division), G (group), D (department), C (class), and S (subclass).
Merchandise Hierarchy ID	Always	Holds the merchandise hierarchy ID for the selected level.
Merchandise Hierarchy parent ID	Optional	This field will hold the parent of the hierarchy identified in the Merchandise Hierarchy ID field. This column will only be populated if the Merchandise Hierarchy Level is class or subclass.
Merchandise Hierarchy Grandparent ID	Optional	This field will hold the grandparent ID of the hierarchy identified in the Merchandise Hierarchy ID field. This column will only be populated if the Merchandise Hierarchy Level is subclass.

Error Handling

If any errors are encountered in the validations described above or any of the message structure validations, a status of E is returned to the external system along with the appropriate error message. If the message has been successfully persisted, a success status (S), is returned to the external system indicating that the message has been successfully received and persisted to the Merchandising database.

Message XSD

Below are the filenames that correspond with each message type. Please consult the Oracle Retail Integration Guide for each message type for the details on the composition of each message.

Message Types	Message Type Description	XML Schema Definition (XSD)
xmrchhrclscre	Create Merchandise Hierarchy Reclassification	XMrchHrRclsDesc.xsd
xmrchhrclsmod	Modify Merchandise Hierarchy Reclassification	XMrchHrRclsDesc.xsd
xmrchhrclsdel	Delete Merchandise Hierarchy Reclassification	XMrchHrRclsRef.xsd

Merchandise Hierarchy Subscription API

This section describes the merchandise hierarchy subscription API.

Functional Area

Merchandise Hierarchy

Business Overview

The merchandise hierarchy allows the retailer to create the relationships that are necessary to support the product management structure of a company. This hierarchy reflects a classification of merchandise into multi-level descriptive categorizations to facilitate the planning, tracking, reporting, and management of merchandise within the company. If Merchandising is not the system of record for merchandise hierarchy information, then this API may be used to create, update, or delete elements of the merchandise hierarchy, including division, group, department, class, and subclass, based on an external system.

Deleting a division or group will process immediately upon receipt of the message, assuming there are no dependent levels below them. However, departments, classes, and subclasses will not actually be deleted from the system upon receipt of the message. Instead, they will be added to a table, where a background process (Daily Purge of Foundation Data) will occur to ensure the records can be deleted as part of the delete process. For more on this batch process, see the *Retail Merchandising System Operations Guide, Volume 1 - Batch Overviews and Designs*.

If you are implementing Merchandising with Simple VAT as the default tax type, then department-level VAT records can be created and edited within the department message (VAT records are not deleted). VAT creates can be passed in with a department create message, or they can be passed in with their own specific message type. VAT region and VAT codes records must exist prior to creating department VAT records. Also, when passing in a new VAT region to an existing department with attached items, the VAT information will default to all items.

The merchandise hierarchy must be created from the highest level down. Conversely, the hierarchy must be deleted from the lowest level up. Each lower level references a parent level. This means a department is associated with a group; a class is associated with a department; and a subclass is associated with department/class combination because classes are not unique across departments.

Creating a Company

When a new company is created, this API will first validate that all required fields are present in the message. Business-level validation on the input information will be performed, which checks that the company doesn't already exist. If the validation is met, the company in the message data is created. Only one company can exist in Merchandising.

Updating a Company

When a company is updated, this API will first validate that all required fields are present in the message. Business level validation on the input information will be performed, which verifies if the company ID to be updated already exists. If the company already exists, the company's details in the message data are updated.

The format and validation for creating and updating the company is shown in the table below:

Table 3-63 Company

Message Element	Required?	Notes
Company	Yes	This field contains the unique number which identifies the company for which the system is running.

Table 3-63 (Cont.) Company

Message Element	Required?	Notes
Company Name	Yes	This field contains the name of the company for which the system is running.
Add_1	Yes	This value contains the first line of the company headquarters address.
Add_2	No	This value contains the second line of the company headquarters address.
Add_3	No	This value contains the third line of the company headquarters address.
City	Yes	This field contains the city of the company headquarters.
State	No	This field contains the state code of the company headquarters. This value must be a valid state in Merchandising.
Country Code	Yes	This field contains the country code of the company headquarters. This value must be a valid country in Merchandising.
Postal Code	No	This field contains the postal code of the company headquarters.

Creating Divisions

When a new division is created, this API will first validate that all required fields are present in the message. Business-level validation on the input information will be performed. The business validation:

- Verifies division is not already present.
- Verifies that, if total market amount is received, then it should be at least 1000.

If all the validations are met, the division in the message data is created.

Updating Divisions

When a division is updated, this API will first validate that all required fields are present in the message. Business-level validation on the input information will be performed. The business validation:

- Verifies division is present.
- Verifies that, if total market amount is received, then it should be at least 1000.

If all the validations are met, the division's details are updated.

The format and validation for creating and updating divisions is shown in the table below:

Table 3-64 Division

Message Element	Required?	Notes
Division	Yes	This field contains the unique identifier of the division.
Division Name	Yes	This field contains the name of the division.
Merchant	No	This field contains the number of the merchant associated with the division. This value must be a valid merchant in Merchandising.
Buyer	No	This field contains the number of the buyer associated with the division. This value must be a valid buyer in Merchandising.
Total Market Amount	No	This field contains the total market amount that is expected for the division. If this field is not null it must be at least 1000.

Deleting Divisions

When a division is deleted, this API will first validate that all required fields are present in the message. Business-level validation on the input information will be performed to:

- Verify the division already exists.
- Verify the division is not associated to a diff group.
- Verify the division is not associated to a season id.
- Verify the division is not associated to a ticket type.
- Verify the division is not associated to a UDA.

If the information passes these validations, the division will be added to a purging staging table for processing in the Daily Purge of Foundation Data process.

The format and validation for deleting division is shown in the table below:

Table 3-65 Division

Message Element	Required?	Notes
Division	Yes	This field contains the unique identifier of the division being deleted.

Creating Groups

When a new group is created, this API will first validate that all required fields are present in the message. Business-level validation on the input information will be performed, which checks whether the group already exists. If it does not exist, the group in the message data is created.

Updating Groups

When a group is updated, this API will first validate that all required fields are present in the message. Business-level validation on the input information will be performed, which verifies

whether the group to be updated already exists. If group already exists, the group details are updated.

The format and validation for creating and updating divisions is shown in the table below:

Table 3-66 Groups

Message Element	Required?	Notes
Group Number	Yes	This field contains the number which uniquely identifies the group.
Group Name	Yes	This field contains the name of the group.
Division	Yes	This field contains the identifier of the division of which the group is a member. This value must be a valid division in Merchandising.
Merchant	No	This field contains the number of the merchant associated with the division. This value must be a valid merchant in Merchandising.
Buyer	No	This field contains the number of the buyer associated with the division. This value must be a valid buyer in Merchandising.

Deleting Groups

When a group is deleted, this API will first validate that all required fields are present in the message. Business-level validation on the input information will be performed to:

- Verify the group already exists.
- Verify the group is not associated to a diff group.
- Verify the group is not associated to a season id.
- Verify the group is not associated to a ticket type.
- Verify the group is not associated to a UDA.

If the information passes these validations, the group will be added to a purging staging table for processing in the Daily Purge of Foundation Data process.

The format and validation for deleting groups is shown in the table below:

Table 3-67 Division

Message Element	Required?	Notes
Group Number	Yes	This field contains the unique identifier of the group being deleted.

Creating Departments

When a new department is created, this API will first validate that all required fields are present in the message. Business-level validation on the input information will be performed. The business validation:

- Verifies the department is not already present.

- Verifies if total market amount is received then it should be at least 1000.
- Verifies the child messages, if included, have their required fields present. The child messages contain the VAT and upcharge details for a department.

If all the validations are met, the department in the message data is created. Custom flex attributes can also be created for the department through this API, if they are active for the department.

Updating Departments

When a department is updated, this API will first validate that all required fields are present in the message. Business-level validation on the input information will be performed. The business validation:

- Verifies if the department is present.
- Verifies if total market amount is received then it should be at least 1000.
- Verifies the child messages, if included, contain all required fields. The child messages contain VAT and upcharge details for a department.

If all the validations are met, the department in the message data is created. Like with create, custom flex attributes can also be updated for the department, if active.

The format and validation for creating and updating departments is shown in the table below:

Table 3-68 Department

Message Element	Required?	Notes
Department	Yes	This contains the number which uniquely identifies the department.
Department Name	Yes	This contains the name of the department being created or updated.
Buyer	No	This field contains the number of the buyer associated to the department. This value must be a valid buyer in Merchandising.
Purchase Type	Yes	This field contains the code which indicates whether items in this department are normal merchandise (0) consignment stock (1) and concession stock (2).
Total Market Amount	No	This field contains the total market amount that is expected for the department. This value cannot be less than 1000.
Merchandiser	No	This field contains the number of the merchandiser that is associated to the department. This value must be a valid merchant in Merchandising.
Group Number	Yes	This field contains the number of the group to which the department belongs. This value must be a valid group in Merchandising
Budgeted Markup	No	This field contains the budgeted markup percentage, the markup percent of cost. If this value is not populated on the message, it will be calculated to be the inverse of the budgeted intake percentage. This column will hold 70% as 70, not .70. This field must have a value if Budgeted Intake is NULL.

Table 3-68 (Cont.) Department

Message Element	Required?	Notes
Profit Calc Type	Yes	This field contains the number which indicates whether profit will be calculated by direct cost (1) or retail inventory (2). It determines the accounting method to be used in the stock ledger for this department.
Markup Calc Type	Yes	This field contains the code that indicates how markup is calculated in this department. Valid values are cost (C) and retail (R).
OTB Calc Type	Yes	This field contains the code which indicates how open to buy (OTB) is calculated for this department. Valid values are cost (C) and retail (R).
Maximum Average Counter	Yes	This field contains the maximum count of days with acceptable data to include in an average for items within the department. The value should be greater than zero.
Average Tolerance Percentage	Yes	This field contains the tolerance percentage value used in averaging for items within this value. This column will hold 70% as 70, not .70. The value should be greater than zero.
Budgeted Intake	No	This field contains the budgeted intake percentage, which is the percent of the total take that is income. If this field is not populated on the message, it will be calculated as the inverse of the budgeted markup percentage. This column will hold 70% as 70, not .70. This field must have a value if Budgeted Markup is NULL.
Department VAT	No	Child node
Custom Flex Attributes	No	Child node
Department Up-Charges	No	Child node

Deleting Departments

When a department is deleted, this API will first validate that all required fields are present in the message. Business-level validation on the input information will be performed to:

- Verify the department already exists.
- Verify the department is not associated to a diff group.
- Verify the department is not associated to a season id.
- Verify the department is not associated to a ticket type.
- Verify the department is not associated to a UDA.

If the information passes these validations, the department will be added to a purging staging table for processing in the Daily Purge of Foundation Data process. Flex attributes are deleted when the department is deleted.

The format and validation for deleting departments is shown in the table below:

Table 3-69 Department

Message Element	Required?	Notes
Department	Yes	This contains the number of the department being deleted.
Department Up-Charges	No	Child node

Creating VAT Information for Departments

If you are configured to run Merchandising using Simple VAT (SVAT) for your default tax type, then you can set the VAT rates by region for the department using this API. If included, this API will check for all required fields in the message and create the VAT record for a department. When adding new VAT region to an existing department with attached items, the VAT information will default to all items.

Updating VAT Information for Departments

If you are configured to run Merchandising using Simple VAT (SVAT) for your default tax type, then you can update the VAT rates by region for the department using this API. This API will check for all required fields in the message and updates the VAT information for a department. When updating VAT details for a department with attached items, the VAT information will default to all items.

The format and validation for creating and updating VAT information is shown below:

Table 3-70 Department VAT

Message Element	Required?	Notes
VAT Region	Yes	This contains the number of the VAT region to which this department is associated. This value must be a valid VAT Region in Merchandising.
VAT Code	Yes	This field contains the alphanumeric identifier of the VAT code. This value must be a valid VAT Code in Merchandising.
VAT Type	Yes	This field indicates if the VAT rate is used for purchasing (C), selling (R), or both (B).
Reverse VAT Indicator	No	This field indicates if items in the department are subject to reverse charge VAT at the region. Valid values are Yes (Y) and No (N).

Creating Up-charges for Departments

When a message contains up-charge details, first it validates for the required fields, including "from" locations and "to" locations in the message. If no up-charge record is found, this message creates the up-charge for a department and from/to location combination. As part of the addition, you can indicate in the message if you want to have the up-charges added to existing items or only added for new items. Similarly, there is a flag in the message to indicate whether the new upcharges will be cascaded to transfers and allocations which are unshipped and not in closed or deleted status. The department upcharges will be created as soon as the message is consumed, however the new upcharges will be cascaded to items, transfers, and allocations via batches which runs at the end of every day.

Updating Up-charges for Departments

When a message contains up-charge details, first it validates for the required field in the message. If up-charge record exists for a department and the from/to location combination in the message, then the up-charge details are updated for the department. As part of the update there is also an option to have the up-charges updated for items in the department, or unshipped transfers and allocations for items in the department. The department upcharges will be updated as soon as the message is consumed, however the updates will be cascaded to items, transfers, and allocations via batches which runs at the end of every day.

The format and validation for creating and updating up-charge information is shown in the table below:

Table 3-71 Department Up-Charge Header

Message Element	Required?	Notes
From Location	Conditional	This field contains the source location from which goods will be transferred. This column can contain Country/Area/Region IDs when From Location Type is Country (C), Area (A), or Region (R). It will be a store, virtual warehouse, or physical warehouse when To Location Type is 'S', 'W' or 'PW'. Otherwise, it should be left blank when To Location Type is either All Stores (AS) or All Warehouses (AW).
To Location	Conditional	This field contains the destination location to which goods will be transferred. This column can contain Country/Area/Region IDs when To Location Type is Country (C), Area (A), or Region (R). It will be a store, virtual warehouse, or physical warehouse when To Location Type is 'S', 'W' or 'PW'. Otherwise, it should be left blank when To Location Type is either All Stores (AS) or All Warehouses (AW).
From Location Type	Yes	This field contains the type of source location from which goods will be transferred. Valid values are: <ul style="list-style-type: none"> • C - Country • A - Area • R - Region • S - store • W - Virtual Warehouse • PW - Physical Warehouse • AS - All Stores • AW - All Warehouses

Table 3-71 (Cont.) Department Up-Charge Header

Message Element	Required?	Notes
To Location Type	Yes	This field contains the type of destination location to which goods will be transferred. Valid values are: <ul style="list-style-type: none"> • C - Country • A - Area • R - Region • S - store • W - Virtual Warehouse • PW - Physical Warehouse • AS - All Stores • AW - All Warehouses
Department Up-Charge Detail	No	Child node

Table 3-72 Department Up-Charge Detail

Message Element	Required?	Notes
Component ID	Yes	This field contains the unique identifier of the Up Charge component. This should be a valid cost component in merchandising.
Component Rate	Yes	This field contains the rate to be charged against the cost of the Item/To Location combinations within the department.
Per Count	Yes	This field contains a count indicating the amount of the Per Count Unit of Measure to which the rate applies.
Per Count UOM	Yes	This field contains the unit of measure in which the Per Count is specified. This should be a valid unit of measure in merchandising.
Up-Charge Group	Yes	This field contains the group to which the component ID belongs. Valid values can be found on the codes table.
Component Currency	Yes	This field contains the currency of the Up Charge component. This must be a valid currency in Merchandising.
Effective Date	No	This field contains the date from which the new values are effective in the system.
Item Default Indicator	No	This field indicates if component rate information is updated or not for existing items under the department. Valid values are Yes (Y) and No (No).
Transfer Allocation Default Indicator	No	This field indicates if component rate information is updated or not for existing transfers and allocations under the department. Valid values are Yes (Y) and No (N).

Deleting Up-charges for Departments

When a message contains an up-charge, first it validates for the required field in the message. If up-charge record exists for a department, this message deletes the upcharge

details for a department. Deleting up-charges from a department does not automatically remove them from the items or transfers and allocations for items in the department.

The format and validation for deleting up charges is shown in the table below:

Table 3-73 Department Up-Charge Detail

Message Element	Required?	Notes
Component ID	Yes	This field contains the unique identifier of the Up Charge component being deleted.

Creating Classes

When a new class is created, this API will first validate that all required fields are present in the message. Business-level validation on the input information will be performed, which checks whether the class already exists. If the class does not exist, the class in the message data is created. Custom flex attributes can also be created for the class through this API, if they are active for the class.

Updating Classes

When a class is updated, this API will first validate that all required fields are present in the message. Business-level validation on the input information will be performed, which verifies if the class to be updated already exists. If class already exists, the class details are updated. Like with create, custom flex attributes can also be updated for the class, if active.

The format and validation for creating and updating classes is shown in the table below:

Table 3-74 Class

Message Element	Required?	Notes
Class	Yes	This field contains the unique number of the class.
Class Name	Yes	This field contains the name of the class.
Class VAT Indicator	Yes	This field indicates whether retail is displayed and held with or without VAT for items within the class. This field is available when VAT is on in the system and defined at the class level. This field is always defaulted to No (N).
Department	Yes	This field contains the number of the department which contains the class. This value must be a valid department in Merchandising.
Custom Flex Attributes	No	Child node

Deleting Classes

When a class is deleted, this API will first validate that all required fields are present in the message. Business-level validation on the input information will be performed to verify:

- The class exists.
- It is not associated to a diff group.
- It is not associated to a season.
- It is not associated to a ticket type.
- It is not associated to a UDA.

If these validations pass, the class will be added to a purging staging table for processing in the Daily Purge of Foundation Data process. Flex attributes are deleted when the class is deleted.

The format and validation for deleting classes is shown in the table below:

Table 3-75 Class

Message Element	Required?	Notes
Class	Yes	This field contains the unique number of the class.
Department	Yes	The number of the department which contains the class. This value must be a valid department in Merchandising.

Creating Subclasses

When a new subclass is created, this API will first validate that all required fields are present in the message. Business-level validation on the input information will be performed, which checks whether the subclass already exists. If the subclass does not exist, then it is created. Custom flex attributes can also be created for the subclass through this API, if they are active for the subclass.

Updating Subclasses

When a subclass is updated, this API will first validate that all required fields are present in the message. Business-level validation on the input information will be performed which verifies if the subclass to be updated already exists. If subclass already exists, it is updated. Like with create, custom flex attributes can also be updated for the subclass, if active.

The format and validation for creating and updating subclasses is shown in the table below:

Table 3-76 Subclass

Message Element	Required?	Notes
Subclass	Yes	This field contains the unique number of the subclass.
Subclass Name	Yes	This field contains the name of the subclass.
Class	Yes	This field contains the number of the class which contains the subclass. This value must be a valid class in Merchandising.
Department	Yes	This field contains the number of the department which contains the class. This value must be a valid department in Merchandising.
Custom Flex Attributes	No	Child node

Deleting Subclasses

When a subclass is deleted, this API will first validate that all required fields are present in the message. Business-level validation on the input information will be performed to:

- Verify the subclass already exists.
- Verify the subclass is not associated to a diff group.
- Verify the subclass is not associated to a season id.
- Verify the subclass is not associated to a ticket type.
- Verify the subclass is not associated to a UDA.

If these validations pass, the subclass will be added to a purging staging table for processing in the Daily Purge of Foundation Data process. Flex attributes are deleted when the subclass is deleted.

The format and validation for deleting classes is shown in the table below:

Table 3-77 Subclass

Message Element	Required?	Notes
Subclass	Yes	This field contains the unique number of the subclass being deleted
Class	Yes	This field contains the unique number of the class.
Department	Yes	The number of the department which contains the class/subclass. This value must be a valid department in Merchandising.

Flex Attributes

If any custom flex attributes (CFAS) for the department, class, or subclass have been added or modified, it will trigger an update message. The node of the integration that supports this will contain the name of the attribute as it is defined in the group set level view, the value of the custom attribute. If it is a date attribute, the date value is in a separate field. Flex attributes can only be added to or updated; they cannot be deleted.

Table 3-78 Flex Attributes

Message Element	Required?	Notes
Name	Yes	Holds the attribute name.
Value	No	Holds the value of the attribute for number and character type attributes
Value Date	No	Holds the date for date type attributes.

Error Handling

If any errors are encountered in the validations described above or any of the message structure validations, a status of E is returned to the external system along

with the appropriate error message. If the message has been successfully persisted, a success status (S), is returned to the external system indicating that the message has been successfully received and persisted to the Merchandising database.

Message XSD

Below are the filenames that correspond with each message type. Please consult the Oracle Retail Integration Guide for each message type for the details on the composition of each message.

Message Types	Message Type Description	XML Schema Definition (XSD)
xmrchhrcompcre	External Create Company	XMrchHrCompDesc.xsd
xmrchhrcompmod	External Modify Company	XMrchHrCompDesc.xsd
xmrchhrdivcre	External Create Division	XMrchHrDivDesc.xsd
xmrchhrdivmod	External Modify Division	XMrchHrDivDesc.xsd
xmrchhrdivdel	External Delete Division	XMrchHrDivRef.xsd
xmrchhrgrpcre	External Create Group	XMrchHrGrpDesc.xsd
xmrchhrgrpmod	External Modify Group	XMrchHrGrpDesc.xsd
xmrchhrgrpdel	External Delete Group	XMrchHrGrpRef.xsd
xmrchhrdeptcre	External Create Department	XMrchHrDeptDesc.xsd
xmrchhrdeptmod	External Modify Department	XMrchHrDeptDesc.xsd
xmrchhrdeptdel	External Delete Department	XMrchHrDeptRef.xsd
Xmrchhrvatcre	External Merch Hierarchy VAT create	XMrchHrDeptDesc.xsd
xmrchhrvatmod	External Merch Hierarchy VAT modify	XMrchHrDeptDesc.xsd
xmrchhrdeptchrgcre	External Merch Hier Dept Up-Charge create	XMrchHrDeptDesc.xsd
xmrchhrdeptchrgmod	External Merch Hier Dept Up-Charge modify	XMrchHrDeptDesc.xsd
xmrchhrdeptchrgdel	External Merch Hier Dept Up-Charge delete	XMrchHrDeptRef.xsd
xmrchhrclscre	External Create Class	XMrchHrClsDesc.xsd
xmrchhrclsmod	External Modify Class	XMrchHrClsDesc.xsd
xmrchhrclsdel	External Delete Class	XMrchHrClsRef.xsd
xmrchhrsclsdel	External Delete Subclass	XMrchHrScIsRef.xsd
xmrchhrsclsmod	External Modify Subclass	XMrchHrScIsDesc.xsd
xmrchhrsclsdel	External Delete Subclass	XMrchHrScIsRef.xsd

Order Subscription API

This section describes the PO subscription API.

Functional Area

Purchase Orders

Business Overview

This subscription API is used to keep Merchandising in sync with an external system that is responsible for maintaining purchase orders. It is assumed that the source of orders sent in this API is not the supplier, as vendor managed inventory (VMI) POs can be sent using the Upload Purchase Order and Purchase Order Change Acknowledgements from Suppliers to Merchandising (ediupack) batch upload. It also does not support creating customer order POs or contract POs. Customer order POs are assumed to be sent using the Customer Order Fulfillment Subscription API and contract orders are created using replenishment processes in Merchandising or manually using the UI.

POs can be created, modified or deleted at the header or the detail level. This API also creates, edits, and deletes other data associated with a purchase order, including letter of credit, expenses, harmonized tariff schedules (HTS) and assessments, and custom flex attributes (CFAS). It also will apply rounding rules, default inventory management parameters, apply bracket costs, update open to buy buckets, and insert a record into the deals queue for deals to be applied to the order, if applicable. These transactions are performed immediately upon receipt of the message so that success or failure can be sent back to the calling application.

If the location on a purchase order is a franchise store, a corresponding franchise order is also created along with the PO.

Creating Purchase Orders

New purchase order messages pass through a series of validations such as required field and valid value validations for each field as well as business validations. The tables below summarize these validations.

Table 3-79 Header Level Validation

Message Element	Required?	Notes
Order Number	Always	Must be a unique order number not used by any existing purchase orders in Merchandising.
Supplier	Always	Must be a valid, active supplier in Merchandising. The supplier and locations on the order must belong to the same org unit. If the EDI PO indicator is set to Y, the supplier of the order must also be an EDI supplier.
Currency Code	Optional	Must be a valid currency code. If not provided, this defaults to the currency code used by the supplier.
Terms	Optional	Must be a valid payment terms in Merchandising. If terms is not provided in the message, the API will default this to supplier terms.

Table 3-79 (Cont.) Header Level Validation

Message Element	Required?	Notes
Not Before Date	Optional	Must be equal to or after the current date and before the Not After Date, if provided. If the date is not provided, the API will default the value to the current date + default supplier lead time if there are not items attached to the order. If items are already added to the order, it will be defaulted to current date + minimum lead time + minimum pickup lead time among all items in the order.
Not After Date	Optional	Must be equal to or after the current date and after the Not Before Date, if provided. If the date is not provided, the API will default the value to the current date + default supplier lead time if there are not items attached to the order. If items are already added to the order, it will be defaulted to current date + maximum lead time + maximum pickup lead time among all items in the order.
OTB End of Week Date	Optional	Must be a valid end of week date and equal to or after the current date. If the date is not provided, the API will default the value to the last day of the week that the not after date value falls in.
Department	Optional	Must be a valid department in Merchandising. The department field should not be populated if items belonging to different departments are present in the order. Department is required if the Department Level PO system option is Y.
Status	Optional	Valid statuses are Worksheet (W), Submitted (S), or Approved. If not provided, the status will be defaulted to W. Status of closed (C) also allowed, but for order updates only.
Exchange Rate	Optional	The exchange rate should be greater than zero. If not provided, and the currency code is provided, the exchange rate will be based on the given currency code and primary currency. If the currency code is not provided, then the exchange rate will default based on the supplier's currency code and primary currency.
Include On Order Indicator	Optional	Valid values are Y and N. If not provided in the message, it will be defaulted to Y.
Written Date	Optional	If not provided in the message, it will be defaulted to the current date.

Table 3-79 (Cont.) Header Level Validation

Message Element	Required?	Notes
Origin Indicator	Optional	<p>Valid values for origin indicator are:</p> <ul style="list-style-type: none"> • 0 – Merchandising generated PO (from replenishment) • 1 – Other system generated PO • 2 – Manual purchase order • 3 – Buyer worksheet PO • 4 – Consignment sales generated PO • 5 – Vendor generated PO • 6 – AIP generated PO • 7 – SIM generated PO • 8 – Allocation generated PO • 9 – Consignment transfer generated PO • 10 – Consignment ownership change generated PO <p>The expected values for this field in purchase order subscription are 1, 2, 6, 7 and 8.</p> <p>If it is not provided in the message, it will be defaulted to 2.</p>
EDI PO Indicator	Optional	Valid values are Y and N. If not provided in the message, it will be defaulted to the supplier's EDI PO indicator.
Pre Mark Indicator	Optional	Valid values are Y and N. If not provided in the message, it will be defaulted to N. If Y, then the order must be pre-allocated before it is approved.
User ID	Optional	If not passed into the message, then a value will be defaulted for auditing purposes.
Comment	Optional	
Attempt RMS Load	Optional	If not passed into the message, then it will be defaulted to RMS, which means that the information will persist to the Merchandising tables as opposed to the staging tables. Valid values are RMS and STG.
Master PO number	Optional	This is can be used for linking multiple orders together for multiple delivery date orders.
Lading Port	Optional	If not provided, this will be defaulted based on supplier import attributes. If provided, it should be a valid lading port in Merchandising.
Discharge Port	Optional	If not provided, this will be defaulted based on supplier import attributes. If provided, it should be a valid discharge port in Merchandising.
Factory	Optional	If not provided, this will be defaulted based on supplier import attributes. If provided, it should be a valid, active factory in Merchandising.
Agent	Optional	If not provided, this will be defaulted based on supplier import attributes. If provided, it should be a valid, active agent in Merchandising.

Table 3-79 (Cont.) Header Level Validation

Message Element	Required?	Notes
Ship Method	Optional	If not provided, this will be defaulted based on supplier attributes. This must be a valid ship method in Merchandising, which are stored in the codes table under the code type SHPM.
Partner Type 1	Optional	This should be provided if partner 1 is given. Valid values are S1, S2 and S3. These are stored under the codes table under the code type SUHL.
Partner 1	Optional	If not provided, this will be defaulted based on supplier import attributes. If provided, it should be a valid, active partner in Merchandising for the given partner type. Partner and partner type should be provided, or both should be null.
Partner Type 2	Optional	This should be provided if partner 2 is given. Valid values are S1, S2 and S3. These are stored under the codes table under the code type SUHL.
Partner 2	Optional	If not provided, this will be defaulted based on supplier import attributes. If provided, it should be a valid, active partner in Merchandising for the given partner type. Partner and partner type should be provided, or both should be null.
Partner Type 3	Optional	This should be provided if partner 3 is given. Valid values are S1, S2 and S3. These are stored under the codes table under the code type SUHL.
Partner 3	Optional	If not provided, this will be defaulted based on supplier import attributes. If provided, it should be a valid, active partner in Merchandising for the given partner type. Partner and partner type should be provided, or both should be null.
Payment Method	Optional	Valid values are stored under the code type PYMT in the codes table. If not provided, this will default to the supplier payment method.
Purchase Type	Optional	Valid values are stored under the code type PURT in the codes table. If not provided, this will default to the purchase type defined at the supplier inventory management level.
FOB Title Pass	Optional	This is required for import orders with payment method of Letter of Credit. If this is not provided in the message, it will default to the FOB Title Pass defined at the system level. Valid values are stored under the code type FOBT in the codes table.
FOB Title Pass Desc	Optional	This is required for import orders with payment method of Letter of Credit. If this is not provided in the message, this will default to the FOB Title Pass Description defined at the system level.
PO Type	Optional	This should be a valid PO Type in Merchandising. Valid PO Types are found in the PO_TYPE table.

Table 3-79 (Cont.) Header Level Validation

Message Element	Required?	Notes
Import Country ID	Optional	This is required for import orders. The value should be a valid country in Merchandising. If the value is not provided in the message, it will default to the country ID of the primary address of the location in the order.
Order Type	Optional	Indicates the type of order and which Open To Buy bucket will be updated. Valid values are found against code type 'ORDO'.
Buyer	Optional	This field contains the id of the buyer associated to the PO. Validation is performed against the Buyer table.
Location Type	Optional	This field contains the type of location in the location field. Valid values are S (Store) or W (Warehouse).
Location	Optional	This field contains the location where all items on the order will be delivered to. If populated, it will mean a single location order.
Promotion	Optional	Contains the RPCS offer ID associated with the order to provide a link between the order dialog and the promotions dialog.
QC Indicator	Optional	Determines whether or not quality control will be required when items for this order are received. Valid values are 'Y' and 'N'.
Freight Terms	Optional	This refers to the freight terms related to the PO.
Backhaul Type	Optional	This field contains the type of backhaul allowance that will be applied to the order.
Backhaul Allowance	Optional	This field will contain the backhaul allowance value.
Ship Pay Method	Optional	This indicates the payment terms for freight charges associated with the order. The code type 'SHMT' holds the valid values.
FOB Trans Responsibility	Optional	Contains the code indicating the type of the location that is responsible for the transportation of the order.
FOB Trans Responsibility Description	Optional	This field describes the code for the location responsible for the transportation of the order.
Vendor Order No	Optional	This field contains the vendor's unique identifying number for an order. These orders may have originated by the vendor through the EDI process or this number can be associated to an Oracle Retail order when the order is created on-line.
Freight Contract No	Optional	This field contains the number of the contract with a shipper that will give specific freight rates.
Pickup Location	Optional	This field contains the location at which the order will be picked up, if the order is a Pickup order.
Pickup No	Optional	This field contains the reference number of the Pickup order.

Table 3-79 (Cont.) Header Level Validation

Message Element	Required?	Notes
Pickup Date	Optional	This field contains the date when the order can be picked up from the Supplier. This field is only required if the Purchase Type of the order is Pickup.
Appointment Date/Time	Optional	This column will hold the date and time of the receiving appointment at the warehouse.
Import Type	Optional	This is the default importer/exporter assigned to the supplier of the Purchase order. Valid values are stored against code type 'BTLT'.
Invoice Location	Optional	This identifies the importer/exporter assigned to the supplier. It should reference the WH.WH column.
Clearing Zone	Optional	This column will hold the clearing zone ID.
Routing Location	Optional	This is the default routing location for the import order. It refers the OUTLOC.OUTLOC_ID column.
Master PO No	Optional	This field holds the Master Order Number.
Re-Approve	Optional	This field indicates that the update to the corresponding purchase order needs to be performed, and then it should be approved again.
Earliest Ship Date	Optional	The date before which the items on the purchase order cannot be shipped by the supplier. Represents the earliest earliest ship date of all the items on the order.
Latest Ship Date	Optional	The date after which the items on the purchase order cannot be shipped by the supplier. Represents the greatest latest ship date of all the items on the order.

Table 3-80 Detail Level Validation

Message Element	Required?	Notes
Item Number	Required	Must be an approved, orderable transaction level item in Merchandising supplied by the supplier indicated in the message. The Item Number should not be in the process of being deleted and must be active at the location specified in the message.
Location	Required	Must be a valid stockholding store or virtual warehouse in Merchandising.
Location Type	Required	Valid values are store (S) and warehouse (W).
Unit Cost	Optional	Must be greater than or equal to zero. If more than one virtual warehouse for the same physical warehouse are included in the details for the order, then the unit cost must be the same for those item/warehouses.
Reference Item	Optional	Must be a valid, approved reference item of the item being ordered and should not be in the process of being deleted. The Reference Item should be supplied by the supplier in the message.

Table 3-80 (Cont.) Detail Level Validation

Message Element	Required?	Notes
Origin Country ID	Optional	Must exist as a valid country for the supplier/item provided in the message. If the country is not provided in the message, the value is defaulted to the item's primary country of sourcing.
Supplier Pack Size	Optional	Must be greater than zero. If there are several order lines with the same item in the message, the supplier pack size and origin country of these records should all be the same. If not provided in the message, the API will default the value based on the supplier/country for the item.
Quantity Ordered	Required	Must be greater than zero.
Cancel Indicator	Optional	This is used for purchase order detail modification only.
Reinstate Indicator	Optional	This is used for purchase order detail modification only.
Delivery Date	Optional	The date by which goods are to be delivered.
Qty Cancelled	Optional	This field contains the cancelled quantity for the item in the order.
Cancel Code	Optional	This field contains the reason that the line item was cancelled. This field is required if a line item is cancelled.
Estimated Instock Date	Optional	Date that the item on the PO is expected to be available to ship from the PO location to another location.
Earliest Ship Date	Optional	The date before which the item cannot be shipped by the supplier. It will be validated to be a valid date equal to or greater than the business date.
Latest Ship Date	Optional	The date after which the item cannot be shipped by the supplier. It will be validated to be a valid date equal to or greater than the business date.

If the above validation passes, then the purchase order and details will be created with the status set in the message. If the status in the message is approved, then the order will also be subjected to a series of approval checks. If the order cannot be approved, it will not be created.

You can also include information on the order - letter of credit (if the payment method is letter of credit), landed cost expenses, and for import orders you can include HTS and assessments. If included, these will be created simultaneously with the creation of the order.

Updating Purchase Orders

Updates can be made either at header level or at detail level for orders that are in Worksheet, Approved or Closed status. For both kinds of update messages, the API will validate that the order number included in the message already exists in Merchandising while item number and location will be also validated for existence in detail level updates.

Header Level Updates

Only header level fields need to be provided for header-level updates. Any order details included in the message will be ignored for a header-level update message. There are certain fields that are not allowed to be updated at the header level depending on the status, and if these are still provided in the message, no error message will be returned. The values will simply be ignored. However, modifying the following header level fields is allowed while the order is submitted or approved, without having to set the order in worksheet status:

- status
- not before date
- not after date
- terms
- include on-order indicator
- comments

Detail Level Updates

Order details can be updated for orders in Approved, Worksheet, Submitted or Closed status. The only information needed at the header level is the order number, which if not provided, will cause the message to be rejected. All other details provided at the header level will be ignored. Modifying order quantity, as well as supplier pack size or unit cost on an approved or submitted order will in effect set the order status to worksheet and subject it for automatic re-approval. When modifying order quantities, the full amount should be provided, not just the difference in the old and new values. Validations are also done on quantity changes, such as the ordered quantity should not go below the allocated quantity or replenishment quantity, quantity ordered cannot not be less than quantity received.

Fields that can be modified in worksheet, submitted and approved status:

- Supplier Pack Size
- Unit Cost - for items with no received quantities
- Quantity Ordered

Fields that cannot be modified in statuses other than worksheet:

- Origin Country ID
- Location

Fields that can be modified only in approved status:

- Quantity Cancelled
- Cancel Code

Reinstating Order Lines

To reinstate orders, the reinstate indicator should be set to Y. In effect, this will set the cancelled quantities of the line items to 0 and reinstate the ordered quantities. This will set the status of the reinstated order to Worksheet.

Canceling a Line Item in an Approved Order

In order to cancel a line item on the order, you can set the cancel indicator at the detail level to Y and at the same time, the quantity ordered for that line item must be set to 0. For partial cancellations, either reduce the quantity of an approved order or populate the quantity cancelled field making sure the cancel indicator is blank or set to N. This will allow for the automatic re-approval of the entire order, if there are line items still on the order once processed by the API. The cancel indicator and reinstate indicator cannot be set to Y at the same time.

Deleting Purchase Orders

If you are deleting a line item on the purchase order or deleting the whole purchase order, the API will first validate that the order number is valid. The order number is the only required field for a header delete message. All other fields will be ignored. For detail delete messages, you must provide the item as well and optionally, the location. These should exist in Merchandising, or else a reject message will be returned.

Deleting the Entire Order

In order to delete an entire order, you must send a header delete message. This will in effect set the status of the order at the header level to D. Only worksheet orders can be deleted. Deleting the purchase order cannot be done if the order is submitted, approved or has been approved, or if allocations exist for the order. Delete messages will still be processed, however it will be treated as an update of cancelled quantity and the quantity ordered will be reduced to the quantity available to be cancelled. If this results in all line items being cancelled or if the delete is made at header level, the status of the order will become Closed.

If an order is still in worksheet status, the entire order will be deleted. If the order involves any franchise stores, then any franchise order or return created with the order will also be cancelled or deleted.

Deleting a Line Item

If an order is still in worksheet status, line items will be deleted from the order. If all line items are deleted, the order header will also be deleted. For orders that are not in worksheet status, when a detail delete is requested, it will update the quantities to cancelled quantities and will be subject for re-approval.

Table 3-81 Header Level Validation

Message Element	Required?	Notes
Order Number	Always	Must be an existing order number in Merchandising.
Order Detail	Optional	Child node
Attempt RMS Load	Optional	This field indicates if the message will persist in RMS or the staging tables. Valid values are RMS or STG. If not defined, the default is STG.
Location Expense	Optional	Child node
Item HTS	Optional	Child node

Table 3-82 Detail Validation

Message Element	Required?	Notes
Item	Optional	This field contains the item number that will be deleted in the order.
Location	Always	This field contains the location the item is ordered to.
Reference Item	Optional	The ID of a reference item which can be used instead of using the item field. If the item field is not populated this field is required.

Creating a Purchase Order Letter of Credit

A letter of credit may be created together with the creation of a new order or added to an existing order with a payment method of Letter of Credit. In order to create/edit/delete a letter of credit, the order should be in worksheet status. Below are the validations:

Table 3-83 Creating a Purchase Order Letter of Credit

Message Element	Required?	Notes
Letter of Credit Reference ID	Optional	The reference ID must be exist in Merchandising for the given beneficiary and applicant. The Free on Board title pass description and purchase type in the letter of credit table must match that of the values in the order.
Letter of Credit Group ID	Optional	If included, must be a valid value in Merchandising.
Applicant	Always	The applicant must be an active partner in Merchandising.
Beneficiary	Always	Must be a valid active supplier who can be a beneficiary.
Merchandise description	Always	A description of the merchandise being imported for customs purposes.
Transshipment Indicator	Always	Valid values are Y or N.
Letter of credit indicator	Always	Valid values are Y or N.

Updating a Purchase Order Letter of Credit

In order to update an order letter of credit, the letter of credit must exist for the order in Merchandising, otherwise, an error will be returned. All fields identified in the create section above are updateable and will go through the same validation as in the creation of a letter of credit.

Deleting a Purchase Order Letter of Credit

In order to delete an order letter of credit, the letter of credit must exist for the order in Merchandising, otherwise, an error will be returned.

Creating Expenses

Expenses may be created together with the creation of a new order or added to an existing order that has location records defined. In order to create/edit/delete expenses, the order should be in worksheet status. Below are the validations:

Table 3-84 Create Expenses

Message Element	Required?	Notes
Item	Always	The item/location/location type combination must be present in the order. For buyer packs with an order as type of Pack, this should be a component item in the pack and the item on the order should be present in the pack item field.
Pack item	Optional	If provided, the item/pack item/location/location type combination must be present in the order. This is required if the item on the order is a buyer pack with an order as type of Pack.
Location	Always	The item/location/location type combination must be present in the order.
Location type	Always	The item/location/location type combination must be present in the order. Valid values are S or W.
Component ID	Always	Must be a valid expense component in Merchandising. This should be present in ELC_COMP.
CVB code	Conditional	Required if the component rate calculation basis is value (V), otherwise this will be defaulted to NULL. Must be a valid CVB code in Merchandising.
Cost basis	Optional	Valid values are supplier (S) or order (O). If CVB code is provided, then this should be null.
Component rate	Optional	This will be defaulted based on the component if not provided.
Component currency	Optional	This will be defaulted based on the component if not provided. If it is present in the message, this must be a valid currency code.
Exchange rate	Optional	This should be the exchange rate used in relation to the location on the order. If this is not provided in the message, the API defaults it, depending on the order exchange indicator set at system level. If the indicator is Y, it defaults based on the component currency. If the component currency of the component is the same as that of the location, then exchange rate should be 1. If the system level indicator is set to N, the exchange rate will be based on the location currency.
Per count	Optional	If the component rate calculation basis is specific (S), it is defaulted based on the expensed component if not provided in the message.

Table 3-84 (Cont.) Create Expenses

Message Element	Required?	Notes
Per count UOM	Optional	If the component rate calculation basis is specific (S), it is defaulted based on the expense component if not provided in the message. This must be a valid unit of measure.
Nominal flag 1	Optional	This will be defaulted based on the expense component if not provided. If it is present in the message, this must be N,+, or -.
Nominal flag 2	Optional	This will be defaulted based on the expense component if not provided. If it is present in the message, this must be N,+, or -.
Nominal flag 3	Optional	This will be defaulted based on the expense component if not provided. If it is present in the message, this must be N, +, or -.
Nominal flag 4	Optional	This will be defaulted based on the expense component if not provided. If it is present in the message, this must be N, +, or -.
Nominal flag 5	Optional	This will be defaulted based on the expense component if not provided. If it is present in the message, this must be N,+, or -.

Updating Expenses

In order to update expenses, the order/item/location/component ID must exist for the order in Merchandising, otherwise, an error will be returned. All fields identified in the create section above except for order/item/pack item/location/component id are updateable and will go through the same validation as in the creation of expenses.

Deleting Expenses

In order to delete expenses, the order/item/location/component ID must exist for the order in Merchandising, otherwise, an error will be returned.

Table 3-85 Order Expense Deletion

Message Element	Required?	Notes
Item	Always	This field contains the item number that will be deleted in the order.
Pack Item	Optional	If provided, the item/pack item/location combination must be present in the order.
Location	Always	This field contains the location that the item was ordered to. The item/location combination must be present in the order.
Component ID	Always	Must be a valid expense component in Merchandising. This should be present in ELC_COMP.

Creating HTS and Assessments

HTS and Assessment may be created together with the creation of a new import order or added to an existing order that has location records defined. In order to create/edit/delete HTS and assessments, the order should be in worksheet status. Below are the validations:

Table 3-86 HTS Validation

Message Element	Required?	Notes
Item	Always	The item must exist on the order. For buyer packs with an order as type of Pack, this should be a component item in the pack and the item on the order should be present in the pack item field.
Pack item	Conditional	The item must exist on the order. This is required if the item on the order is a buyer pack with an order as type of Pack.
HTS	Always	Must be a valid HTS code for the supplier's import country in Merchandising
Status	Always	Valid values are Worksheet (W) or Approved (A).
Origin Country ID	Optional	Must be a valid country in Merchandising. If the HTS tracking level is based on country of manufacture, this should be a valid country of manufacture for the item/supplier. If it is not provided, it will default to the item's primary manufacturing country. If the HTS tracking level is country of sourcing, it will default to the primary sourcing country for the item/supplier.
Import Country ID	Optional	Must be a valid country in Merchandising. If not provided, this will default to the import country ID at the order level.

Table 3-87 HTS Assessment Validation

Message Element	Required?	Notes
Component ID	Always	Must be a valid assessment component ID in Merchandising.
Component rate	Optional	This will be defaulted based on the component if not provided.
Per count	Optional	This is defaulted to the component when the calculation basis is specific (S), otherwise this will be defaulted to NULL.
Per count UOM	Optional	This is defaulted to the component when the calculation basis is specific (S), otherwise this will be defaulted to NULL.
CVB code	Optional	This is defaulted to the component when the calculation basis is value (V), otherwise this will be defaulted to NULL.
Nominal flag 1	Optional	This will be defaulted based on the assessment component if not provided. If it is present in the message, this must be N, +, or -.

Table 3-87 (Cont.) HTS Assessment Validation

Message Element	Required?	Notes
Nominal flag 2	Optional	This will be defaulted based on the assessment component if not provided. If it is present in the message, this must be N, +, or -.
Nominal flag 3	Optional	This will be defaulted based on the assessment component if not provided. If it is present in the message, this must be N, +, or -.
Nominal flag 4	Optional	This will be defaulted based on the assessment component if not provided. If it is present in the message, this must be N, +, or -.
Nominal flag 5	Optional	This will be defaulted based on the assessment component if not provided. If it is present in the message, this must be N, +, or -.

Updating HTS and Assessments

In order to update HTS and assessments, the record to be updated must exist in Merchandising, otherwise, an error will be returned. Status and origin country ID can be updated at the HTS level. For assessments, all fields identified in the create section above except for component ID are updateable and will go through the same validation as in the creation of assessments.

Deleting HTS and Assessments

In order to delete HTS and assessments, expenses, the record to be deleted must exist in Merchandising, otherwise, an error will be returned.

Table 3-88 Order HTS

Message Element	Required?	Notes
Item	Always	This field contains the item number that will be deleted in the order.
Pack Item	Optional	If provided, the item/pack item combination must be present in the order.
HTS	Always	This field contains the unique identifier for the Harmonized Tariff Schedule code. The item/pack item/HTS must be present in the order.
HTS Assessments	Optional	Child node

Table 3-89 Order HTS Assessments

Message Element	Required?	Notes
Component ID	Always	This field contains the assessment component ID. The item/pack item/HTS/component ID must be present in the order.

Publishing Updates

Purchase orders will be published back to the RIB if approved or previously approved, such that system responsible for managing the purchase orders are notified.

Flex Attributes

If custom flex attributes (CFAS) have been defined for purchase orders, or at the order/item or order/item/location level, then they can be integrated as part of this API. The node of the integration that supports this will accept the name of the attribute as it is defined in the group set level view and the value for the attribute. Flex attributes can only be added to or updated on a purchase order at header and detail levels but cannot be deleted.

Table 3-90 Flex Attributes

Message Element	Required?	Notes
Name	Yes	Holds the attribute name.
Value	No	Holds the value of the attribute for number and character type attributes
Value Date	No	Holds the date for date type attributes.

Error Handling

If any errors are encountered in the validations described above or any of the message structure validations, a status of E is returned to the external system along with the appropriate error message. If the message has been successfully persisted, a success status (S), is returned to the external system indicating that the message has been successfully received and persisted to the Merchandising database.

Message XSD

Here are the filenames that correspond with each message type. Please consult the RIB documentation for each message type in order to get a detailed picture of the composition of each message.

Message Types	Message Type Description	XML Schema Definition (XSD)
XorderCre	Order Create Message	XOrderDesc.xsd
XorderCre (CustFlexAttriVo)	Order Flex Attribute Create Message for the Order Header	XOrderDesc.xsd
XorderDtlCre	Order Detail Create Message	XOrderDesc.xsd
XorderDtlCre (CustFlexAttriVo)	Order Detail Flex Attribute Create message	
XorderLCCre	Order LC Create Message	XOrderDesc.xsd
XorderLocExpCre	Order Location Expense Create Message	XOrderDesc.xsd
XorderSkuHtsCre	Order SKU HTS Create Message	XOrderDesc.xsd

Message Types	Message Type Description	XML Schema Definition (XSD)
XorderSkuHtsAssessCreate	Order SKU HTS Assess Create Message	XOrderDesc.xsd
XorderMod	Order Modify Message	XOrderDesc.xsd
XorderMod (CustFlexAttrVo)	Order Header Flex Attribute Modify Message	XOrderDesc.xsd
XorderDtIMod	Order Detail Modify Message	XOrderDesc.xsd
XorderDtIMod (CustFlexAttrVo)	Order Detail Flex Attribute Modify Message	XOrderDesc.xsd
XorderLCMod	Order LC Modify Message	XOrderDesc.xsd
XorderLocExpMod	Order Location Expense Modify Message	XOrderDesc.xsd
XorderSkuHtsMod	Order SKU HTS Modify Message	XOrderDesc.xsd
XorderSkuHtsAssessMod	Order SKU HTS Assess Modify Message	XOrderDesc.xsd
XorderDel	Order Delete Message	XOrderRef.xsd
XorderDtIDel	Order Detail Delete Message	XOrderRef.xsd
XorderLCDel	Order LC Delete Message	XOrderRef.xsd
XorderLocExpDel	Order Location Expense Delete Message	XOrderRef.xsd
XorderSkuHtsDel	Order SKU HTS Delete Message	XOrderRef.xsd
XorderSkuHtsAssessDel	Order SKU HTS Assess Delete Message	XOrderRef.xsd

Organizational Hierarchy Subscription API

This section describes the organizational hierarchy subscription API.

Functional Area

Organizational Hierarchy

Business Overview

If Merchandising is not the system of record for organizational hierarchy information for an implementation, then this API may be used to create, update or delete elements of the hierarchy based on an external system. The organization hierarchy subscription also assigns existing location traits to, or deletes them from the stores in the given organization hierarchy level.

The following organizational hierarchy elements can be created, modified, or deleted using this API: chain, area, region, or district. The organizational hierarchy must be created from the highest level down. Conversely, the hierarchy must be deleted from the lowest level up.

The following validation is applicable for the data sent in this API:

Message Element	Required?	Notes
Hierarchy Value	Always	Indicates the ID of the hierarchy component being added, modified, or deleted. It is validated based on the hierarchy level attribute included.
Hierarchy Description	Conditional	Indicates the description for the hierarchy component. This is required for create only. For modifications, it is optional. For delete, it is ignored.
Hierarchy Level	Always	Indicates the level of the hierarchy being created, updated, or deleted. Valid values are CH (chain), AR (area), RE (region) and DI (district).
Parent	Conditional	Identifies the parent level of the hierarchy for this component. For example, if the hierarchy level is district, this would be the ID of the region for the district.
Manager Name	Optional	Name of the manager for this hierarchy component.
Currency Code	Optional	The code that identifies the currency under which the hierarchy component operations.

Error Handling

If any errors are encountered in the validations described above or any of the message structure validations, a status of E is returned to the external system along with the appropriate error message. If the message has been successfully persisted, a success status (S), is returned to the external system indicating that the message has been successfully received and persisted to the Merchandising database.

Message XSD

Below are the filenames that correspond with each message type. Consult the RIB documentation for each message type in order to get a detailed picture of the composition of each message.

Message Type	Message Type Description	XML Schema Definition (XSD)
XOrgHrCre	External Create Organizational Hierarchy	XOrgHrDesc.xsd
XOrgHrLocTrtCre	External Create Location Trait	XOrgHrDesc.xsd
XOrgHrDel	External Delete Organizational Hierarchy	XOrgHrRef.xsd
XOrgHrLocTrtDel	External Delete Location Trait	XOrgHrRef.xsd
XOrgHrMod	External Modify Organizational Hierarchy	XOrgHrDesc.xsd

Payment Terms Subscription API

This section describes the payment terms subscription API.

Functional Area

Payment Terms

Business Overview

Payment terms are supplier-related financial arrangement information that can be subscribed to by Merchandising from a financial system. Payment terms are the terms established for paying a supplier (for example, 2.5% for 30 days) for purchase orders. After confirming the validity of the records enclosed within the message, Merchandising updates its tables with the information.

Creating Payment Terms

When a new payment term is subscribed to by Merchandising, it will first validate that all required fields are present in the message. Payment terms details should also be present when creating a new payment term, and when creating and updating a new payment term detail. After that, business level validation on the input information will be performed. The tables below summarize these two types of validations.

Table 3-91 Header Level Validation

Message Elements	Required?	Notes
terms	Always	This represents the unique ID to track this payment term in Merchandising.
terms code	Always	This is value is intended to hold the code in the financial system.
terms desc	Always	Description of the supplier terms
rank	Always	Unique number to rate invoice payment terms against purchase order terms

Table 3-92 Message Elements

Message Elements	Required?	Notes
due max amount	Always	This is the maximum amount due by a certain date
percent	Always	Percentage discount if payment is made within the time frame
terms sequence	Optional	The order in which to apply the discount percent
due days	Conditional	This is the number of days until payment is due. The following due days data combinations are valid: <ol style="list-style-type: none"> 1. Due days, due day of month and due months forward should all be provided 2. Due days, due day of month and due months forward are all NULL 3. Due days should be provided and both due day of month and due months forward are provided.

Table 3-92 (Cont.) Message Elements

Message Elements	Required?	Notes
due day of month	Conditional	<p>Day of month used to calculate due date of invoice payment line.</p> <p>The following due days data combinations are valid:</p> <ol style="list-style-type: none"> 1. Due days, due day of month and due months forward should all be provided 2. Due days, due day of month and due months forward are all NULL 3. Due days should be provided and both due day of month and due months forward are provided.
due months forward	Conditional	<p>Number of months ahead used to calculate due date of invoice payment line.</p> <p>The following due days data combinations are valid:</p> <ol style="list-style-type: none"> 1. Due days, due day of month and due months forward should all be provided 2. Due days, due day of month and due months forward are all NULL 3. Due days should be provided and both due day of month and due months forward are provided.
discount days	Conditional	<p>This is the number of days in which payment must be made in order to receive the discount.</p> <p>The following discount days data combinations are valid:</p> <ol style="list-style-type: none"> 1. Discount days, discount day of month and discount months forward should all be provided 2. Discount days, discount day of month and discount months forward are all NULL 3. Discount days should be provided and both discount day of month and discount months forward are provided. 4. Discount days is NULL and both discount day of month and discount months forward are provided.

Table 3-92 (Cont.) Message Elements

Message Elements	Required?	Notes
discount day of month	Conditional	Day of month used to calculate discount date for invoice payment line The following discount days data combinations are valid: <ol style="list-style-type: none"> Discount days, discount day of month and discount months forward should all be provided Discount days, discount day of month and discount months forward are all NULL Discount days should be provided and both discount day of month and discount months forward are provided. Discount days is NULL and both discount day of month and discount months forward are provided.
discount months forward	Conditional	Number of months ahead to calculate discount date for invoice payment line. The following discount days data combinations are valid: <ol style="list-style-type: none"> Discount days, discount day of month and discount months forward should all be provided Discount days, discount day of month and discount months forward are all NULL Discount days should be provided and both discount day of month and discount months forward are provided. Discount days is NULL and both discount day of month and discount months forward are provided.
fixed date	Optional	This is the fixed due date
enabled flag	Always	This flag should be 'Y' if the start active date is less than or equal to the current date and the end date is greater than or equal to the current date. Otherwise, the enabled flag should be 'N'.
start active date	Optional	Start active date must be less than end active date.
end active date	Optional	End date should be greater than start active date.
cutoff day	Optional	This is the last day before payment is scheduled

Payment term details sent via the detail message can also be added when a payment term already exists in Merchandising. If the terms detail being added already exists, an error is raised.

After the message passes all validations, the payment terms are inserted into the Merchandising tables.

Updating Payment Terms

Payment terms can be updated at header or detail level. When updating at the header level, the payment term details for the term being updated should already exist in Merchandising. When updating payment term details, the term header and detail must already exist.

After the message passes all validations, the payment terms in Merchandising are updated. Rank, terms, code, and terms desc are the values that can be updated at header level. At the detail level, due days, due max amount, due months forward, discount days, percent, discount day of month, discount months forward, fixed date, enabled flag, start active date, end active date and cutoff day may be updated.

Error Handling

If an error occurs in this procedure, a call will be placed to a function to build a complete error message. This message together with a status of E is returned to the external system. If the message has been successfully persisted, a success status (S), is returned to the external system indicating that the message has been successfully received and persisted to the Merchandising database.

Message XSD

Here are the filenames that correspond with each message type. Please consult the RIB documentation for each message type in order to get a detailed picture of the composition of each message.

Message Types	Message Type Description	XML Schema Definition (XSD)
PayTermCre	Payment Terms Create Message	PayTermDesc.xsd
PayTermDtlCre	Payment Terms Detail Create Message	PayTermDesc.xsd
PayTermMod	Payment Terms Modify Message	PayTermDesc.xsd
PayTermDtlMod	Payment Terms Detail Modify Message	PayTermDesc.xsd

Receiving Subscription API

This section describes the receiving subscription API.

Functional Area

Receiving

Business Overview

This API processes receipts that Merchandising receives against purchase orders, transfers, and allocations received at a store, warehouse or finisher. Purchase orders can be received at the item level only using this message, while transfers and allocations, collectively referred to as stock orders, may be received at the bill of lading

(BOL) level, where the entire shipment is received without including details, carton level, or item level.

When a transfer, PO or allocation is received at a location, Merchandising will update the appropriate tables, including the shipment, the transfer, allocation, or purchase order, stock on hand, as well as other inventory buckets (for example, in-transit). A record of the receipt is also made in the transaction level stock ledger.

Receiving Exceptions

Merchandising has the ability to automatically resolve several common exceptions that can occur during the receiving process. Here are some of the exceptions that can be supported:

Additionally, the following exceptions are automatically processed when a stock order is received through this API:

Receipt Against the Wrong BOL

In this case, the receiving location should send a carton status of Dummy (D) or Overage (O), indicating that a dummy BOL number was used. But, even if that status is not used, this exception processing can still take place. Merchandising will attempt to match the contents of the receipt to a valid BOL as follows:

- If the carton belongs to a valid BOL at the given location, Merchandising receives the carton against the intended BOL at the given location.
- If the carton belongs to a valid BOL at a related **walk-through store**, Merchandising receives the carton against the intended BOL at the intended location.
- If the carton belongs to a valid BOL at an unrelated location, Merchandising uses the **wrong store receiving** process.

Walk-through Store Receiving

If you have configured two or more stores as “walk-through” locations, through attribution at on the store table in Merchandising, then if the BOL was intended to be received at the walk-through location instead of the location on the message, Merchandising will automatically adjust the receipt and process against the correct location.

Misdirected Container

Misdirected containers, or wrong store receiving, is when one or more containers on a receipt are identified as having been originally shipped to a different location (Location A) than the location that sent the receipt message (Location B). Whether or not misdirected container receiving is supported in Merchandising is controlled by a system option called **Wrong Store Receipt Exception Handling**. If this option is unchecked (N), then the receipt at the Location B will raise an error in this API. If set to checked (Y), then the shipment at Location A will be backed out, including in-transit updates, WAC adjustments, and stock ledger postings, and re-applied to Location B, prior to processing the receipt into Location B.

In order correctly manage this processing, Merchandising must receive the original carton number on the receipt. In some cases, such as when integrating with Store Inventory and Operations Cloud Service (SIOCS), the carton ID is reassigned by the receiving location. In that case, the reference carton field in the Receipt Detail node of the message must be populated to trigger this process. Otherwise, it will be treated as an overage at the actual receiving location and the original location will not have its quantities reversed until the transfer is reconciled.

**Note:**

Wrong location receiving is supported only for item-level transfer/allocation receipts.

Unwanded Cartons

An unwanded carton is a situation where Merchandising never received notification of the original shipment, only the receipt. In this case, if receiving is done at the item level, Merchandising will process both the shipment and receipt together. If item level details are not included for the carton, an error will be raised, as Merchandising will not be able to determine the contents of the carton, having never received the initial shipment details.

Zero Receipts

This type of exception occurs when a location indicates to Merchandising that nothing was received for the item at the location by sending a receipt of zero and indicating that the carton is closed. Merchandising will reconcile the original ship-to location based on system option settings to determine where to write off the lost items.

If a zero receipt occurs for an item that is part of a misdirected container, then some slightly different processing will occur. If the zero receipt is sent **after** a misdirected container reconciliation, then no further updates will be made, as the line would have been previously reconciled. If a zero receipt occurs **before** misdirected container processing, then the misdirected container processing at the actual receiving location will be treated as an overage, as the original location would have already been reconciled.

Other Key Notes

- Externally generated warehouse-to-warehouse transfers are not supported in Merchandising, where the transfer is created in Merchandising at the physical warehouse level for both locations. For example, a warehouse-to-warehouse transfer created in WMS. This includes the receipt of such a transfer using this API.
- Wrong store receiving is not supported for franchise transactions.
- Merchandising doesn't process the top level of this message (ReceiptDesc) or the UIN level details.

Receipt

This node contains the receipt level details to be processed by Merchandising.

Message Element	Required?	Notes
DC Destination ID	Always	Indicates the location that has processed the receipt. For stores, this will be the store ID. For warehouses, this will be the physical warehouse ID.
PO Number	Always	Indicates the transaction that the receipt is for. This must be a valid allocation, transfer or PO number
Customer Order Number	Optional	Not used in Merchandising

Message Element	Required?	Notes
Fulfillment Order Number	Optional	Not used in Merchandising
Document Type	Always	Specifies whether the receipt is for an allocation (A), purchase order (P), or transfer (T, D, or V).
Reference Document Number	Optional	Contains a reference number for a document associated to the shipment (e.g., Fiscal Document ID for Brazilian based transactions).
ASN Number	Conditional	Used to relate the receipt message to the previous ASN message. This field is required for transfers and allocations.
Receipt Type	Optional	This field is used in receiving transfers or allocations to determine if the receipt is at the BOL level (BL) or SKU level (SK). If not provided, the value will be defaulted to SK. It is not used for a PO receipt.
From Location	Optional	This field contains the source location of the shipment.
From Location Type	Optional	This field contains the source location type of the shipment. Valid values are: <ul style="list-style-type: none"> • SU - supplier • ST - store • WH - warehouse
Status	Optional	Indicates the status of the ASN received. This field is used only for stock order receiving. A status of C indicates that the entire ASN is will be set to closed.

Child Nodes

- Receipt Detail
- Receipt Carton Detail

Receipt Detail

This is a required child node to the receipt level only for item level stock order receipts. For carton level receipts, this should not be populated.

Message Element	Required?	Notes
Item ID	Always	Specifies the item on the allocation, purchase order, or transfer that has been received.
Unit Quantity	Always	Contains the quantity received for the allocation, purchase order, or transfer in the standard unit of measure.
Receipt Transaction Type	Always	Specifies whether the receipt detail line item is for a Receipt (R), Transshipment (T), or Adjustment (A).
Receipt Date	Optional	Identifies the date on which the transaction was received.

Message Element	Required?	Notes
Receipt Number	Optional	An externally generated numerical identifier corresponding to the receipt of the item at the location.
Destination ID	Optional	Only used for purchase order receipt, when the purchase order has an allocation attached to it. This element specifies the location to which the allocation is being sent.
Container ID	Optional	Identifies the carton number for the item being received.
Reference Container ID	Optional	Identifies the original carton number the item was shipped under, if it was being received at the wrong destination. This is required by Merchandising to process the updates correctly, as SIOCS reassigns the container ID at the receiving location.
Distro Number	Optional	Only used for purchase order receipts, when the purchase order has an allocation attached to it. This element contains the allocation id.
Distro Document Type	Optional	Only used for purchase orders, when the purchase order has an allocation attached to it. When populated, this value should always be A to specify an allocation.
From Disposition	Optional	This value is used to determine inventory availability. Valid values are in the INV_STATUS_CODES table. The from disposition is used when the to disposition is not provided.
To Disposition	Optional	This value is used to determine if the inventory is available or unavailable, based on the code's INV_STATUS value on the INV_STATUS_CODES table.
To WIP	Optional	Not used in Merchandising
From WIP	Optional	Not used in Merchandising
To Trouble	Optional	Not used in Merchandising
From Trouble	Optional	Not used in Merchandising
User ID	Optional	Not used in Merchandising
Dummy Carton	Optional	Indicates if this carton is a dummy carton. This field is only used for transfer receipts. Valid values are yes (Y) or no (N).
Tampered Carton	Optional	Indicates if the carton was tampered. This field is only used for transfer receipts. Valid values are yes (Y) or no (N).
Unit Cost	Optional	Only used for purchase order receipts. This specifies the unit cost of the item on the order. Cost is converted to the order's currency before insert/update.
Shipped Quantity	Optional	Only used for purchase order receipts. Updates the number of items expected to be received, originally set by the ASN process.
Weight	Optional	Contains the actual weight of the item received for the shipment. This will be included for some catch weight items.

Message Element	Required?	Notes
Weight UOM	Optional	Contains the unit of measure of the received weight (e.g., pounds, kilograms) where UOM class is of type MASS. Weight and Weight UOM must both be populated, or both must be NULL.
Gross Cost	Optional	Contains the Unit cost and Expenses incurred on an item in a particular transaction.
Item Line Number	Optional	This field indicates the item line number from customer orders. This must be populated for systems with OMS_IND = Y and RESV_CO_ON_RECEIPT = Y.

Receipt Carton Detail

This is only used for stock order receiving as an alternative to the item level node.

Message Element	Required?	Notes
Carton Status	Conditional	Indicates the status of the carton being received. Valid values are: <ul style="list-style-type: none"> Actual (A) – this is used for normal carton receiving at the correct location Overage (O) – this should be used if the carton doesn't belong to the BOL/location being received. In this case, Merchandising will attempt to match the container to the correct BOL. See receiving exceptions above. Dummy BOL (D) – this should be used if the BOL used in the receipt is not valid for the carton/location. In this case, Merchandising will attempt to match the container to the correct BOL. See receiving exceptions above. Closed (C) – indicates that the carton is closed and no more receipts are expected. In this case, Merchandising will reconcile any outstanding quantity for the carton immediately.
Container ID	Conditional	Indicates the container being received. This must be populated for carton level receiving.
Reference Container ID	Optional	Not being used in Merchandising.
Destination ID	Optional	Identifies the location to which the stock order is received.
Receipt Transaction Type	Required	Specifies whether the carton receipt is for a Receipt (R), Transshipment (T), or Adjustment (A).
Receipt Date	Optional	This field contains the date when the carton is received.
Receipt Number	Optional	This field holds the externally generated number when the shipment is received at the location.
User ID	Optional	Not used by Merchandising.
To Disposition	Optional	This value is used to determine inventory availability of the received quantity. Valid values are in the INV_STATUS_CODES table.
Weight	Optional	Contains the actual weight of the item received for the carton. This is used for catch weight containers.

Message Element	Required?	Notes
Weight UOM	Optional	Contains the unit of measure of the received weight (e.g., pounds, kilograms) where UOM class is of type MASS. Weight and Weight UOM must both be populated, or both must be NULL.

Error Handling

If any errors are encountered in the validations described above or any of the message structure validations, a status of E is returned to the external system along with the appropriate error message. If the message has been successfully persisted, a success status (S), is returned to the external system indicating that the message has been successfully received and persisted to the Merchandising database.

Message XSD

Below are the filenames that correspond with each message type. Please consult the Oracle Retail Integration Guide for each message type for the details on the composition of each message.

Message Types	Message Type Description	XML Schema Definition (XSD)
receiptcre	Receipt Create Message	ReceiptDesc.xsd
receiptordcre	Receipt Create Message for Customer Orders	ReceiptDesc.xsd
receiptmod	Receipt Modify (Adjustment) Message	ReceiptDesc.xsd

RTV Subscription API

This section describes the RTV subscription API.

Functional Area

Returns to Vendor

Business Overview

Merchandising subscribes to return-to-vendor (RTV) messages when an RTV is shipped out from a warehouse or store. This shipment could be for an RTV that was initially created in Merchandising or one initiated in the store or warehouse. The RTV information is sent from a warehouse management system (WMS), such as Oracle WMS Cloud, or the store inventory solutions, such as Oracle Retail Store Inventory and Operations Cloud Service (SIOCS) when the RTV is shipped out of the location.

**Note:**

Unlike other RIB messages, both new and updates sent through this message use the RTVCre message type.

New RTVs

If the message contains a new RTV generated in the store or warehouse, then it must contain both header and detail information. RTV create messages can only be sent in Approved or Shipped status.

Updated RTVs

If this is an update to an RTV, it can be performed through this API. To update an RTV, you can send either the header information only or both header and detail information. The most common update is to ship a previously created RTV. Approved RTVs can be Shipped if the RTV is created in Merchandising or SIOCS.

It is assumed that RTVs from the warehouse are always created in Shipped status.

**Note:**

Once RTVs are shipped, they cannot be changed back to Approved. Alternatively, approved RTVs can also be set to Cancelled status, if for some reason they cannot be shipped.

RTV Header

Message Element	Required	Notes
Destination ID	Always	Contains the location shipping the RTV - either the store ID or the physical warehouse ID.
RTV ID	Always	This contains the external reference number for RTVs created outside of Merchandising.
Return Authorization Number	Optional	Contains the number that the supplier provides when the decision is made that the merchandise may be returned. This will be required depending on the configuration of the supplier site in Merchandising.
Vendor Number	Always	Contains the supplier site ID to which the merchandise is being returned. The site must be configured in Merchandising to allow returns.

Message Element	Required	Notes
Address 1	Optional	Contains the first line of the supplier's address for returns. If not provided, the address for an externally created RTV will be pulled from the supplier address information in Merchandising. This applies for all of the below address fields as well.
Address 2	Optional	Contains the second line of the supplier's address for returns.
Address 3	Optional	Contains the third line of the supplier's address for returns.
State	Optional	Contains the state of the supplier's address for returns. The state and country combination must be valid.
City	Optional	Contains the city of the supplier's address for returns.
Postal Code	Optional	Contains the postal code of the supplier's address for returns.
Country	Optional	Contains the country ID of the supplier's address for returns. The state and country combination must be valid.
Creation Time Stamp	Optional	Contains the date the vendor return was created. This defaults to current date if not specified for an externally generated RTV.
Comments	Optional	Contains any comments associated with the return.
RTV Order Number	Optional	This contains the RTV ID generated by Merchandising when the event was created. For externally generated RTVs, this should be NULL.
Status	Optional	This value is used to determine the current status of an externally generated RTV. Valid values are Approved (A) or Shipped (S). If this is Approved, Merchandising will create the RTV and set it to an In Progress status. If this is Shipped or null, it will be set to Shipped status when created or updated.

Child Node

- RTV Details
- Flex Attribute

RTV Details

Message Element	Required	Notes
Item ID	Always	Unique identifier for the item on the RTV. The item must be a transaction level inventory item or reference item.
Unit Quantity	Always	Contains the quantity of the item being returned to the supplier under this RTV number. A quantity less than zero indicates a quantity cancellation. If the RTV is already in progress, the quantity in the message must not be zero.
Container Quantity	Optional	Not used in Merchandising
From Disposition	Optional	This value is used to determine if the inventory is available or unavailable. Valid values are based on the INV_STATUS values in the INV_STATUS_CODES table.
To Disposition	Optional	Not used in Merchandising
Unit Cost	Optional	Contains the cost per unit for the SKU being returned. This field is stored in the supplier's currency. If not provided, then value will be defaulted based on the system option RTV Unit Cost Source setting - either the from location's weighted average cost, last received cost, or standard cost.
Reason	Optional	Identifies the reason for the return. Valid values are for this field are held in the Merchandising codes tables under code type RTVR. By default, the values are: Q - QC Failed U - Unavailable Inventory. W - Externally Initiated RTV If the reason is not provided for an externally generated RTV, it will be defaulted to W.
Weight	Optional	Actual weight shipped for the item on the RTV, which is used for some catch weight items. It is expected that either both weight and weight UOM have a value or neither have a value.

Message Element	Required	Notes
Weight UOM	Optional	Indicates the unit of measure represented for the provided weight (for example, pounds, kilograms) where UOM class is of type MASS. It is expected that either both weight and weight UOM have a value or neither have a value.
Gross Cost	Optional	Contains the unit cost and expenses incurred on an item in a particular transaction. This is used for Brazil processing only. For all implementations not using the Brazil Localization configuration, this should be NULL.
RTV Detail UIN	Optional	Not used in Merchandising

Flex Attributes

If flex attributes have been defined for an RTV they can be included in this node of the message.

Message Element	Required	Notes
Name	Always	The flex attribute name defined by the business
Value	Optional	The value of the flex attribute defined by the business for alphanumeric or number attributes.
Value Date	Optional	The date value of the flex attribute, if the flex attribute is defined as a date.

Error Handling

If any errors are encountered in the validations described above or any of the message structure validations, a status of E is returned to the external system along with the appropriate error message. If the message has been successfully persisted, a success status (S), is returned to the external system indicating that the message has been successfully received and persisted to the Merchandising database.

Message XSD

Here are the filenames that correspond with each message type. Please consult the RIB documentation for each message type in order to get a detailed picture of the composition of each message.

Message Types	Message Type Description	XML Schema Definition (XSD)
rtvcre	RTV Create Message	RTVDesc.xsd

Stock Count Schedule Subscription API

This section describes the stock count schedule subscription API.

Functional Area

Inventory

Business Overview

Stock count schedule messages are published to the RIB by an inventory sub-system, such as Oracle Retail Store Inventory and Operations Cloud Service (SIOCS), to communicate unit and value stock count schedules to Merchandising. Merchandising uses stock count schedule data to help synchronize the inventories of the integrated system and Merchandising. The integrated system then performs a physical inventory count and uploads the results, and Merchandising compares the discrepancies.

This API allows the external systems to create, update, and delete Unit and Value stock count requests within Merchandising. The count is assumed to be for the full location, unless department, class and subclass data are included.

Creating/Updating Stock Count Requests

When a new stock count request is created or an existing stock count request is modified, this API will validate all the required fields are present in the message. Required information for the stock count includes a description, date, type (always B), location type, and locations. Optionally the merchandise hierarchy information can also be included, but, if not included, it will be assumed the entire location will be counted. After the required field and business validations, the stock counts will be created or updated in Merchandising.

The format used when creating or modifying a stock count is shown below.

Table 3-93 Stock Count Header

Message Element	Required?	Notes
Cycle Count	Yes	This field contains the number which uniquely identifies the stock or cycle count.
Cycle Count Description	Yes	This field contains a description of the cycle or stock count which, along with the cycle count number, identifies the cycle or stock count.
Location Type	Yes	This field contains an indicator which identifies whether the cycle count will be for Stores or Warehouses. Valid values are Store (S) and Warehouse (W).
Stocktake Date	Yes	This field contains the date on which the stock or cycle count will take place. This date should be greater or equal to the current date plus the stake lockout days defined at system level.

Table 3-93 (Cont.) Stock Count Header

Message Element	Required?	Notes
Stocktake Type	Yes	This field contains a value which indicates the stock take type. Valid values are Both Unit and Dollar (B) and Unit only (U). However, the RIB interface only allows modification of Unit and Dollar stock take types.
Stock Count Schedule Products	No	Child node
Stock Count Schedule Locations	No	Child node

Table 3-94 Stock Count Schedule Products

Message Element	Required?	Notes
Department	Yes	This field contains the department number where the cycle count will occur. If the value = -1, the stock count will apply to all departments. The dept/class/subclass hierarchy must be a valid hierarchy in Merchandising.
Class	No	This field contains the class number where the cycle count will occur. The dept/class/subclass hierarchy must be a valid hierarchy in Merchandising.
Subclass	No	This field contains the subclass number where the cycle count will occur. The dept/class/subclass hierarchy must be a valid hierarchy in Merchandising.

Table 3-95 Stock Count Schedule Locations

Message Element	Required?	Notes
Location	Yes	This field contains the store or warehouse number on the cycle count. This must be a valid store or a stockholding warehouse in Merchandising.

Deleting Stock Count Requests

When an existing stock count request is deleted, this API will validate all the required fields are present in the message. After required field and business validation, the stock counts will be removed in Merchandising. This API also supports deleting a location from the count. The count and locations can only be deleted through this API if no results have been processed for the location on the count. If the last location is deleted from the count, then the count itself will be deleted.

The format used when deleting a stock count is shown below.

Table 3-96 Stock Count Header

Message Element	Required?	Notes
Cycle Count	Yes	This field contains the number which uniquely identifies the stock or cycle count being deleted.
Stock Count Schedule Locations	No	Child node

Table 3-97 Stock Count Schedule Locations

Message Element	Required?	Notes
Location	Yes	This field contains the store or warehouse being deleted on the cycle count.

Error Handling

If an error occurs in this procedure, a call will be placed to a function to build a complete error message. This message together with a status of E is returned to the external system. If the message has been successfully persisted, a success status (S), is returned to the external system indicating that the message has been successfully received and persisted to the Merchandising database.

Message XSD

Below are the filenames that correspond with each message type. Please consult the RIB documentation for each message type to get a detailed picture of the composition of each message.

Message Types	Message Type Description	XML Schema Definition (XSD)
stkcountschcre	Stock Count Schedule Create Message	StkCountSchDesc.xsd
stkcountschmod	Stock Count Schedule Modify Message	StkCountSchDesc.xsd
stkcountschdel	Stock Count Schedule Delete Message	StkCountSchRef.xsd
stkcountschtdel	Stock Count Schedule Delete Message for a Location	StkCountSchRef.xsd

Stock Order Status Subscription API

This section describes the stock order status subscription API.

Functional Area

Inventory Movement

Business Overview

A stock order is an outbound merchandise request from a warehouse or store. In Merchandising, a stock order takes the form of either a transfer or allocation. Merchandising subscribes to stock order status messages published by an external application, such as a store system (SIOCS, for example) or a warehouse management system (Oracle WMS Cloud, for example) to communicate the status of a specific stock order.

Stock Order Statuses

The following tables describe the stock order statuses for both transfers and allocation document types and what occurs in Merchandising after receiving the respective status. Statuses other than listed below are ignored by Merchandising.

Statuses for Document Types T, D, and S

Document types of T, D, and S all refer to transfers and indicate whether the transfer is initiated in Merchandising, a warehouse system, or a store system, respectively.

Stock Order Status	What Merchandising does
SI (Stock Increased)	Insert or increase transfer quantity and increase item/location transfer reserve quantity for the source location and transfer expected quantity for the destination location.
SD (Stock Decreased)	Delete or decrease the transfer quantity for the transfer/item combination. Transfer quantity for the transfer/item combination will be deleted if the transfer has been created but has not been shipped. Additionally, the item/location transfer reserved quantity for the source location and the transfer expected quantity for the destination location will be decreased.
DS (Details Selected)	Increase the selected quantity for the transfer/item combination.
DU (Details Un-selected)	Decrease the selected quantity for the transfer/item combination.
NI (WMS Line Cancellation)	Decrease selected and transfer quantity for the transfer/item by the quantity on the message and increase the cancelled quantity. Additionally, it will decrease the reserved quantity for the source location and decrease the expected quantity for the destination location by the lesser of the quantity on the message and transfer - shipped quantity. The transfer will also be added to the document close queue if transfer status is not closed. Document Close batch program will then determine if the transfer should be closed based on certain conditions. Transfers with outstanding appointments are not closed.
PP (Distributed)	Decreases the selected quantity and increases the distro quantity for the transfer/item.
PU (Un-Distribute)	Decreases the distro quantity for the transfer/item.

Stock Order Status	What Merchandising does
RS (Return to Stock)	Decreases distro quantity and transfer quantity for the transfer/item; the cancelled quantity for the transfer/item is increased. Additionally, transfer reserved is decreased for the item/source location and transfer expected is decreased for the item/destination location for the lesser of the quantity in the message and the transfer - shipped quantity if the transfer status is not closed.
EX (Expired)	Decreases transfer quantity for the transfer/item; the cancelled quantity for the transfer/item is increased. Additionally, transfer reserved is decreased for the item/source location and transfer expected is decreased for the item/destination location for the lesser of the quantity in the message and the transfer - shipped quantity if the transfer status is not closed. The transfer will also be added to the document close queue if transfer status is not closed. Document Close batch program will then determine if the transfer should be closed based on certain conditions. Transfers with outstanding appointments are not closed.
SR (Store Reassign)	Updates the distro quantity for the transfer/item. This can either increase or decrease the value, depending on whether a positive or negative value is sent.

Statuses for Document Type A

Document type A is always used for Allocations.

Stock Order Status	What Merchandising does
SI (Stock Increased)	Insert or increase allocated quantity and increase item/location transfer reserve quantity for the source location and transfer expected quantity for the destination location.
SD (Stock Decreased)	Decrease the allocated quantity for the allocation/item combination. Additionally, the item/location transfer reserved quantity for the source location and the transfer expected quantity for the destination location will be decreased.
DS (Details Selected)	Increase the selected quantity for the allocation/item combination.
DU (Details Un-Selected)	Decrease the selected quantity for the allocation/item combination.
NI (WMS Line Cancellation)	Decrease selected and allocation quantity for the allocation/item by the quantity on the message and increase the cancelled quantity. Additionally, it will decrease the reserved quantity for the source location and decrease the expected quantity for the destination location by the lesser of the quantity on the message and allocation - shipped quantity if the allocation is not closed. The allocation will also be added to the document close queue if allocation status is not closed. Document Close batch program will then determine if the allocation should be closed based on certain conditions. Allocations with outstanding appointments are not closed.
PP (Distributed)	Decreases the selected quantity and increases the distro quantity for the allocation/item.
PU (Un-Distribute)	Decreases the distro quantity for the allocation/item.

Stock Order Status	What Merchandising does
RS (Return to Stock)	Decreases distro quantity and allocation quantity for the allocation/item; the cancelled quantity for the allocation/item is increased. Additionally, transfer reserved is decreased for the item/source location and transfer expected is decreased for the item/destination location for the lesser of the quantity in the message and the allocation - shipped quantity if the allocation status is not closed.
EX (Expired)	Decreases allocation quantity for the allocation/item; the cancelled quantity for the allocation/item is increased. Additionally, transfer reserved is decreased for the item/source location and transfer expected is decreased for the item/destination location for the lesser of the quantity in the message and the allocation - shipped quantity if the allocation status is not closed. The allocation will also be added to the document close queue if allocation status is not closed. Document Close batch program will then determine if the allocation should be closed based on certain conditions. Allocations with outstanding appointments are not closed.
SR (Store Reassign)	Updates the distro quantity for the allocation/item. This can either increase or decrease the value, depending on whether a positive or negative value is sent.

For customer orders, Merchandising assumes it will receive updates from an OMS for customer order related stock orders. Therefore, to avoid duplicate processing, Merchandising will ignore No Inventory, Expired, Stock Decreased, and Stock Increased statuses received for a customer order transfer.

Stock Order Status Message Details

The table below summarizes the elements applicable for this API.

Stock Order Status Header

Message Element	Required?	Notes
DC Destination ID	Always	This field contains the location number of the stock order source location.
Location Type	Always	This field contains the type of location in the DC Destination ID field. Valid values are: <ul style="list-style-type: none"> • S - Store • W - Warehouse • E - Finisher
Store Type	Optional	This field indicates the store type of the DC Destination ID if it is a store. Valid values are: <ul style="list-style-type: none"> • C - company store • F - franchise store
Stockholding Indicator	Optional	This field indicates if the DC Destination ID is stockholding or not. Only populated if location type is 'S'. Valid values are Yes (Y) and No (N).
Distro Number	Optional	This field contains the stock order number. This is either the transfer or allocation number in Merchandising.

Message Element	Required?	Notes
Distro Document Type	Always	This field specifies whether the stock order status pertains to an allocation (A) or transfer (T) that is already existing in Merchandising. Created Stock Order (D), Customer Order (C), and Virtual Distro (V) are also valid document types but will be ignored by Merchandising.
Context Type	Optional	This field holds the functional area code to which the transfer relates to. Valid values are Promotion (PROM) and Repairing (REPAIR).
Context Value	Optional	This field holds the value relating to the context type like promotion number.
Inventory Type	Optional	This field indicates if a transfer is made from the available (A) or unavailable (U) inventory.
Customer Order Number	Optional	This field holds the master customer order number for a stock order associated with a customer order.
Fulfillment Order Number	Optional	This field holds the number related to the fulfillment details for a stock order associated with a customer order. One or more fulfillment orders could relate back to a single customer order.
System Code	Optional	Not used by Merchandising
From Location Virtual WH	Optional	This field contains the virtual warehouse number of the stock order source location.
Stock Order Status Details	Optional	Child node
Custom Flex Attributes	Optional	Child node – not used by Merchandising

Stock Order Status Detail

Message Element	Required?	Notes
Destination ID	Always	This field contains the location number of the stock order receiving location.
Location Type	Always	This field contains the type of location in the Destination ID field. Valid values are: <ul style="list-style-type: none"> • S - Store • W – Warehouse • E - Finisher
Store Type	Optional	This field indicates the store type of the Destination ID, if a store. Valid values are: <ul style="list-style-type: none"> • C - Company Store • F – Franchise Store
Stockholding Indicator	Optional	This field indicates if the Destination ID is stockholding. Only populated if location type is 'S'. Valid values are: Yes (Y) and No (N).
Item	Always	This field contains the unique identifier for the item.
Original Item	Optional	This field contains the ID of the item being replaced. Populated only when this record is for a substitute item on a customer order.

Message Element	Required?	Notes
Order Line Number	Optional	This field is used to carry the customer order line number value for customer orders. It is a derived value for non-customer orders.
Unit Quantity	Optional	This field contains the difference between the number of item units shipped versus the receiving count for the given item. This is subtracted from the document-line-item-unit-count to yield an over/under variance between what a supplier said was shipped and what was counted and received at by the store's staff.
Status	Optional	This field contains the status of the stock order. Valid values are: Accepted (SI), Rejected (SD), Distributed (PP), Un-Distributed (PU), Details Selected (DS), Details Un-selected (DU), WMS Line Cancellation (NI), Return To Stock (RS), Expired (EX), and Store Reassign (SR). Statuses other than listed are ignored by Merchandising. For stock order status explanations, please refer to the Stock Order Statuses section.
User ID	Optional	This field contains the user ID of the user who created the stock order status.
Updated Date	Optional	This field contains the date the stock order status was updated.

Error Handling

If any errors are encountered in the validations described above, or in any of the message structure validations, a status of E is returned to the external system along with the appropriate error message. If the message has been successfully persisted, a success status (S), is returned to the external system indicating that the message has been successfully received and persisted to the Merchandising database.

Message XSD

Below are the filenames that correspond with each message type. Please consult the *Oracle Retail Integration Guide* for each message type for the details on the composition of each message.

Message Types	Message Type Description	XML Schema Definition (XSD)
sostatuscre	Stock Order Status Create Message	SOStatusDesc.xsd
CustFlexAttriVo	Stock Order Status Flex Attribute Create	SOStatusDesc.xsd

Store Subscription API

This section describes the store subscription API.

Functional Area

Foundation Data

Business Overview

The Store Subscription API provides the ability to keep store data in Merchandising in sync with an external system if Merchandising is not being used as the system of record for organizational hierarchy information. The store data handled by the API includes basic store data in addition to addresses, store hours, location traits, up-charges, and walk-through stores. Walkthrough stores are used in Merchandising as part of the transfer reconciliation process and are used to indicate two or more stores that have a 'walk through' connection between them - on the sales floor and/or the backroom.

Create or Update Stores

When creating a new store in Merchandising through this API, the data is validated and placed onto the staging table `STORE_ADD`. The store creation in Merchandising reads from this table and creates the store in Merchandising in an asynchronous mode. When updating an existing store in Merchandising, the API performs the update immediately upon message receipt.

Delete Stores

The API also handles store delete messages. But, like the store creation message subscription process, stores will not actually be deleted from the system upon receipt of the message. After the data has been validated, the store is added to the `DAILY_PURGE` table for processing through a batch process.

Other Store Attributes

Location traits, walkthrough stores, department-level up-charges, store hours, address data and CFAs must be processed separately as they each have their own distinct message types for creation and deletion.

Location traits and walkthrough store attributes cannot be sent in on a store create message. The store create batch must first process the store before it can have these attributes attached to it. Up-charges, store hours, and addresses can be included for new stores.

The store subscription message also supports the ability to import custom flex attributes from an external system.

Other Notes

- Location traits must already exist prior to being added to the store.
- Stores and warehouses in Merchandising cannot have the same unique identifier.
- Location trait and walkthrough store data cannot be sent on a store create message.

Store Message Details

The table below summarizes the elements applicable for this API.

Table 3-98 Store Create or Update

Message Element	Required?	Notes
Store	Always	This field contains the unique identifier of the store.
Store Name	Always	This field contains the name of the store.
Store Type	Optional	Indicates the type of store. Valid values are: company store (C), franchise store (F).
Store Name (10)	Optional	This field contains the ten-character abbreviation of the store name.
Store Name (3)	Optional	This field contains the three-character abbreviation of the store name
Store Class	Always	This field contains code of the class of which the store is a member. Valid values are A through E.
Store Manager Name	Always	This field holds the name of the store manager.
Store Open Date	Always	This field holds the date on which the store opened.
Store Close Date	Optional	This field holds the date on which the store closed.
Acquired Date	Optional	This field holds the date on which the store was acquired.
Remodel Date	Optional	This field holds the date on which the store was remodeled.
Fax Number	Optional	This field contains the fax number for the store.
Phone Number	Optional	This field contains the phone number for the store.
Email	Optional	This field contains the email address of the store.
Total Square Feet	Optional	This field contains the total square footage of the store.
Selling Square Feet	Optional	This field contains the total square footage of the store's selling area.
Linear Distance	Optional	This field contains the total merchandise space of the store.
Stockholding Indicator	Always	This field indicates if the store can hold stock. Valid values are yes (Y) or no (N). This field cannot be modified.
Channel ID	Always	This field contains the identifier of the channel. This value must be predefined on the CHANNELS table.
Store Format	Optional	This field contains the code of the store format of the store. This value must be predefined on the STORE_FORMAT table.
Mall Name	Optional	This field contains the name of the mall in which the store is located.
District	Always	This field contains the number of the district of which the store is a member. This value must be predefined on the DISTRICT table.
Promo Zone	Optional	Not used by Merchandising.
Transfer Zone	Always	This field contains the transfer zone in which the store is located. This value must be predefined on the TSFZONE table.

Table 3-98 (Cont.) Store Create or Update

Message Element	Required?	Notes
Default WH	Optional	This field contains the default warehouse for the store. This value must be a virtual warehouse predefined on the WH table.
Stop Order Days	Optional	This field contains the number of days before the store close date that the store will stop accepting orders.
Start Order Days	Always	This field contains the number of days before the store open date that the store will begin accepting orders.
Currency Code	Always	This field contains the currency code under which the store operates. This value must be predefined on the CURRENCIES table. It cannot be modified.
Language	Optional	This field contains the code of the language used at the store. This value must be predefined on the LANG table. Either ISO code or Lang should not be null. If this field is null, ISO Code will be used for looking up the language to be used.
ISO Code	Optional	This field contains the International Organization for Standards (ISO) character code corresponding to the language used at the store. Either ISO code or Lang should not be null.
Integrated POS Indicator	Always	This field indicates whether Sales Audit should expect files from this store for processing. Valid values are Yes (Y) or No (N).
DUNS Number	Optional	This field holds the Dun and Bradstreet number to identify the store.
DUNS Location	Optional	This field holds the Dun and Bradstreet number to identify the location.
Copy Delivery Indicator	Conditional	This field indicates if the like store's delivery schedule information should be copied to the new store. This value cannot be modified. It will only be used on a store create message.
Copy Activity Indicator	Conditional	This field indicates if the like store's closing date schedule should be copied to the new store. This value cannot be modified. It will only be used on a store create message.
Price Store	Conditional	This field contains the store from which pricing information will be copied to the new store. The pricing store does not need the same currency as the new store.
Cost Location	Conditional	This field contains the location from which to copy cost information to the new store. This field should only be included on store create messages. This value must be existing store or virtual warehouse.
VAT Include Indicator	Optional	This field contains whether tax will be included in the retail prices for the store. Valid values are Yes (Y) or No (N).

Table 3-98 (Cont.) Store Create or Update

Message Element	Required?	Notes
VAT Region	Conditional	This field contains the ID of the tax region the store is associated with. This value must be predefined on the VAT_REGION table. It is required if Merchandising is configured for Simple VAT or Global Tax.
Like Store	Conditional	This field holds the store from which the new store will have item/locations copied. This value must be predefined on the STORE table. It cannot be modified and will only be used in a create message.
Copy Replenishment Indicator	Conditional	This field indicates whether replenishment information should be copied from the like store to the new store. This field cannot be modified. It will only be populated on a store create message.
Transfer Entity	Conditional	This field contains the transfer entity of which the store is a part. This value must be predefined on the TSF_ENTITY table. If the system allows intercompany transfers this field is required.
Sister Store	Optional	This field contains the store which will be used to relate historical data to the new store in Allocation. This value must be predefined on the STORE table.
Transaction Number Generated	Always	This field holds the level at which unique POS transaction numbers are generated. Valid values are Store (S) and Register (R).
County	Optional	This field contains the county in which the store is located.
Time Zone Name	Optional	This field contains the Time Zone name.
WF Customer ID	Optional	This field contains the ID associated with a franchise customer for a franchise store. This is required for a franchise store type and should be null for a company store.
Org Unit ID	Optional	This field contains the organizational unit ID value of the store.
Secondary Store Name	Conditional	This field contains the secondary name of the store.
Customer Order Location Indicator	Optional	Indicates that the store can be used to source or fulfill customer orders. Valid values are yes (Y) or no (N).
Gift Wrapping Indicator	Optional	This field indicates whether a gift wrapping needs to be done or not.
Customer Order Ship Indicator	Optional	This field indicates whether the customer order has been shipped from Warehouse or not.
Tax Id	Optional	Contains the unique tax identification number of the store.
Online Store Indicator	Optional	This field indicates how store day will be managed by ReSA. If the indicator is Y then ReSA will automatically open and close the store day.
Store Location Trait	Optional	Child node

Table 3-98 (Cont.) Store Create or Update

Message Element	Required?	Notes
Walkthrough Store	Optional	Child node
Store Address	Optional	Child node
Store Hours	Optional	Child node
Store Department Up-charge	Optional	Child node
Custom Flex Attributes	Optional	Child node

Table 3-99 Store Location Trait Create

Message Element	Required?	Notes
Location Trait	Conditional	The identifier of the location trait. Though the node is optional, this field is required if the node is included. The node cannot be populated in the store create message.

Table 3-100 Walkthrough Store Create

Message Element	Required?	Notes
Walkthrough Store	Conditional	A walk-through store of the store being modified. Though the node is optional if it is included this field is required. This node cannot be populated in a store create message.

Table 3-101 Store Address Create or Update

Message Element	Required?	Notes
City ID	Optional	This field contains the city ID or code.
State Name	Optional	This field contains the state name.
Country Name	Optional	This field contains the country name.
Address	Always	This field holds the unique address ID from the source system.
Address Type	Always	This field indicates the type for the address. Valid values are in the ADD_TYPE table. Mandatory address types for a new store are Business (01) and Postal (02).
Primary Address Indicator	Always	This field indicates whether this address is the primary address for this address type. At least one address must be designated as primary for each type.
Address 1	Always	This field contains the first line of the address.
Address 2	Optional	This field contains the second line of the address.
Address 3	Optional	This field contains the third line of the address.
City	Always	This field contains the name of the city that is associated with the address.

Table 3-101 (Cont.) Store Address Create or Update

Message Element	Required?	Notes
State	Optional	This field contains the abbreviation for the state in which the store is located.
Country ID	Always	This field contains the country code where the address exists.
Post	Optional	This field contains the postal code for the address.
Contact Name	Optional	This field contains the name of the contact for the supplier at this address.
Contact Phone	Optional	This field contains the phone number of the contact person at this address.
Contact Telex	Optional	This field contains the telex number of the store's representative contact.
Contact Fax	Optional	This field contains the fax number of the contact person at this address.
Contact Email	Optional	This field contains the email address of the store's representative contact.
Oracle Vendor Site ID	Conditional	This field contains the unique identifier of this address in the Oracle Financials, if used.
County	Optional	This field contains the county name for the location.
Jurisdiction Code	Optional	This field contains the ID associated to the tax jurisdiction of the country-state relationship.
Custom Flex Attributes	Optional	Child node

Table 3-102 Store Hours Create or Update

Message Element	Required?	Notes
Store ID	Always	This field contains the unique ID of the store.
Day Number	Always	This field indicates the day of the week for which store timing is being stored. Valid values are from 1 (Sunday) through 7 (Saturday).
Store Open Time	Optional	This field contains the open time for the store. Open time should be in 12-hour format (HH:MI AM or HH:MI).
Store Close Time	Optional	This field contains the close time for the store. Close time should be in 12-hour format (HH:MI AM or HH:MI).

Table 3-103 Store Department Up-charge Header Create or Update

Message Element	Required?	Notes
Hierarchy Level	Always	The hierarchy level for which the up charges are being added or updated. Valid values are Division (DI), Group (GR), Department (DE), and All Departments (AD).

Table 3-103 (Cont.) Store Department Up-charge Header Create or Update

Message Element	Required?	Notes
Hierarchy	Optional	The value of the hierarchy based on the level. If the level is division, group, or department, then this should contain a valid ID of that type. If All Departments is the hierarchy level, then this should be left blank.
From Location	Optional	Contains the source location for which the up changes for the hierarchy will apply. Valid values are a country, area, region, store, virtual warehouse, or physical warehouse ID, depending on the from location type included. If the from location type sent is AS or AW, this should be left null.
To Location	Optional	Contains the destination location for which the up changes for the hierarchy will apply. Valid values are a country, area, region, store, virtual warehouse, or physical warehouse ID, depending on the from location type included. If the from location type sent is AS or AW, this should be left null.
From Location Type	Always	Indicates the type of location included as the From Location. Valid values are Country (C), Area (A), Region (R), Store (S), Virtual Warehouse (W), Physical Warehouse (PW), All Stores (AS), or All Warehouses (AW).
To Location Type	Always	Contains the type of location included as the To Location. Valid values are Country (C), Area (A), Region (R), Store (S), Virtual Warehouse (W), Physical Warehouse (PW), All Stores (AS), or All Warehouses (AW).
Store Department Up-charge Details	Optional	Child node

Table 3-104 Store Department Up-charge Details Create or Update

Message Element	Required?	Notes
Component ID	Always	This field contains the unique identifier of the up-charge component.
Component Rate	Always	This field contains the rate to be charged based on the transfer/allocation cost for items in the hierarchy and between from and to locations defined in the up-charge header.
Per Count	Optional	If the component was defined with a calculation basis of Specific, then this field is used to determine how the rate is applied to the items on a transfer or allocation. If the component is defined as Value, this should be blank.
Per Count UOM	Optional	Indicates the unit of measure in which the Per Count is specified.

Table 3-104 (Cont.) Store Department Up-charge Details Create or Update

Message Element	Required?	Notes
Up-charge Group	Always	This field contains the group to which the component ID belongs. Valid values can be found on the codes table with a code type of UCHG.
Component Currency	Always	This field contains the currency of the Up Charge component.
Effective Date	Optional	The date from which the new values are effective in the system.
Item Default Indicator	Optional	By default, up-charges defined will be defaulted for new items created in the hierarchy designated at the header level. But this indicator allows you to add the new or updated upcharge to an existing item in the hierarchy. Valid values are Y or N.
Transfer Allocation Default Indicator	Optional	Indicates if component information should be applied to any unshipped transfers or allocations with items and locations that qualify. Valid values are Y or N.

Table 3-105 Store Delete

Message Element	Required?	Notes
Store	Always	The store number being deleted, or for which a location trait or walk through store is being disassociated.
Store Location Trait	Optional	Child node
Walkthrough Store	Optional	Child node
Store Address	Optional	Child node
Store Hours	Optional	Child node
Store Department Up-charge	Optional	Child node

Table 3-106 Store Location Trait Delete

Message Element	Required?	Notes
Location Trait	Conditional	The identifier of the location trait. Though the node is optional, this field is required if the node is included. The node cannot be populated in the store create message.

Table 3-107 Walkthrough Store Delete

Message Element	Required?	Notes
Walkthrough Store	Conditional	A walk-through store of the store being modified. Though the node is optional if it is included this field is required. This node cannot be populated in a store create message.

Table 3-108 Store Address Delete

Message Element	Required?	Notes
Address	Always	The unique identifier of the address being deleted.

Table 3-109 Store Hours Delete

Message Element	Required?	Notes
Store ID	Always	This field contains the unique ID of the store.
Day Number	Always	This field indicates the day of the week for which store timing is being stored. Valid values are from 1 (Sunday) though 7 (Saturday).

Table 3-110 Store Department Up-charge Header Delete

Message Element	Required?	Notes
Hierarchy Level	Always	The hierarchy level for which the up charges are being deleted. Valid values are Division (DI), Group (GR), Department (DE), All Departments (AD).
Hierarchy	Optional	The value of the hierarchy.
From Location	Optional	Contains the source location from which goods will be transferred. This column can contain Country/Area/Region IDs when From Location Type is 'C', 'A', or 'R'. It will be a store, virtual warehouse, or physical warehouse when From Location Type is 'S', 'W' or 'PW'. Otherwise, it should be left blank when From Location Type is either 'AS' or 'AW'.
To Location	Optional	Contains the destination location to which goods will be transferred. This column can contain Country/Area/Region IDs when To Location Type is 'C', 'A', or 'R'. It will be a store, virtual warehouse or physical warehouse when To Location Type is 'S', 'W' or 'PW'. Otherwise, it should be left blank when To Location Type is either 'AS' or 'AW'.
From Location Type	Always	Contains the type of source location from which goods will be transferred. Valid values are Country (C), Area (A), Region (R), Store (S), Virtual Warehouse (W), Physical Warehouse (PW), All Stores (AS), or All Warehouses (AW).
To Location Type	Always	Contains the type of destination location to which goods will be transferred. Valid values are Country (C), Area (A), Region (R), Store (S), Virtual Warehouse (W), Physical Warehouse (PW), All Stores (AS), or All Warehouses (AW).
Store Department Up-charge Details	Optional	Child node

Table 3-111 Store Department Up-charge Details Delete

Message Element	Required?	Notes
Component ID	Optional	This field contains the unique identifier of the up-charge component.
Item Default Indicator	Optional	Indicates if component rate information is deleted or not for existing items under the department.
Transfer Allocation Default Indicator	Optional	Indicates if component rate information is deleted or not for existing transfers and allocations under the department.

Flex Attributes

If you have defined any custom flex attributes (CFAs) for stores or store addresses, then they can be integrated as part of this API. The node of the integration that supports this will accept the name of the attribute as it is defined in the group set level view and the value for the attribute.

Table 3-112 Custom Flex Attributes

Message Element	Required?	Notes
Name	Always	Indicates the name of the column defined in the group set view for flex attributes defined for the store.
Value	Always	Indicates the value of the attribute for the store if the attribute is a character or number.
Value Date	Always	Indicates the value of the attribute for the store if the attribute is a date.

Error Handling

If any errors are encountered in the validations described above or any of the message structure validations, a status of E is returned to the external system along with the appropriate error message. If the message has been successfully persisted, a success status (S), is returned to the external system indicating that the message has been successfully received and persisted to the Merchandising database.

Message XSD

Below are the filenames that correspond with each message type. Please consult the Oracle Retail Integration Guide for each message type for the details on the composition of each message.

Message Types	Message Type Description	XML Schema Definition (XSD)
XStoreCre	External Store Create	XStoreDesc.xsd
XStoreLocTrtCre	External Store Location Trait Create	XStoreDesc.xsd
XStoreWTCre	External Walk-Through Store Create	XStoreDesc.xsd

Message Types	Message Type Description	XML Schema Definition (XSD)
XStoreDeptChrgCre	External Department Up-Charge Create	XStoreDesc.xsd
XStoreAddrCre	External Store Address Create	XStoreDesc.xsd
XStoreHrCre	External Store Hours Create	XStoreDesc.xsd
CustFlexAttriVO	External Store Flex Attribute Create	XStoreDesc.xsd
XStoreMod	External Store Modification	XStoreDesc.xsd
XStoreDeptChrgMod	External Department Up-Charge Modify	XStoreDesc.xsd
XStoreAddrMod	External Store Address Modify	XStoreDesc.xsd
XStoreHrMod	External Store Hours Modify	XStoreDesc.xsd
CustFlexAttriVO	External Store Flex Attribute Modify	XStoreDesc.xsd
XStoreDel	External Store Delete	XStoreRef.xsd
XStoreLocTrtDel	External Store Location Trait Delete	XStoreRef.xsd
XStoreWTDel	External Walk-Through Store Delete	XStoreRef.xsd
XStoreDeptChrgDel	External Department Up-Charge Delete	XStoreRef.xsd
XStoreAddrDel	External Store Address Create	XStoreRef.xsd
XStoreHrDel	External Store Hours Delete	XStoreRef.xsd

Transfer Subscription API

This section describes the transfer subscription API.

Functional Area

Transfers

Business Overview

This API subscribes to transfers from external systems to create, update or delete transfers in Merchandising. Within Oracle Retail solutions, this API is also leveraged by Advanced Inventory Planning (AIP) to create standalone transfers generated out of its replenishment processing. AIP does not use this API to update or delete previously created transfers.

Creating Transfers

When a new transfer is created, this API will first validate that all required fields are present in the message. Certain of the fields are required regardless of transfer type and system configuration, while others are dependent on other Merchandising configurations. Additionally, when creating a new transfer at least one detail line must also be included in the message. After that, business level validation on the input information will be performed. The tables below summarize these two types of validation.

Table 3-113 Header Level Validation

Message Element	Required?	Notes
Transfer Number	Yes	Must be a unique transfer number not used by any existing transfers in Merchandising.
From Location Type	Yes	Must be either a store (S) or a warehouse (W).
From Location	Yes	See below.
To Location Type	Yes	Must be either a store (S), warehouse (W), or external finisher (E). For more on transfers with finishing, see below.
To Location	Yes	See below.
Delivery Date	Conditional	When AIP is part of your implementation, this is required for all transfer types, except RAC, EG, and SIM transfers. If included in the message, this must be today or a future date.
Department	Conditional	A system option determines whether or not the department is required for transfers. If the system option is set to require a department, then this must be included in the message. If the system option is set to not require the department, then the department must be null in this message unless the transfer type is SIM, AIP, or EG.
Routing Code	Conditional	If the freight code is Expedite (E), then this must have a value. Otherwise, it must be null. Valid values are 1, 2, or 3. The descriptions for these three options are held in the Codes table under code TRRC and can be configured as needed for your business.
Freight Code	No	If this is included in the message, it must have a value of normal (N), hold (H), or expedite (E). If no value is provided, it will default to normal.
Transfer Type	No	The following types of transfers can be created in this API: <ul style="list-style-type: none"> • Administrative (AD) • AIP Generated (AIP) • Book (BT) • Confirmation (CF) • Externally Generated (EG) • Intercompany (IC) • Manual Requisition (MR) • Reallocation (RAC) • Return to Vendor (RV) • SIM Generated (SIM) If the transfer type is not specified for the new transfer, then it will be defaulted to either Manual Requisition or Intercompany, depending on the legal entities of the locations on the transfer. See below for more details on transfer types.
Status	No	Transfers can be created in Input (I) or Approved (A) status in this API. See below for more on transfer status validation.

Table 3-113 (Cont.) Header Level Validation

Message Element	Required?	Notes
Create ID	No	If not passed into the message, then a value will be defaulted for auditing purposes.
Comments	No	Can support up to 2000 characters of text.
Context Type	No	Valid values for this field are found in the Codes table under code type CNTX.
Context Value	No	This may be used to provide additional information about the context of the transfer. For example, if the context type is promotion, this may indicate the promotion number or a description.
Custom Flex Attribute	No	Child node

Table 3-114 Detail Level Validation

Message Element	Required?	Notes
Item	Yes	An item must be a transaction-level, inventoried, and approved in order to be included on a transfer. If the transfer is from a store, then it cannot be a pack item, unless the transfer is of type AIP, SIM, or EG. Also, packs can also only be transferred from a warehouse if they have a receive as type of Pack, so that inventory exists at that level. If a department has been included in the transfer message at the header level, then all items must belong to that department.
Transfer Quantity	Yes	If the item has a standard unit of measure in the quantity class, then the quantity for the transfer must be an integer.
Supplier Pack Size	No	If included, this must be greater than zero. If not included, this will default to the primary supplier's pack size for the item at the from location, if an orderable item. If not orderable, this will default to 1.
Inventory Status	No	All items on a transfer must either be from available status or from unavailable status. A single transfer cannot mix available and unavailable statuses. Available inventory transfers should use a -1 or a null value in this field in the message. Unavailable inventory transfers should include a valid status from the Inventory Status Types configured in your environment.
Unit Cost	No	This is not used by Merchandising.

Table 3-114 (Cont.) Detail Level Validation

Message Element	Required?	Notes
Adjustment Type	Conditional	<p>This field, along with the adjustment value, is used to calculate the transfer price for intercompany transfers. It will be ignored for all other transfers. If the adjustment value is provided, then the type must also be specified. The valid values for this field are:</p> <ul style="list-style-type: none"> • IA - Increase by Amount • IP - Increase by Percent • DA - Decrease by Amount • DP - Decrease by Percent • S - Set Price <p>IA and IP can only be used if you have your system options set to allow the transfer price to exceed weighted average cost.</p>
Adjustment Value	Conditional	If the adjustment type is provided, then the value must also be specified. This must always be a positive amount.

Location Validation

The from and to locations passed into the message must be valid stores or warehouses in Merchandising; but they cannot be the same. If both locations are stores, then they must both exist in the same transfer zone. Additionally, if the to location is a store, then it must be open. This is determined based on whether there is a close date defined for the store and the stop order days.

If either location is a warehouse, then it can be either a physical warehouse or a virtual warehouse, depending on transfer type. A physical warehouse is only allowed as the from location type for an EG type of transfer. Additionally, only Book type transfers are allowed between two warehouses in the same physical warehouse.

If either the from or to location is a franchise store, then the other location cannot be a finisher. If the franchise store is a non-stockholding location, then the other location on the transfer must be a warehouse.

Validation is also done at the item level based on the locations on the transfer. Each item on the transfer must be in active, inactive, or discontinued status at the from location. It also must have been ranged to the from location in Merchandising, when that location is a warehouse. However, if the from location is a store, there is an exception where the transfer can still be created even though it is not yet ranged, which also bypasses inventory validation. This is to support a specific function in Oracle Retail Store Inventory Management (SIM). See the section on SIM Generated transfers below for more details.

If the item is not already ranged to the to location, then ranging will occur when the transfer is created, regardless of status. The ranging that occurs will flag the item/ location as unintentionally ranged for all transfer types except AIP.

If the to location is an external finisher, see the section below on transfers with finishing.

Inventory Validation

Another part of the validation that is applicable for all transfers created is that inventory is available for transfer if the status passed through the integration is approve (A), with a few exceptions. First, EG type transfers do not have inventory validated as it is assumed that this type of transfer is generated in the store or warehouse and the inventory availability check has been done in that solution as part of the shipping of the inventory. Additionally, if the system option titled Validate External Warehouse Availability is set to No (unchecked), then warehouse inventory will not be validated for any transfers initiated in this API regardless of type. Store inventory availability is never validated by this API because of support for the process where the item does not need to be ranged to the shipping store.

Status Validation

Transfers can be created in a status of Input (I) or Approved (A) using this API. Transfers in input status are not subject to inventory validation, but all other validations are applicable. Book type transfers can only be created in Input status using this API, as there isn't really a concept of an "approved" book transfer - as soon as it is approved it is executed. Additionally, transfers of type Reallocation (RAC) and Return to Vendor (RV) can also only be created in Input status. Conversely, transfers of type AIP, SIM, and EG must always be created in Approved status. If any validation fails when processing the new transfer that results in it not being able to be approved, the transfer will be created but will remain in input status. The exception to this is for transfers of type AIP, SIM, and EG, as they must always be created in approved status. If they are not able to be approved, the transfer is not created or updated.

Transfer Type Specific Validation

Most of the validation defined above is relevant regardless of transfer type, except where noted. However, there are also some other validations done as part of this API's processing that are specific to a type of transfer.

Administration (AD)

- See Manual Requisition

AIP Generated (AIP)

This type of transfer is expected only to be sent from AIP as an output of the replenishment process. As such, Merchandising assumes certain validations have been done by AIP in advance of receiving the transfer and slightly different validation is enforced. The following special validations apply for this transfer type using this API:

- Must be created in Approved status
- Can only be to stockholding locations
- Supports transferring packs from stores
- Allows the department number to be passed even when the system option is N
- Item/location ranging to the to location will result in the Ranged flag being set to Yes as it is assumed this an intentional ranging.
- Can be an intercompany transfer

Book (BT)

Book transfers processed through this API can be created for two virtual warehouses in the same physical warehouse only. This is usually used for inventory rebalancing between virtual locations. The following special validations apply for this transfer type using this API:

- Can only be created in Input status
- Can only be created for virtual warehouses in the same physical warehouse
- Warehouses must be in the same legal entity

Confirmation (CF)

- See Manual Requisition

Externally Generated (EG)

Externally Generated transfers are assumed to be created in the store or warehouse. Further, it is assumed that once they get to Merchandising, the transfer is already in process at that location. As such, there are certain validations that are managed differently for this transfer type in this API:

- Must be created in Approved status
- Supports transferring packs from stores
- Allows the department number to be passed even when the system option is N
- Can be an intercompany transfer
- Uses the physical warehouse number, not a virtual warehouse number, if warehouses are involved

Intercompany (IC)

An intercompany transfer is a type of business to business transaction that sells product from one legal entity and purchases it into another. Legal entities in Merchandising are determined based on the setting of the Intercompany Basis system option, which indicates whether the transfer entity or the set of books of a location should be used. This transfer type is used when either it is explicitly passed into the API or if the transfer type is NULL in the inbound message and the locations are in different legal entities. Other transfer types may also be intercompany, as well, but the below rules apply for those flagged as intercompany type explicitly:

- The legal entity of the from and to locations must be different.
- If an adjustment type or value is passed into the message, that will be used to calculate the "selling" price between entities. Otherwise, the from location's weighted average cost is used.

Manual Requisition (MR)

This is the most basic type of transfer in Merchandising, so it is used as a default transfer type when either it is explicitly passed into the API or if the transfer type is NULL in the inbound message and the locations are in the same legal entity. The behavior for this transfer type is the same as that for AD and CF types of transfers -

those could be used as different reasons for a transfer. For this transfer type the following validation rules are enforced:

- Locations must be in the same legal entity

Reallocation (RAC)

A reallocation transfer is assumed to be used to pull back inventory from stores or warehouses to a single warehouse for re-allocation to other stores or other warehouses. This is the type of transfer that is created when a mass-return transfer is created, for example. Because it has unique rules tied to it related to MRTs, some additional validations are followed:

- Can only be created in Input status in this API
- Locations must be in the same legal entity

Return to Vendor (RV)

A return to vendor type of transfer is similar to a reallocation type, in that it is assumed to be pulling inventory back to a warehouse from stores or other warehouses, but in this case, for the purpose of returning the merchandise to the supplier. This is the type of transfer that is created when a mass-return transfer is created, for example. Because it has some unique rules tied to it related to MRTs, some additional validations are followed:

- Can only be created in Input status in this API
- Locations must be in the same legal entity

SIM Generated (SIM)

SIM generated transfers are created only by the store orders process in SIM. This functionality is not available in SIOCS. Because of this, they have special rules applied, including the ability to create the transfer even though no item/store relationship exists for the originating location in Merchandising. The rules that apply for this type of transfer include:

- Must be created in Approved status
- Supports transferring packs from stores
- Allows the department number to be passed even when the Merchandising system option is No
- Can be an intercompany transfer

All Transfer Types

For all of the above transfer types, if all validation described above passes, then the transfer will be created. If the transfer is created in Approved status, then in addition to the transfer itself, other details may also be created based on the items and locations involved.

- Inventory will be updated to reflect the reserved quantity at the from location and expected quantity at the to location.
- Upcharges will be applied, if configured, for transfers that do not include a physical warehouse location. For transfers with a physical warehouse, the records for upcharges are added when the transfer is shipped.
- An associated franchise order or return will be created if the transfer involves a franchise location.

Transfers with Finishing

Transfers with finishing are sometimes referred to as a two-legged transfer, as they generate two transfers in Merchandising. One from the originating store or warehouse to the finisher and one from the finisher back to a store or warehouse. This API supports the creation of a transfer with finishing only through an external finisher, a type of partner, and back to the originating location. Transfers to an internal finisher are not supported via this integration. To do this, when sending the transfer details in the message, you will indicate the external finisher as the "to" location. Then when the transfer is created, it will automatically generate the second leg.

When creating transfers in this way, it does not generate any work order activities to send to the finisher with the transfer - these will either need to be added manually in the Merchandising screens, or sent separately to the finisher.

Updating Transfers

For updates, the transfer number included in the message must already exist in Merchandising. Changes can be sent for header level updates or detail level updates. If the changes are at the header level, then the all the required header level information need to be included in the update, similar to that described above for creating a new transfer. However, the transfer details should not be included in a header level update. Fields that can be updated at the header level using this API include:

- Delivery Date - must always be a date today or later.
- Routing Code - if the freight code is updated to expedite (E), then this must also have a value. If freight code is updated to something other than expedite, then this should be null.
- Freight Code
- Status - to move from Input or Submitted to Approved only. Transfers cannot be moved back to Input status using this API.
- Comments
- Context Type
- Context Value

If the update is at the detail level - to add or update a line item - only the transfer number is required in the header record, the other details are ignored. If not included, then the message will be rejected. Adding a new item to the transfer will use similar validation to that described above when creating the transfer.

If modifying an existing transfer line item, the full transfer quantity should be sent with the update, not the difference from the original quantity. This will be compared to the previous transfer quantity to determine how to update the transfer. For example, if the transfer is in approved or submitted status, a reduction in quantity would update the cancelled quantity on the transfer. It will also be validated to ensure that the quantity change doesn't result in the total transfer quantity being lower than what has already been shipped or what is expected to be picked based on updates to the selected or distro quantities on the transfer. For increases in transfer quantity, if the transfer is in submitted or approved status, then inventory will be validated based on the changed quantity (depending on system option settings) to validate that the additional units are available. The inventory status for the item cannot be modified.

Deleting Transfers

If you are deleting a line item on the transfer or deleting the whole transfer, then the API will validate that the transfer number is valid and that the transfer or transfer line was not already shipped or received, at least partially, or is not in process at the shipping warehouse or store. If you are deleting the whole transfer, then no details should be included in the message. If you are deleting a line on the transfer, then validation will be done to ensure that the item exists on the transfer.

Transfers are not actually deleted via this API, rather they are updated to a deleted status and a secondary process does the actual removal. Transfers can be deleted in any status, other than those already in a closed or deleted status, using this API. If the transfer involved an external finisher, then both legs on the transfer will be marked for delete. Deleting the last line on the transfer will also result in the transfer being flagged for delete.

If the transfer is in a status other than input, moving it to a deleted status or deleting a line will also update inventory to release the reserved inventory at the from location and decrease expected quantity at the to location. As well, if the transfer involves any franchise stores, then any franchise order or return created with the transfer will also be cancelled.

Table 3-115 Transfer Header Delete

Message Element	Required?	Notes
Transfer Number	Always	Must be an existing transfer in Merchandising.
Transfer Detail	Optional	Child node.

Table 3-116 Transfer Detail Delete

Message Element	Required?	Notes
Item	Always	Must be an item on an existing transfer in Merchandising.

Publishing Updates

Because these transfers that can be created, updated, or deleted using this API are managed in an external system, there are some cases where it is not published back out by Merchandising after it is processed to avoid the source system from receiving unneeded updates. This applies for transfers of type EG only. All other transfers will be published back to the RIB if approved or previously approved, such that the store and warehouse solutions responsible for executing the transfers are notified.

Flex Attributes

If you have defined any custom flex attributes (CFAS) for transfers, then they can be integrated as part of this API. The node of the integration that supports this will accept the name of the attribute as it is defined in the group set level view and the value for the attribute. Flex attributes can only be added or updated to a transfer, they cannot be deleted. Additionally, for transfers with finishing, flex attributes can only be added to the first leg of the transfer.

Table 3-117 Flex Attributes

Message Element	Required?	Notes
Name	Yes	Holds the attribute name.
Value	No	Holds the value of the attribute for number and character type attributes
Value Date	No	Holds the date for date type attributes.

Error Handling

If any errors are encountered in the validations described above or any of the message structure validations, a status of E is returned to the external system along with the appropriate error message. If the message has been successfully persisted, a success status (S), is returned to the external system indicating that the message has been successfully received and persisted to the Merchandising database.

Message XSD

Below are the filenames that correspond with each message type. Please consult the Oracle Retail Integration Guide for each message type for the details on the composition of each message.

Message Types	Message Type Description	XML Schema Definition (XSD)
Xtsfcre	Transfer Create	XTsfDesc.xsd
Xtsfdtlcre	Transfer Detail Create	XTsfDesc.xsd
CustFlexAttriVO	Transfer Flex Attribute Create	XTsfDesc.xsd
Xtsfmod	Transfer Modify	XTsfDesc.xsd
Xtsfdtlmod	Transfer Detail Modify	XTsfDesc.xsd
Xtsfdtlmod	Transfer Flex Attribute Modify	XTsfDesc.xsd
Xtsfdel	Transfer Delete	XTsfDesc.xsd
Xtsfdtldel	Transfer Detail Delete	XTsfRef.xsd

Vendor Subscription API

This section describes the vendor subscription API.

Functional Area

Foundation Data

Business Overview

Merchandising subscribes to vendor information that is published from an external financial application; however, this API is not used by Oracle Retail Financial Integration (RFI). Vendor can refer to either a partner or a supplier. Any partners loaded in this API must be created in Merchandising manually using the same ID as is

sent in this API, to facilitate integration back to financials. Supplier information subscribed to by this API includes supplier addresses, org unit, and any flex attributes defined at the supplier or address levels.

Vendor Create and Update

When new suppliers or supplier sites are sent from an external system, they must contain the required header details, as well as address and org unit information. If the supplier information sent is for an ID that does not already exist, then this API will create a new supplier or site. If the ID already exists, then this API will update the existing supplier or site. For supplier sites, all addresses are expected to be the same, regardless of type. If a mandatory address type is not included in the message, it will be defaulted based on the provided address. One or more org units can be associated with a supplier site using this API, if applicable.

Table 3-118 Vendor Create and Update

Message Element	Required?	Notes
Vendor Header	Always	Child node
Vendor Address	Optional	Child node
Vendor Org Unit	Optional	Child node

Table 3-119 Vendor Header Create and Update

Message Element	Required?	Notes
Supplier	Always	Unique identifying number for a supplier site within the system.
Supplier Name	Always	This field contains the supplier site name.
Supplier Name Secondary	Optional	This type can hold secondary name for the supplier site with a max length of 240 characters.
Contact Name	Always	This field contains the name of the supplier representative contact for this site.
Contact Phone	Always	This field contains a telephone number for the supplier's representative contact.
Contact Fax	Optional	This field contains a fax number for the supplier's representative contact.
Contact Pager	Optional	This field contains a pager number for the supplier's representative contact.
Supplier Status	Always	This field contains the status of the supplier site. Valid values are: <ul style="list-style-type: none"> • A - Active • I - Inactive
QC Ind	Always	This field determines whether orders from this supplier will require quality control.
QC Percentage	Optional	This field contains the percentage of items per receipt that will be marked for quality checking.
QC Frequency	Optional	This field contains the frequency for which items per receipt will be marked for quality checking.

Table 3-119 (Cont.) Vendor Header Create and Update

Message Element	Required?	Notes
VC Ind	Always	This field determines whether orders from this supplier will require vendor control.
VC Percentage	Optional	This field contains percentage of items per receipt that will be marked for vendor checking.
VC Frequency	Optional	This field contains the frequency for which items per receipt that will be marked for vendor checking.
Currency Code	Always	This field contains code identifying the currency the supplier site uses for business transactions.
Language	Optional	This field contains the supplier's preferred language.
Terms	Always	This field contains an indicator identifying the purchase terms that will default when an order is created for the supplier site. These terms specify when payment is due and if any discounts exist for early payment.
Freight Terms	Always	This field contains code indicating what freight terms will default when an order is created for the supplier site.
Return Allow Indicator	Always	This field indicates whether the supplier site will accept returns. Valid values are Yes (Y) or No (N).
Return Authorization Required	Always	This field indicates if returns must be accompanied by an authorization number when sent back to the vendor.
Returns Minimum Amount	Optional	This field contains a value if the supplier site requires a minimum merchandise value to be returned to accept the return. Returns of less than this amount will not be processed by the system. This field is stored in the supplier's currency.
Return Courier	Optional	This field contains the name of the courier that should be used for returns to the supplier site.
Handling Percentage	Optional	This field contains the default percent to be multiplied by the return's total cost to determine the handling cost for the return.
EDI PO Ind	Always	This field indicates whether purchase orders will be sent to the supplier via EDI.
EDI PO Change	Always	This field indicates whether purchase order changes will be sent to the supplier via EDI.
EDI PO Confirmation	Always	This field indicates whether acknowledgements of purchase orders will be sent from the supplier via EDI.
EDI ASN	Always	This field indicates whether the supplier will send Advance Shipment Notifications electronically.
EDI Sales Report Frequency	Optional	This field contains the EDI sales report frequency for the supplier. Valid values are weekly (W) or daily (D).
EDI Supplier Availability Indicator	Always	This field indicates whether the supplier will send availability via EDI.
EDI Contract Indicator	Always	This field indicates whether the supplier site supports contract ordering sent via EDI.

Table 3-119 (Cont.) Vendor Header Create and Update

Message Element	Required?	Notes
EDI Invoice Indicator	Always	This field indicates whether invoices, debit memos, and credit note requests will be sent to/from the supplier via EDI.
Cost Change Percentage Variance	Optional	This field contains a percent that determines whether a cost change can be auto approve via induction. If the cost change falls within these boundaries, it will be approved when uploaded.
Cost Change Amount Variance	Optional	This field contains an amount (in supplier currency) that determines whether a cost change can be auto approve via induction. If the cost change falls within these boundaries, it will be approved when uploaded.
Replenishment Approval Indicator	Always	This field indicates whether contract orders created via replenishment should be created in Approved status.
Ship Method	Optional	This field contains the default method used to ship the items on the purchase order from the supplier site. Valid values are held in code type SHPM.
Payment Method	Optional	This field indicates the default method for how purchase orders for this site will be paid. Valid options are: <ul style="list-style-type: none"> • LC - Letter of Credit • WT - Wire Transfer • OA - Open Account
Contact Telex	Optional	This field contains a telex number for the supplier's representative contact.
Contact Email	Optional	This field contains an email address for the supplier's representative contact.
Settlement Code	Always	This field indicates which payment process method is used for the supplier. Valid values are N/A (N) or Evaluated Receipts Settlement (E).
Pre-Mark Indicator	Always	This field indicates whether the supplier site supports pre-marking containers for cross dock orders.
Auto Approved Invoice Indicator	Always	This field indicates whether the supplier's invoices can be automatically approved for payment.
Debit Memo Code	Optional	This field indicates when a debit memo will be sent to the supplier site to resolve a discrepancy. Valid values are: <ul style="list-style-type: none"> • Y - if debit memos are always to be sent • L - if debit memos are used only if a credit note is not sent by the invoice due date • N - if debit memos are never sent
Freight Charge Indicator	Always	This field indicates whether a supplier site can charge freight costs.
Auto Approve Debit Memo Indicator	Always	This field indicates whether debit memos sent to the supplier site can be automatically approved on creation.

Table 3-119 (Cont.) Vendor Header Create and Update

Message Element	Required?	Notes
Inventory Management Level	Optional	This field indicates the level for managing supplier inventory information. Valid values are supplier (S), supplier/location (L), supplier/department (D), or supplier/department/location (A). If no value is provided, then if the department level orders system option is set to Yes, then this is defaulted to supplier/department, otherwise it is defaulted to supplier.
Backorder Indicator	Always	This field indicates if backorders or partial shipments will be accepted.
VAT Region	Optional	This field contains the unique identifying number for the VAT region applicable for this site.
Prepay Invoice Indicator	Always	This field indicates whether all invoices for the supplier can be pre-paid.
Service Performed Required Indicator	Always	This field indicates if the supplier's services must be confirmed as performed before paying an invoice from that supplier site.
Invoice Payment Location	Optional	This field indicates where invoices from this supplier site are paid - at the store (S) or centrally through corporate accounting (C).
Invoice Received Location	Optional	This field indicates where invoices from this supplier site are received - at the store (S) or centrally through corporate accounting (C).
Invoice At	Optional	This field indicates if the supplier site invoice lists items at gross cost (G) or net cost (N).
Delivery Policy	Always	This field contains the default delivery policy of the supplier site. Valid values come from the DLVY code type.
Comments	Optional	This field contains any miscellaneous comments associated with the supplier.
Default Item Lead Time	Optional	This field holds the default lead time for the supplier site. The lead time is the time the supplier needs between receiving an order and having the order ready to ship. This value will be defaulted to item/supplier relationships.
DUNS Number	Optional	The Dun and Bradstreet number of the supplier.
DUNS Location	Optional	The Dun and Bradstreet number of the location of the supplier.
Bracket Costing Indicator	Always	This field will determine if the supplier site supports bracket costing pricing structures.
VMI Order Status	Optional	This field determines the status in which any inbound POs from this supplier will be created. A NULL value indicates that the supplier is not a VMI supplier.
End Active Date	Optional	Not used by Merchandising.
DSD Supplier Indicator	Always	This field specifies whether the vendor supports DSD ordering, where the supplier replenishes the store directly, creating the PO and receipt at the same time.

Table 3-119 (Cont.) Vendor Header Create and Update

Message Element	Required?	Notes
Supplier Quantity Level	Always	This field indicates the supplier site order quantity level. Valid values are cases (CA) or eaches (EA).
Supplier Parent	Optional	This is the supplier number for the supplier sites.
Store Delivery Discrepancy	Optional	Not used by Merchandising
Final Destination Indicator	Always	This field indicates whether the supplier site can ship to final destination or not. Valid values are Yes (Y) or No (N).
External Reference Indicator	Optional	This column holds the ID for the supplier used in the external financial system.
Deal Upload Status	No	This field indicates the status in which the deal will be created when uploaded. The valid values are: <ul style="list-style-type: none"> • W - Worksheet • S - Submitted • A - Approved
Tax ID	No	This field contains the unique tax identification number of the supplier site.
Custom Flex Attributes	Optional	Child Node

Table 3-120 Vendor Address Create or Update

Message Element	Required?	Notes
Module	Always	This field indicates the data type that the address is attached to. In this case, it will always be 'SUPP'.
Key Value 1	Always	This field holds the ID the address is attached to. In this case, it will be the supplier number.
Key Value 2	Optional	This is not used.
Sequence Number	Always	Number indicating the sequence that addresses within the same type were entered.
Address Type	Always	This field contains the address type. Valid address types are: <ul style="list-style-type: none"> • Business (01) • Postal (02) • Returns (03) • Order (04) • Invoice (05) • Remittance (06) <p>If any address types have been flagged as mandatory and are not included when the supplier is being created, then if an ordering address has been included, the missing mandatory addresses will be defaulted to that address. If not, then the remittance address will be used. If neither an order nor remittance address is included, then the first address sent is used.</p>
Primary Address Indicator	Always	This field indicates whether the address is the primary address for the address type.
Address 1	Always	This field contains the first line of the address.

Table 3-120 (Cont.) Vendor Address Create or Update

Message Element	Required?	Notes
Address 2	Optional	This field contains the second line of the address.
Address 3	Optional	This field contains the third line of the address.
City	Always	This field contains the name of the city that is associated with the address.
State	Optional	This field contains the state abbreviation for the address.
Country ID	Always	This field contains the country where the address exists.
Jurisdiction Code	Optional	This field contains the ID associated to the tax jurisdiction of the country-state relationship.
Post	Optional	This field contains the zip code for the address.
Contact Name	Optional	This field contains the name of the contact for the supplier at this address.
Contact Phone	Optional	This field contains the phone number of the contact person at this address.
Contact Telex	Optional	This field contains the telex number of the contact person at this address.
Contact Fax	Optional	This field contains the fax number of the contact person at this address.
Contact Email	Optional	This field contains the email address of the supplier site's contact person.
Custom Flex Attributes	Optional	Child Node

Table 3-121 Vendor Org Unit Create

Message Element	Required?	Notes
Org Unit ID	Always	This field contains org unit ID added or updated for the supplier site.
Primary Pay Site Indicator	Always	This field contains the primary pay site indicator.

Flex Attributes

If custom flex attributes (CFAS) have been defined for suppliers or addresses, then they can be integrated as part of this API. The node of the integration that supports this will accept the name of the attribute as it is defined in the group set level view and the value for the attribute. Flex attributes can only be added to or updated for a supplier and supplier address but cannot be deleted.

Message Element	Required?	Notes
Name	Always	Holds the name of the attribute.
Value	Optional	Holds the value of the attribute for non-date attributes.

Message Element	Required?	Notes
Value Date	Optional	Holds the value of the attribute for date attributes.

Error Handling

If any errors are encountered in the validations described above or any of the message structure validations, a status of E is returned to the external system along with the appropriate error message. If the message has been successfully persisted, a success status (S), is returned to the external system indicating that the message has been successfully received and persisted to the Merchandising database.

Message XSD

Below are the filenames that correspond with each message type. Please consult the Oracle Retail Integration Guide for each message type for the details on the composition of each message.

Message Types	Message Type Description	XML Schema Definition (XSD)
vendorcre	Vendor Create	VendorDesc.xsd

Work Order Status Subscription API

This section describes the work order status subscription API.

Functional Area

Transfers

Business Overview

For transfers with finishing, Merchandising subscribes to work order status messages sent from internal finishers indicating that the work order activities are complete. This message is used for internal finishers located in the same physical warehouse as the final destination for the transfer, as there is no physical shipment of goods. Other finishing scenarios exist in which the finisher is not a virtual warehouse that shares a physical warehouse with the transfer's final receiving location. In these instances, Work Order Status messages are not necessary, and Merchandising will disregard Work Order Status messages sent in these scenarios.

Work order status messages contain the items for which the activities have been completed, along with the quantity that was completed. All items on transfers that pass through an internal finisher must have at least one work order activity associated with them. When work order status messages are received for a particular item/quantity, it is assumed that all activities on the work order associated with the item/quantity have been completed. If work order activities involve item transformation or repacking, the work order status messages are always created in terms of the resultant item or pack.

On processing the work order status update, a book transfer is executed between the internal finisher (which is held as a virtual warehouse) and the final receiving location (also a virtual

warehouse). If the internal finisher belongs to the sending location's transfer entity, intercompany out and intercompany in transactions are recorded. Quantities on hand, reserved quantities, and weighted average costs are adjusted to accurately reflect the status of the stock.

It is possible to receive multiple Work Order Status messages for a particular item/transfer. Work order completion of partial quantities addresses the following scenarios:

1. Work order activities could not be performed for the entire quantity of a particular item at one time.
2. A given quantity of the particular item was damaged while work order activities were performed.

Work Order Example

Assume that a quantity of 20 of item 100 (White XL T-shirt) are sent to an internal finisher at the receiving physical warehouse, where they will be dyed black, thereby transforming them into item 101 (Black XL T-shirt). If all finishing activities were successfully completed in this example, Merchandising could expect to receive a Work Order Status message containing item 101 with a quantity of 20.

Work Order Status Creation

While consuming the Work Order Status message, Merchandising validates that the finisher and the transfer's final receiving location are in the same physical warehouse. If not, processing is halted. If the message contains an item, work order complete processing will be called for that item. Otherwise, said processing will be called for all items on the transfer. If the entire transfer is processed, the child transfer (that is, the second leg) will be set to Shipped status. Note that work orders are always associated with the second leg of multi-leg transfers. Whether processing is performed at the item or transfer level, transfer closing logic will be used to determine if the entire multi-leg transfer can be closed.

Table 3-122 Work Order Status

Message Element	Required?	Notes
Work Order ID	Yes	This field contains the work order number under which the finishing activities were performed.
Distro Number	Yes	This field contains the transfer number containing the work order. This is the 2nd leg transfer number.
Distro Doc Type	Yes	This should always be transfer (T).
Distro Parent Number	No	This is the first leg transfer number. If this field must be the same as the tsf_parent_no of the distro number in the tsfhead table in Merchandising.
Distro Parent Type	No	Not used
Item	Conditional	This contains the item on which the work order has been completed. If an item transformation occurred, this should be the resultant item. If the completed quantity field has a value, then this field is required.
Warehouse	Yes	Not used, but required as input

Table 3-122 (Cont.) Work Order Status

Message Element	Required?	Notes
Location Type	Yes	This field contains the destination location type where the finished goods are sent. Not used, but required as input
Location	Yes	This field contains the destination location where the finished goods are sent. This is the final location of the 2-legged transfer.
Sequence Number	Yes	Not used, but required as input
Work In Process Code	Yes	Not used, but required as input
Instructions	Yes	Not used, but required as input
Complete Date	Yes	This field contains the date when the work order was completed.
Completed Quantity	Conditional	This contains the number of items that resulted from the work order activity. If the item field has a value, then this field is required.
Completed Indicator	Yes	Not used
Work Order Status Inventory Adjustment	No	Child node Not used

Table 3-123 Work Order Status Inventory Adjustment

Message Element	Required?	Notes
From Disposition	No	Not used
To Disposition	No	Not used
Unit Quantity	No	Not used

Error Handling

If any errors are encountered in the validations described above or any of the message structure validations, a status of E is returned to the external system along with the appropriate error message. If the message has been successfully persisted, a success status (S), is returned to the external system indicating that the message has been successfully received and persisted to the Merchandising database.

Message XSD

Here are the filenames that correspond with each message type. Please consult RIB documentation for each message type in order to get a detailed picture of the composition of each message.

Message Types	Message Type Description	XML Schema Definition (XSD)
wostatuscre	Work Order Status Create Message	WOStatusDesc.xsd

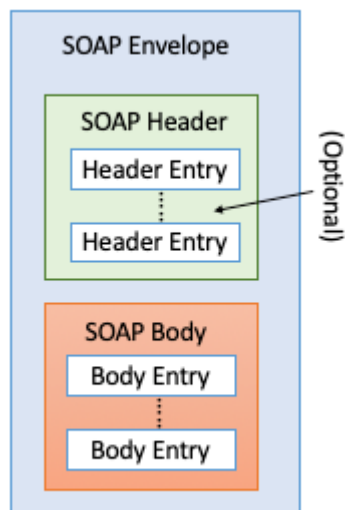
4

SOAP Web Services

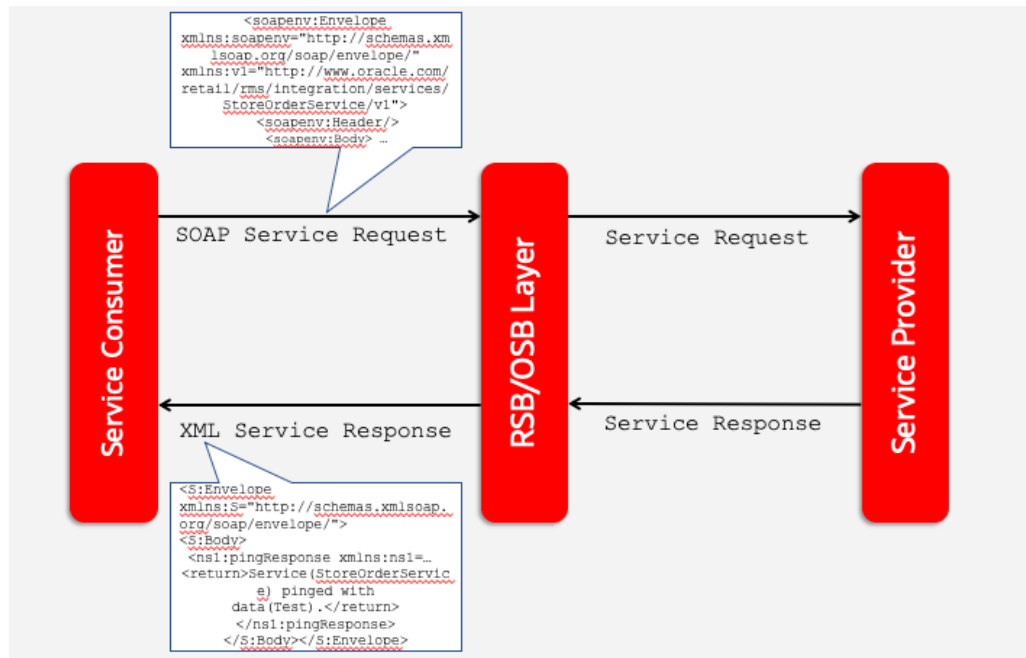
This chapter gives an overview about the SOAP Web service implementation used in Merchandising.

The Simple Object Access Protocol (SOAP) is a general-purpose messaging protocol that is the de facto standard for web services messaging and interaction through the Oracle Retail Integration Cloud Service (RICS) Retail Service Backbone (RSB), which provides monitoring for the SOAP services used by Merchandising. The basic unit of interaction between a SOAP client and a SOAP-enabled service is a message. A SOAP message is basically an XML document that consists of two parts:

1. An optional header providing information on authentication, encoding of data, or how a recipient of a SOAP message should process the message.
2. The body that contains the message. These messages are defined using the WSDL specification.



An envelope can enclose any number of optional headers. The following diagram shows the high-level architecture of SOAP web service implementation with respect to Merchandising:



Using SOAP Services During Batch Window

The services should not be used during the restricted batch window.

Common Characteristics of Merchandising SOAP Services

A Retail Application will package its SOAP services as part of the application's Enterprise Archive (EAR) file. Installation of the SOAP web services is therefore done by default as part of the application install. Refer to *Oracle Retail Service Backbone Implementation Guide* for more details.

Security

Services are secured using a standard policy-based security model supported by WebLogic and OSB.

For more details, refer to the *Oracle Retail Service Backbone Security Guide*.

Standard Success Response

Example response payload in case of service success is depicted below:

```
<S:Envelope xmlns:S="http://schemas.xmlsoap.org/soap/envelope/">
<S:Body>
<ns1:createDetailXAllocDescResponse xmlns:ns1="http://www.oracle.com/
retail/rms/integration/services/AllocationService/v1"
xmlns:ns3="http://www.oracle.com/retail/integration/base/bo/
InvocationSuccess/v1"
xmlns:ns2="http://www.oracle.com/retail/integration/base/bo/XAllocDesc/v1"
xmlns:ns4="http://www.oracle.com/retail/integration/base/bo/XAllocColRef/v1"
xmlns:ns5="http://www.oracle.com/retail/integration/base/bo/XAllocRef/v1">
```



```

        <ns3:InvocationSuccess>
          <ns3:success_message>createDetailXAllocDesc service call was successful.</
ns3:success_message>
        </ns3:InvocationSuccess>
      </ns1:createDetailXAllocDescResponse>
    </S:Body>
  </S:Envelope>

```

Standard Error Response

Example response payload in case of service error is depicted below:

```

<S:Envelope xmlns:S="http://schemas.xmlsoap.org/soap/envelope/">
  <S:Body>
    <ns0:Fault xmlns:ns0="http://schemas.xmlsoap.org/soap/envelope/"
xmlns:ns1="http://www.w3.org/2003/05/soap-envelope">
      <faultcode>ns0:Server</faultcode>
      <faultstring>Invalid Item. 1003500087</faultstring>
      <detail>
        <ns0:IllegalStateWSFaultException xmlns:ns0="http://www.oracle.com/retail/
integration/services/exception/v1">
          <ns0:shortErrorMessage>Invalid Item. 1003500087</ns0:shortErrorMessage>
          <ns0:errorDescription>
            com.oracle.retail.integration.services.exception...
          </ns0:errorDescription>
          <ns0:BusinessProblemDetail>
            <ns0:problemDescription>Invalid Item. 1003500087</
ns0:problemDescription>
          </ns0:BusinessProblemDetail>
        </ns0:IllegalStateWSFaultException>
      </detail>
    </ns0:Fault>
  </S:Body>
</S:Envelope>

```

URL Path

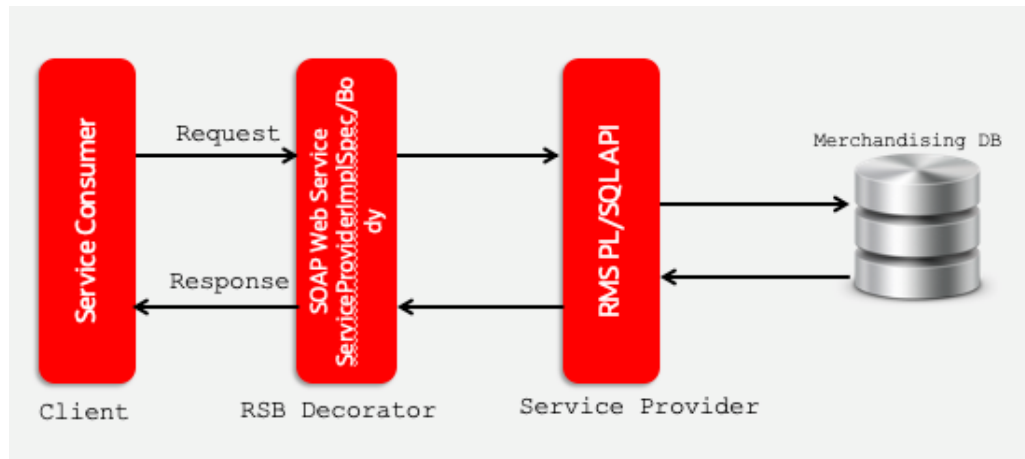
To access the Merchandising SOAP web services WSDL file:

```
https://<hostname>/<end-point>
```

The end point information is in the descriptions of each of the provider services later in this chapter.

Web Service APIs Process Flow

The following diagram shows the Web Service API process flow for Merchandising as a Service Provider:



Provider Services

This section gives an overview about the SOAP Web service provider implementation API designs used in the Merchandising environment and various functional attributes used in the APIs.

Note:

The following service provider implementation API designs are intended only to give a high-level overview of the APIs available. The implementation of these services, along with the associated Web Service Definition Language (WSDL), may be used to get a full understanding of the data requirements, validation rules, persistence rules, and return values associated with the service.

To provide visibility to the background processing that's occurring, services write to the JOB_AUDIT_LOGS table in the database. Reports can be built based on this to provide visibility to what is happening in the background. Additionally, to assist users and developers in troubleshooting any error that may arise, the payload that was processed is also stored in the JOB_AUDIT_PAYLOAD table in the database.

Once the nightly batch run has started, web service execution will be halted, and users will receive a warning message that the nightly batch run has commenced.

Consumer Services

This section lists the details on the SOAP services where Merchandising is the consumer of the service.

Provider Services

This section gives an overview about the SOAP Web service provider implementation API designs used in the Merchandising environment and various functional attributes used in the APIs.

 **Note:**

The following service provider implementation API designs are intended only to give a high level overview of the APIs available.

The implementation of these services, along with the associated Web Service Definition Language (WSDL), may be used to get a full understanding of the data requirements, validation rules, persistence rules, and return values associated with the service.

Allocation Service

Functional Area

Allocation

RSB Proxy WSDL

`/rms-Allocation-AppServiceDecorator/ProxyService/AllocationAppServiceProxy?wsdl`

Merchandising Service WSDL

`/AllocationBean/AllocationService?WSDL`

Overview

This service allows an external application to create, update, and delete allocations within Merchandising based on warehouse inventory or to cross-dock a purchase order.

This service uses the same logic as is supported in the Allocation Subscription RIB API. For information about this functionality, see [Allocation Subscription API](#) in the "RIB Subscription Designs" chapter of this document.

Operation XSD

Here are the filenames that correspond with each operation. Please consult the RSB documentation for each in order to get a detailed picture of the composition.

Operation Name	Input Object Type	Output Object Type	XML Definition
create	XAllocDesc	InvocationSuccess	XAllocDesc.xsd
createDetail	XAllocDesc	InvocationSuccess	XAllocDesc.xsd
modifyHeader	XAllocDesc	InvocationSuccess	XAllocDesc.xsd
modifyDetail	XAllocDesc	InvocationSuccess	XAllocDesc.xsd
delete	XAllocColRef	InvocationSuccess	XAllocRef.xsd XAllocColRef.xsd
deleteDetail	XAllocColRef	InvocationSuccess	XAllocRef.xsd XAllocColRef.xsd

Average Cost Service

Functional Area

Finance

RSB Proxy WSDL

`/rms-AverageCost-AppServiceDecorator/ProxyService/AverageCostAppServiceProxy?wsdl`

Merchandising Service WSDL

`/AverageCostBean/AverageCostService?WSDL`

Overview

This service supports updating weighted average cost from an external system for one or more item/locations combinations. It also creates a tran data record posting with tran code 70 for the difference in cost, based on owned inventory at the location at the time the cost change is applied.

The web service will be called with the following details:

- Item
- Location
- Location Type
- New average cost (must be greater than 0)

Operation XSD

Here are the filenames that correspond with each operation. Please consult the RSB documentation for each in order to get a detailed picture of the composition.

Operation Name	Input Object Type	Output Object Type	XML Definition
modifyAvgCost	ItLocAgCstColDesc	InvocationSuccess	ItLocAgCstDesc.xsd ItLocAgCstColDesc.xsd

Cost Change Service

Functional Area

Cost Change

RSB Proxy WSDL

`/rms-CostChange-AppServiceDecorator/ProxyService/CostChangeAppServiceProxy?wsdl`

Merchandising Service WSDL

/CostChangeBean/CostChangeService?WSDL

Overview

This service is exposed to allow an external application to create cost changes in Merchandising. It takes a collection of cost changes and will return success and failure through the service response object.

This service uses the same logic as is supported in the Cost Change Subscription RIB API. For information about this functionality, see [Cost Change Subscription](#) in the "RIB Subscription Designs" section of this document.

Operation XSD

Here are the filenames that correspond with each operation. Please consult the RSB documentation for each in order to get a detailed picture of the composition.

Operation Name	Input Object Type	Output Object Type	XML Definition
create	XCostChgColDesc	InvocationSuccess	XCostChgDesc.xsd XCostChgColDesc.xsd

Customer Credit Check Service

Functional Area

Franchise

RSB Proxy WSDL

/rms-CustomerCreditCheck-AppServiceDecorator/ProxyService/
CustomerCreditCheckAppServiceProxy?wsdl

Merchandising Service WSDL

/CustomerCreditCheckBean/CustomerCreditCheckService?WSDL

Overview

This API provides a way for an external source, usually a financials system, to update the credit status for a franchise customer in Merchandising. This status is used when determining whether a franchisee order can be approved. Valid values are Y (credit is good) and N (credit issues). For each collection of customer and customer group passed into the API, the credit flag will be updated with the value indicated in the service call.

Merchandising returns failure status as part of the response object in the web service call if credit flag is not updated due to validation errors.

Operation XSD

Here are the filenames that correspond with each operation. Please consult the RSB documentation for each in order to get a detailed picture of the composition.

Operation Name	Input Object Type	Output Object Type	XML Definition
updateCustCredit	CustCreditChkCol	InvocationSuccess	CustCreditChkDesc.xsd CustCreditChkCol.xsd

Customer Order Fulfillment Service

Functional Area

Customer Order Fulfillment

RSB Proxy WSDL

```
/rms-FulfillOrder-AppServiceDecorator/ProxyService/FulfillOrderAppServiceProxy?wsdl
```

Merchandising Service WSDL

```
/FulfillOrderBean/FulfillOrderService?WSDL
```

Overview

This service is used to process Customer Order Fulfillment requests from an order management system (OMS). Merchandising supports two integration methods for processing Customer Order Fulfillment messages from OMS - either through RIB or web service. At implementation time, you should decide on either one or the other integration method, but not both. The same core logic is used to validate and persist customer orders.

In a web service implementation, the web service is used to create or cancel a customer order in Merchandising. This service

- Accepts a collection of fulfillment orders as input. If one order fails, the entire service call fails and no orders will be created.
- Returns Failure status as part of the response object in the web service call if customer orders are not created due to validation errors.
- Returns Success status and a confirmation message as part of the response object of type
 - X if customer orders are not created due to lack of inventory
 - P if customer orders are partially created due to insufficient inventory
 - C if customer orders are completely created, when sufficient inventory is available

In a web service implementation, confirmation messages will be sent in a collection as part of the response object.

This is the web service version of the same logic as is supported in the RIB version of the API. See [Customer Order Fulfillment Subscription API](#) in the "RIB Subscription Designs" chapter for more information.

Operation XSD

Here are the filenames that correspond with each operation. Please consult the RSB documentation for each in order to get a detailed picture of the composition.

Operation Name	Input Object Type	Output Object Type	XML Definition
create	FulfilOrdColDesc	FulfilOrdCfmCol	FulfilOrdDesc.xsd FulfilOrdColDesc.xsd
cancel	FulfilOrdColRef	InvocationSuccess	FulfilOrdRef.xsd FulfilOrdColRef.xsd

Customer Order Item Substitution Service

Functional Area

Customer Orders

RSB Proxy WSDL

```
/rms-CustOrdSubstitute-AppServiceDecorator/ProxyService/  
CustOrdSubstituteAppServiceProxy?wsdl
```

Merchandising Service WSDL

```
/CustOrdSubstituteBean/CustOrdSubstituteService?WSDL
```

Business Overview

When a store is picking inventory to fulfill a customer order, if the inventory of the item ordered does not meet quality standards or is unavailable, and the order indicates that substitutions are allowed for that item, the store may choose to fulfill the order with a substitute item. If that occurs, SIM has the ability to substitute items on the customer order with another predefined substitute item. In such cases, SIM notifies OMS via the Stock Order Status message that an alternative item has been pushed into the order.

Based on the notification from SIM, OMS updates the customer order and notifies Merchandising with the same details received from SIM using this API. Merchandising will then update the inventory and customer order details - removing the reservation for the original item and adding a reservation for the new item. Merchandising will also update the cancelled quantity for the original item on the order and add the details for the substituted item, with a cross reference to the original item.

Assumptions

- Substitution logic holds good only for the customer orders fulfilled from stores.

- Catchweight, Transformable, Consignment, Concession and Deposit container items are not supported for customer order item substitution.
- The quantities are always in Standard UOM.

Operation XSD

Here are the filenames that correspond with each operation. Please consult the RSB documentation for each in order to get a detailed picture of the composition.

Operation Name	Input Object Type	Output Object Type	XML Definition
create	CustOrdSubColDesc	InvocationSuccess	CustOrdSubDesc.xsd CustOrdSubColDesc.xsd

Diff Management Service

Functional Area

Foundation

RSB Proxy WSDL

/rms-DiffManagement-AppServiceDecorator/ProxyService/
DiffManagementAppServiceProxy?wsdl

Merchandising Service WSDL

/DiffManagementBean/DiffManagementService?WSDL

Overview

This service supports the following functions

- Creating new differentiator (diff) IDs
- Updating existing diff IDs
- Deleting existing diff IDs
- Creating diff group header and details
- Updating existing diff group headers and details
- Deleting existing diff group headers and details

This API uses the same logic that is used for managing diffs through the Diff Subscription RIB API. See [Diff Group Subscription API](#) and [Differentiator Subscription API](#) in the "RIB Subscription Designs" chapter of this document for more details.

Operation XSD

Here are the filenames that correspond with each operation. Please consult the RSB documentation for each in order to get a detailed picture of the composition.

Operation Name	Input Object Type	Output Object Type	XML Definition
createDiffId	XDiffIDColDesc	InvocationSuccess	XDiffIDDesc.xsd XDiffIDColDesc.xsd
modifyDiffId	XDiffIDColDesc	InvocationSuccess	XDiffIDDesc.xsd XDiffIDColDesc.xsd
deleteDiffId	XDiffIDColRef	InvocationSuccess	XDiffIDRef.xsd XDiffIDColRef.xsd
createDiffGrp createDiffGrpDtl	XDiffGrpColDesc	InvocationSuccess	XDiffGrpDesc.xsd XDiffGrpColDesc.xsd
modifyDiffGrp modifyDiffGrpDtl	XDiffGrpColDesc	InvocationSuccess	XDiffGrpDesc.xsd XDiffGrpColDesc.xsd
deleteDiffGrp deleteDiffGrpDtl	XDiffGrpColRef	InvocationSuccess	XDiffGrpRef.xsd XDiffGrpColRef.xsd

Inventory Back Order Service

Functional Area

Inventory

RSB Proxy WSDL

```
/rms-InventoryBackOrder-AppServiceDecorator/ProxyService/  
InventoryBackOrderAppServiceProxy?wsdl
```

Merchandising Service WSDL

```
/InventoryBackOrderBean/InventoryBackOrderService?WSDL
```

Overview

Retailers selling through ecommerce channels often take customer orders even if inventory is not available with the expectation of future inventory being available to fill the order. If an order is captured against future inventory by the Order Management System (OMS), then a backorder message is sent to Merchandising through this service.

This web service will update the backorder quantity in Merchandising - increasing when the backorder is taken and decreasing when the backorder is released for fulfillment or cancellation.

Assumptions

- Backorders can be taken against both stores and warehouses. OMS will determine which location will be back ordered.
- An item does not need to have an open purchase order in order to increase backorder quantity.

- Catchweight, Transformable, Consignment, Concession and Deposit container items are not supported for backorder requests.

Operation XSD

Here are the filenames that correspond with each operation. Please consult the RSB documentation for each in order to get a detailed picture of the composition.

Operation Name	Input Object Type	Output Object Type	XML Definition
create	InvBackOrdColDesc	InvocationSuccess	InvBackOrdDesc.xsd InvBackOrdColDesc.xsd

Inventory Lookup Service

Functional Area

Inventory

RSB Proxy WSDL

```
/rms-InventoryDetail-AppServiceDecorator/ProxyService/  
InventoryDetailAppServiceProxy?wsdl
```

Merchandising Service WSDL

```
/InventoryDetailBean/InventoryDetailService?WSDL
```

Overview

This real-time inventory availability lookup facility can be used by external systems, such as an on-line order capture system (OOC) or order management system (OMS), to retrieve item/location available inventory based on Merchandising's current view of inventory. Merchandising will provide this information for any warehouse or store which is valid for customer order sourcing/fulfillment.

Available inventory is calculated as

Stock on Hand - (transfer reserved + customer reserved + RTV + non-sellable)

Any failures (validation errors) encountered during the processing are passed back into the response object. If there are no failures, success status is returned.

Assumptions

- Catchweight, transformable, consignment, concession and deposit container items are not supported in this API.
- This inventory detail lookup is only for customer orderable inventory - sellable items at customer order locations. If a physical warehouse and channel are passed into the API, then only the inventory for the customer orderable virtual warehouses for that physical warehouse/channel are returned.

- If the inventory lookup is for a pack item at store, the pack inventory is estimated based on the maximum number of complete packs which can be created by using the available inventory of its components.
- Merchandising does not use the Search Area information in this service. It will only lookup inventory for the specific locations included in the input object.

Operation XSD

Here are the filenames that correspond with each operation. Please consult the RSB documentation for each in order to get a detailed picture of the composition.

Operation Name	Input Object Type	Output Object Type	XML Definition
lookup	InvAvailCriVo	InvAvailColDesc	InvAvailCriVo.xsd InvAvailDesc.xsd InvAvailColDesc.xsd

Item Management Service

Functional Area

Item

RSB Proxy WSDL

/rms-ItemManagement-AppServiceDecorator/ProxyService/ItemManagementAppServiceProxy?wsdl

Merchandising Service WSDL

/ItemManagementBean/ItemManagementService?WSDL

Overview

The Item Management service allows an external systems to request pre-issued item numbers, as well as create, modify and delete various aspects of an item.

Request Item Numbers

The Item Number Reservation web service allows external systems such as Oracle Retail Assortment Planning (AP) to reserve item numbers in Merchandising. This web service contains the following details:

Column Name	Notes
Item Number Type	Required. Indicates the type of items numbers being requested. Valid options are: <ul style="list-style-type: none"> • ITEM - which is type Oracle Retail Item Number • UPC-A - which is type UCC12 • UPC-AS - which is type UCC12 with Supplement • EAN13 - which is type EAN/UCC-13

Column Name	Notes
Quantity	Indicates the number of item numbers being requested. Required.
Days Until Expiry	Indicates how long the calling solution wants Merchandising to retain the reservation. After this many days, the reservation will be released allowing these numbers to be used for other purposes. This is required and must be a value greater than 0.

The requested item numbers are sent back to the calling solution as a response. This operation is only available as part of the web service.

Create/Manage Items

The operations supported in this service for creating and managing items are as follows:

- Creating and modifying items
- Creating, modifying, and removing item suppliers
- Creating, modifying, and removing item supplier sourcing country
- Creating, modifying, and removing item supplier country dimensions
- Creating, modifying, and removing item UDA combinations
- Creating and removing item reclassifications

This service uses the same logic to manage these operations as is used in the Item Subscription RIB API. For information on this functionality, see [Item Subscription API](#) in the "RIB Subscription Designs" chapter of this document.

Operation XSD

Here are the filenames that correspond with each operation. Please consult the RSB documentation for each in order to get a detailed picture of the composition.

Operation Name	Input Object Type	Output Object Type	XML Definition
reservItemNumber	ItemNumCriVo	ItemNumColDesc	ItemNumCriVo.xsd ItemNumDesc.xsd ItemNumColDesc.xsd
createItem	XItemColDesc	InvocationSuccess	XItemDesc.xsd XItemColDesc.xsd
createSupplier	XItemColDesc	InvocationSuccess	XItemDesc.xsd XItemColDesc.xsd
createSupplierCountry	XItemColDesc	InvocationSuccess	XItemDesc.xsd XItemColDesc.xsd
createSupplierCountryDim	XItemColDesc	InvocationSuccess	XItemDesc.xsd XItemColDesc.xsd
createUDA	XItemColDesc	InvocationSuccess	XItemDesc.xsd XItemColDesc.xsd
modifyItem	XItemColDesc	InvocationSuccess	XItemDesc.xsd XItemColDesc.xsd

Operation Name	Input Object Type	Output Object Type	XML Definition
modifySupplier	XItemColDesc	InvocationSuccess	XItemDesc.xsd XItemColDesc.xsd
modifySupplierCountry	XItemColDesc	InvocationSuccess	XItemDesc.xsd XItemColDesc.xsd
modifySupplierCountryDim	XItemColDesc	InvocationSuccess	XItemDesc.xsd XItemColDesc.xsd
modifyUDA	XItemColDesc	InvocationSuccess	XItemDesc.xsd XItemColDesc.xsd
deleteSupplier	XItemColRef	InvocationSuccess	XItemRef.xsd XItemColRef.xsd
deleteSupplierCountry	XItemColRef	InvocationSuccess	XItemRef.xsd XItemColRef.xsd
deleteSupplierCountryDim	XItemColRef	InvocationSuccess	XItemRef.xsd XItemColRef.xsd
deleteUDA	XItemColRef	InvocationSuccess	XItemRef.xsd XItemColRef.xsd
createItemReclass	XItemRclsDesc	InvocationSuccess	XItemRclsDesc.xsd
createItemReclassDetail	XItemRclsDesc	InvocationSuccess	XItemRclsDesc.xsd
deleteItemReclass	XItemRclsRef	InvocationSuccess	XItemRclsRef.xsd
deleteItemReclassDetail	XItemRclsRef	InvocationSuccess	XItemRclsRef.xsd

Pay Term Service

Functional Area

Financial Integration

RSB Proxy WSDL

`/rms-PayTerm-AppServiceDecorator/ProxyService/PayTermAppServiceProxy?wsdl`

Merchandising Service WSDL

`/PayTermBean/PayTermService?WSDL`

Overview

The Pay Term Service is used by Oracle Retail Financial Integration (RFI) for integration of payment terms with PeopleSoft Financials, and can also be used by an external financial systems to send new and updated payment terms information to Merchandising. The operations supported in this service are:

- Create: Create payment terms and details

- Create Detail: Add details to an existing payment term
- Update Header: Modify existing payment term header information
- "Update or Update Details: Modify existing details for a payment term

The operations supported by this service involve an external system sending Merchandising details to create or update payment terms. In the response back, the terms keys are returned on success. The create and update options for this service use the same logic as is supported in the Payment Terms Subscription RIB API.

For information about this functionality, see [Payment Terms Subscription API](#) in the "RIB Subscription Designs" chapter of this document.

Operation XSD

Here are the filenames that correspond with each operation. Please consult the RSB documentation for each in order to get a detailed picture of the composition.

Operation Name	Input Object Type	Output Object Type	XML Definition
create	PayTermDesc	PayTermRef	PayTermDesc.xsdPayTermRef.xsd
createDetail	PayTermDesc	PayTermRef	PayTermDesc.xsdPayTermRef.xsd
update	PayTermDesc	PayTermDesc	PayTermDesc.xsd
updateHeader	PayTermDesc	PayTermDesc	PayTermDesc.xsd
updateDetail	PayTermDesc	PayTermDesc	PayTermDesc.xsd

Pricing Cost Service

Functional Area

Foundation Data

RSB Proxy WSDL

`/rms-PricingCost-AppServiceDecorator/ProxyService/PricingCostAppServiceProxy?wsdl`

Merchandising Service WSDL

`/PricingCostBean/PricingCostService?WSDL`

Overview

This web service is used to expose pricing cost information to external systems. The primary user of this information is assumed to be an Order Management System (OMS), which manages franchise customer orders and needs visibility to cost information as part of the negotiation process for margin visibility.

Pricing cost for an item at an owned location is the unit cost for the primary supplier/ country, less off invoice deals, plus estimated landed costs. Pricing cost for an item at a customer (franchise) location is the unit cost for the costing location, less any deals

passed through, plus estimated landed costs (based on system option), plus the franchise cost template details. This API supports providing cost information for an item/location or item/supplier/location.

Any failures (validation errors) encountered during the processing are passed back into the response object. If there are no failures, success status is returned.

Assumptions

- Only Approved and transaction level items are valid.
- Location must be company store or physical warehouse that is customer orderable. For a physical warehouse, it must also include the channel ID that should be used.
- For physical warehouses, the cost returned will be for the virtual warehouse that matches channel ID included in the inputs. If there is not a virtual warehouse that matches that channel in the physical warehouse, then next best match will be determined based on channel type and the primary warehouse and protected flags on the virtual warehouses in the physical warehouse.

Operation XSD

Here are the filenames that correspond with each operation. Please consult the RSB documentation for each in order to get a detailed picture of the composition.

Operation Name	Input Object Type	Output Object Type	XML Definition
lookup	PrcCostColCriVo	PrcCostColDesc	PrcCostCriVo.xsd PrcCostColCriVo.xsd PrcCostDesc.xsd PrcCostColDesc.xsd

Purchase Order Management Service

Functional Area

Purchase Order

RSB Proxy WSDL

```
/rms-PurchaseOrderManagement-AppServiceDecorator/ProxyService/  
PurchaseOrderManagementAppServiceProxy?wsdl
```

Merchandising Service WSDL

```
/PurchaseOrderManagementBean/PurchaseOrderManagementService?WSDL
```

Overview

The Purchase Order Management service allows an external systems to request pre-issued order numbers, create a purchase order, modify a purchase order, or delete purchase order details.

Request Order Numbers

The order reservation operation allows external systems such as Oracle Retail Assortment Planning (AP) to reserve order numbers in Merchandising to be used in orders that will later be created and integrated to Merchandising. This web service contains the following inputs:

Column Name	Notes
Supplier Site ID	Optional - used if the requesting entity is a supplier
Quantity	Indicates the number of order numbers being requested. Required.
Days Until Expiry	Indicates how long the calling solution wants Merchandising to retain the reservation. After this many days, the reservation will be released allowing these numbers to be used for other purposes. This is required and must be a value greater than 0.

The requested order numbers are sent back to the calling solution in the response. This operation is only available as part of the web service.

Create/Manage Purchase Orders

The operations supported in this service for creating and managing purchase orders are as follows:

- Create a purchase order header and details
- Modify purchase order header and details
- Delete purchase order details

For the operations, this service uses the same logic as is used in the PO Subscription RIB API. For more information on the functionality, see [PO Subscription API](#) in the "RIB Subscription Designs" section of this document.

Operation XSD

Here are the filenames that correspond with each operation. Please consult the RSB documentation for each in order to get a detailed picture of the composition.

Operation Name	Input Object Type	Output Object Type	XML Definition
preIssueOrderNumber	OrdNumCriVo	OrdNumColDesc	OrdNumCriVo.xsd OrdNumDesc.xsd OrdNumColDesc.xsd
create	XOrderColDesc	InvocationSuccess	XOrderDesc.xsd XOrderColDesc.xsd
modifyHeader	XOrderColDesc	InvocationSuccess	XOrderDesc.xsd XOrderColDesc.xsd
createDetail	XOrderColDesc	InvocationSuccess	XOrderDesc.xsd XOrderColDesc.xsd
modifyDetail	XOrderColDesc	InvocationSuccess	XOrderDesc.xsd XOrderColDesc.xsd

Operation Name	Input Object Type	Output Object Type	XML Definition
deleteDetail	XOrderColRef	InvocationSuccess	XOrderRef.xsd XOrderColRef.xsd

Report Locator Service

Functional Area

Financial Integration

RSB Proxy WSDL

`/rms-ReportLocator-AppServiceDecorator/ProxyService/ReportLocatorAppServiceProxy?wsdl`

Merchandising Service WSDL

`/ReportLocatorBean/ReportLocatorService?WSDL`

Overview

This service is used by Oracle Retail Financial Integration (RFI) to retrieve the URL of a BI Publisher report from Merchandising or Sales Audit that can be invoked from the PeopleSoft Financials General Ledger based on a particular journal entry. The report URL that will be returned will be differ based on the ID sent in the service call. Based on that ID, Merchandising will determine if it was a Merchandising, Sales Audit, or Invoice Matching ID and return a URL for the appropriate report. Possible reports for Merchandising and Sales Audit are:

- GL Fixed Deal Data Report
- GL Item level Data Report
- GL Item Rollup Daily Data Report
- GL Item Rollup Monthly Data Report
- GL Sales Audit Data Report

Also, for Invoice Matching, one of the following reports might be returned:

- Merchandise Invoice Document Report
- Non-Merchandise Invoice Document Report
- Credit Note Document Report
- Credit Memo Cost Document Report
- Credit Memo Quantity Document Report
- Debit Memo Cost Document Report
- Debit Memo Quantity Document Report
- Debit Memo VAT Document Report
- Receipt Write Off Document Report

For cloud service implementations, this configuration should be done for you if you are configured to run with PeopleSoft Financials. For on premise implementations, you may need to configure this yourself in the RETAIL_SERVICE_REPORT_URL table. For cloud service implementations, coordinate this configuration with the Oracle Cloud Operations team.

The report is similar functionality to the Drill Forward and Drill Back functionality available in the Merchandising Transaction Data and Fixed Deal pages and the Sales Audit General Ledger Transaction page. See also [Get Drill Back Forward URL Service](#) for more on the APIs that support this functionality.

Operation XSD

Here are the filenames that correspond with each operation. Please consult the RSB documentation for each in order to get a detailed picture of the composition.

Operation Name	Input Object Type	Output Object Type	XML Definition
publish	ReportLocDesc	ReportLocRef	ReportLocRef.xsd ReportLocDesc.xsd

Store Order Service

Functional Area

Procurement

RSB Proxy WSDL

`/rms-StoreOrder-AppServiceDecorator/ProxyService/StoreOrderAppServiceProxy?wsdl`

Merchandising Service WSDL

`/StoreOrderBean/StoreOrderService?WSDL`

Overview

This service is used by Oracle Retail Store Inventory Management (SIM) to create and manage store orders, as well as to query details to support these two operations.

Create Store Order

The majority of the operations in this service are related to creating, updating, or deleting a store order. A store order is a request from the store for inventory that can result in either a purchase order or transfer being created in Merchandising.

The Create operation allows SIM to request the creation of an order for inventory from either a supplier or warehouse for one or more items. If the source of the inventory will be the supplier, then the order can be for more than one store. For warehouse sourced orders, it will be for a single store.

The Create Detail operation allows SIM to request the addition of an item to a previously created transfer or an item/location to a previously created purchase order.

Orders will be created in either Approved or Worksheet/Input status in Merchandising, depending on what is sent from SIM.

Modify Store Order

The Modify and Modify Detail operations allow SIM to update a previously created transfer or purchase order. For this type of update, SIM must send the status.

Delete Store Order

The Delete and Delete Detail operations allow SIM to request a delete of a previously created order or an order line item. If the order is in approved status and it is being sourced from a supplier, then the result will be a modification of the order to cancel the quantity or full order, rather than delete it. If it is not yet approved, then order details or order can be deleted.

Query Deals

This operation allows SIM to query Merchandising for the deals that an item/store, based on a specific date and source (supplier, partner). Merchandising will reply with details on the off-invoice deals that the item/location are part of based on the date provided. The details provided include the deal dates and discount details.

Query Sales

This operation allows SIM to query Merchandising for a specific item/location combination's sales data. Merchandising will respond by sending the available weeks of sales data, including the quantity sold, retail value, and sales type (for example, regular, promotion, clearance).

Query Store Orders

There are two operations that allow SIM to query store orders from Merchandising. Query Store Order accepts location and location type (store or warehouse), as well as optional filtering details like item, source (supplier site or physical warehouse), dates, and status. The operation returns a collection of header level details for the purchase orders or transfers that match the criteria, including the quantity on the order for the location.

The other operation, Query Store Order Details, accepts a specific order (purchase order or transfer), source type, and source and returns the details of that order, including the destination locations, status, dates, items, cost, and quantity.

Assumptions

- Service operations will return back with the first error encountered.

Operation XSD

Here are the filenames that correspond with each operation. Please consult the RSB documentation for each in order to get a detailed picture of the composition.

Operation Name	Input Object Type	Output Object Type	XML Definition
create	LocPOTsfDesc	LocPOTsfRef	LocPOTsfDesc.xsdLocPOTsfRef.xsd
createDetail	LocPOTsfDesc	InvocationSuccess	LocPOTsfDesc.xsd
modify	LocPOTsfDesc	InvocationSuccess	LocPOTsfDesc.xsd
modifyDetail	LocPOTsfDesc	InvocationSuccess	LocPOTsfDesc.xsd
delete	LocPOTsfDesc	InvocationSuccess	LocPOTsfDesc.xsd
deleteDetail	LocPOTsfDesc	InvocationSuccess	LocPOTsfDesc.xsd
queryDeal	LocPOTsfDealsCriVo	LocPOTsfDealsColDesc	LocPOTsfDealsCriVo.xsd LocPOTsfDealsDesc.xsd LocPOTsfDealsColDesc.xsd
queryItemSales	LocPOTsfItmSlsCriVo	LocPOTsfItmSlsColDesc	LocPOTsfItmSlsCriVo.xsd LocPOTsfItmSlsDesc.xsd LocPOTsfItmSlsColDesc.xsd
queryStoreOrder	LocPOTsfHdrCriVo	LocPOTsfHdrColDesc	LocPOTsfHdrCriVo.xsd LocPOTsfHdrDesc.xsd LocPOTsfHdrColDesc.xsd
queryStoreOrderDetail	LocPOTsfDtIsCriVo	LocPOTsfDesc	LocPOTsfDtIsCriVo.xsdLocPOTsfDesc.xsd

Supplier Service

Functional Area

Foundation Data

RSB Proxy WSDL

/rms-Supplier-AppServiceDecorator/ProxyService/SupplierAppServiceProxy?wsdl

Merchandising Service WSDL

/SupplierBean/SupplierService?WSDL

Overview

This service allows Merchandising to subscribe to supplier information from external financial applications. It is also used by Oracle Retail Financials Integration (RFI) for integrating supplier information into Merchandising from EBS, PeopleSoft, or Cloud Financials. The operations supported by this service are as follows:

- Create a new parent supplier, including the associated sites, org unit, and address; it also supports adding flex attributes (CFAS) for supplier, supplier site, and address levels
- Update an existing supplier, including adding or updating sites, org unit, address for the supplier, and flex attributes for the supplier, sites, and address levels

The operations supported by this service involve an external system sending Merchandising details to create or update suppliers or supplier sites. In the response back, the supplier or site IDs are returned on success. The create and update options for this service use the same logic as is supported in the Vendor Subscription RIB API.

For information about this functionality, see [Vendor Subscription API](#) in the "RIB Subscription Designs" chapter of this document.

Operation XSD

Here are the filenames that correspond with each operation. Please consult the RSB documentation for each in order to get a detailed picture of the composition.

Operation Name	Input Object Type	Output Object Type	XML Definition
create	SupplierDesc	SupplierRef	SupplierRef.xsd SupplierDesc.xsd
create	SupplierColDesc	SupplierColRef	SupplierColDesc.xsdSupplierColRef.xsd
update	SupplierDesc	SupplierDesc	SupplierDesc.xsd
update	SupplierColDesc	SupplierColDesc	SupplierColDesc.xsd

Transfer Service

Functional Area

Transfer

RSB Proxy WSDL

```
/rms-TransferManagement-AppServiceDecorator/ProxyService/  
TransferManagementAppServiceProxy?wsdl
```

Merchandising Service WSDL

```
/TransferManagementBean/TransferManagementService?WSDL
```

Overview

Merchandising exposes a Transfer Management service to allow an external application to create, update, and delete transfers. The web service takes in a collection of transfers and will return success and failure through the service response object. The operations supported in this service for creating and managing transfers are as follows:

- Create a transfer header and details
- Modify transfer header and details

- Delete transfer header and details

For the operations, this service uses the same logic as is used in the Transfer Subscription RIB API. For more information on the functionality, see [Transfer Subscription API](#) in the "RIB Subscription Designs" chapter of this document.

Operation XSD

Here are the filenames that correspond with each operation. Please consult the RSB documentation for each in order to get a detailed picture of the composition.

Operation Name	Input Object Type	Output Object Type	XML Definition
create	XTsfColDesc	InvocationSuccess	XTsfDesc.xsd XTsfColDesc.xsd
createDetail	XTsfColDesc	InvocationSuccess	XTsfDesc.xsd XTsfColDesc.xsd
modifyHeader	XTsfColDesc	InvocationSuccess	XTsfDesc.xsd XTsfColDesc.xsd
modifyDetail	XTsfColDesc	InvocationSuccess	XTsfDesc.xsd XTsfColDesc.xsd
deleteHeader	XTsfColRef	InvocationSuccess	XTsfRef.xsd XTsfColRef.xsd
deleteDetail	XTsfColRef	InvocationSuccess	XTsfRef.xsd XTsfColRef.xsd

Consumer Services

This section gives an overview about the SOAP Web service Consumer Implementation API designs used in the Merchandising environment and various functional attributes used in the APIs.

Customer Address Service

Functional Area

Financials

Overview

The primary role of this service is to query customer address details related to a Sales Audit transaction. This may be required if you have configured Merchandising to not retain customer information (Retain Customer Information system options unchecked) for customer orders, but you wish to have visibility to it in Sales Audit when viewing/auditing transactions.

When Sales Audit calls this service, it will pass the customer ID and expect to receive back the following information in response:

- First Name

- Last Name
- Company Name (if applicable)
- Address Line 1
- Address Line 2
- Address Line 3
- County
- City
- State
- Country
- Postal Code
- Jurisdiction
- Phone
- Email
- Birthdate

As part of your implementation, if you have this system option configured off, you will need to provide a URL for Sales Audit to call. For on premise implementations, this will require updating the RETAIL_SERVICE_REPORT_URL table for code CAS. For cloud service implementations, configuration of this service call should be done in coordination with the Oracle Cloud Operations team by logging an SR.

Customer Order Address Service

Functional Area

Procurement

Overview

The primary role of this service is for Merchandising to query customer/shipping details related to a customer order from an order management system (OMS). This is required if you have configured Merchandising to not retain customer information (Retain Customer Information system options unchecked) for customer orders and are sourcing customer orders from a warehouse or supplier where Merchandising needs to provide the address details for shipping to the customer.

When Merchandising calls this service, it will pass

- Customer order number
- Fulfillment order number
- "Fulfillment location type and ID
- Source location type and ID

And expect to receive back the following information in response:

- Customer ID

- Delivery Details
 - First Name
 - Phonetic First Name
 - Last Name
 - Phonetic Last Name
 - Preferred Name
 - Company Name
 - Address Line 1
 - Address Line 2
 - Address Line 3
 - County
 - City
 - State
 - Country
 - Postal Code
 - Jurisdiction
 - Phone
 - Email
- Billing Details
 - First Name
 - Phonetic First Name
 - Last Name
 - Phonetic Last Name
 - Preferred Name
 - Company Name
 - Address Line 1
 - Address Line 2
 - Address Line 3
 - County
 - City
 - State
 - Country
 - Postal Code
 - Jurisdiction
 - Phone
 - Email

As part of your implementation, if you have this system option configured off, you will need to provide a URL for Merchandising to call. For on premise implementations, this will require updating the RETAIL_SERVICE_REPORT_URL table for code COA. For cloud service implementations, configuration of this service call should be done in coordination with the Oracle Cloud Operations team by logging an SR.

Get Drill Back Forward URL Service

Functional Area

Financial Integration

Overview

If you are implementing the Merchandising solutions with PeopleSoft Financials, then this service can be used allow users in Merchandising or Sales Audit to "drill forward" into Peoplesoft to view the General Ledger journal entries associated with a transaction or to "drill back" into Merchandising and Sales Audit from PeopleSoft General Ledger screens to view the source transactions associated with a journal entry. Both of these actions leverage this service call.

If you are configured use Merchandising with PeopleSoft Financials, then when you are in the following pages, you'll see Drill to Finance options that leverage this call:

- Merchandising Transaction Data
- Merchandising Fixed Deals
- Sales Audit General Ledger Transactions



Note:

Oracle Retail Invoice Matching also leverages this service for viewing transactions in PeopleSoft Payables. And Peoplesoft Payables can drill back to Invoice Matching as well.

As part of your implementation, you will need to configure the URL for the service call in the RETAIL_SERVICE_REPORT_URL table for code RDF. For cloud service implementations, configuration of this service call should be done in coordination with the Oracle Cloud Operations team by logging an SR. For more information, see the *RFI Implementation Guide*.

GL Chart of Accounts Validation Service

Functional Area

Financial Integration

Overview

When using Oracle Retail Financials Integration (RFI) to manage General Ledger integration an Oracle financial solution, a validation service is used to ensure that the segment combinations mapped to by Merchandising and Sales Audit users are valid combinations in the General Ledger. This validation is called from Merchandising and Sales Audit when creating General Ledger cross-reference mappings.

 **Note:**

This validation is also used by Oracle Retail Invoice Matching

As part of your implementation, you will need to configure the URL for the service call in the RETAIL_SERVICE_REPORT_URL table for code RAV. For cloud service implementations, configuration of this service call should be done in coordination with the Oracle Cloud Operations team by logging an SR. For more information, see the *RFI Implementation Guide*.

5

ReSTful Web Services

This chapter gives an overview about the Merchandising and Sales Audit ReSTful Web service implementation and the APIs used in Merchandising and Sales Audit. For more information on ReST architectural style applied for building Web services, access the following URL:

<http://www.oracle.com/technetwork/articles/javase/index-137171.html>

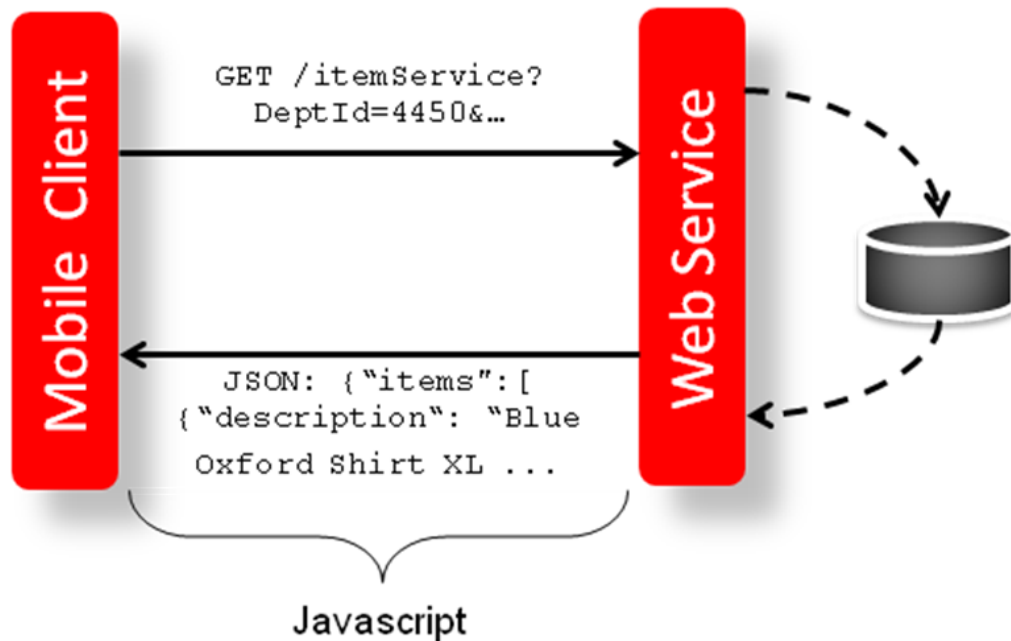
To provide visibility to the background processing that's occurring, services write to the JOB_AUDIT_LOGS table in the database. Reports can be built based on this to provide visibility to what is happening in the background. Additionally, to assist users and developers in troubleshooting any error that may arise, the payload that was processed is also stored in the JOB_AUDIT_PAYLOAD table in the database.

Once the nightly batch run has started, web service execution will be halted, and users will receive a warning message that the nightly batch run has commenced.

Introduction

Merchandising and Sales Audit ReST support several web services, including the ability to query data and the ability to create and update data within the solutions. A few were built specifically to support mobile applications. These may not be useful for general use, however if you wanted to build your own mobile applications leveraging these services, this can be done. The ReSTful Web services Java code cannot be customized. The diagram below shows how the services are intended to interact with a mobile client.

Figure 5-1 Mobile Client and Web Services Integration through Javascript



 **Note:**

The services should not be used during the restricted batch window.

Common Characteristics

Security

Services are secured using J2EE-based security model.

- Realm-based User Authentication: This verifies users through an underlying Realm. The username and password are passed using HTTP basic authentication.
- Role-based Authorization: This assigns users to roles; authenticated users can access the services with Merchandising or Sales Audit application roles or custom roles that are assigned to:
 - For Merchandising `MERCH_SERVICE_ACCESS_PRIV`
 - For Sales Audit `MERCH_SERVICE_ACCESS_PRIV`
- The communication between the server and client is encrypted using one-way SSL. In non-SSL environments the encoding defaults to BASE-64 so it is highly recommended that these ReST services are configured to be used in production environments secured with SSL connections.
- If you are using Merchandising data filtering, that will apply to the services as well. The user ID used for the calling the service should be added to the Merchandising `SEC_USER` table (`APP_USER_ID`), and then associated to the appropriate group

in SEC_USER_GROUP table. For more information on this see the *Merchandising Security Guide - Volume 2*.

Standard Request and Response Headers

Merchandising and Sales Audit ReSTful web services have the following standard HTTP headers:

```
Accept: application/xml or application/JSON
Accept-Language: en-US,en;q=0.8
```



Note:

Accept-Language is not mandatory, and defaults to en-US. User can change it though, in case they need content in a specific language.

Depending on the type of the operation or HTTP method, the corresponding response header is updated in the HTTP response with the following codes:

- GET/READ : 200
- PUT/CREATE : 201 created
- POST/UPDATE : 204
- DELETE : 204

Standard Error Response

Example response payload in case of service error is shown below:

```
<?xml version="1.0" encoding="UTF-8" standalone="yes"?>
<messagesRDOes>
  <messagesRDO>
    <message>REST Service Version Mismatch</message>
    <messageType>ERROR</messageType>
    <status>BAD_REQUEST</status>
  </messagesRDO>
</messagesRDOes>
```

- **Message:** The error message - translated.
- **MessageType:** Value of 'ERROR' is returned.
- **Status:** For a bad request or error, the status is BAD_REQUEST.
- The http error code for an error response is 400.

Merchandising URL Paths

Based on the Implementation you may need to prefix the end point with just the deployment hostname or hostname plus access port.

The following links provide access to the Merchandising services:

- The ReSTful Web services WADL file is available at:

```
https://<hostname>/RmsReSTServices/services/application.wadl
```

- The ReSTful Web services are available at:

```
https://<hostname>/RmsReSTServices/services/private/<service>
```

Fiscal Document Generation URL Paths

Based on your implementation you will need to prefix the end point with just the deployment hostname. The format that should be used for the hostname is `xxx-yyy-mfcs-rhs.oracleindustry.com`, where `xxx-yyy` is specific to your company's name and environment (production, stage, and so on).

The following links provide access to the Fiscal Document Generation services:

- The ReSTful Web services WADL file is available at:

```
https://<hostname>/RfmReSTServices/services/application.wadl
```

- The ReSTful Web services are available at:

```
https://<hostname>/RfmReSTServices/services/private/<service>
```

Sales Audit URL Paths

Based on the Implementation you may need to prefix the end point with just the deployment hostname or hostname plus access port.

The following links provide access to the Sales Audit services:

- The ReSTful Web services WADL file is available at:

```
https://<hostname>/ResaReSTServices/services/application.wadl
```

- To access the ReSTful web services:

```
https://<hostname>/ResaReSTServices/services/private/Resa/<service>
```

Date Format

Few input date and output date fields are in long format. The others are in SQL Date format.

Paging

Some of the Merchandising and Sales Audit ReSTful web services have the potential to bring back a significant number of records, and therefore these services are equipped to segment the result into pages. The page number to retrieve and the size of the page are added as input parameters to all the paged services.

Each paged result includes the following information:

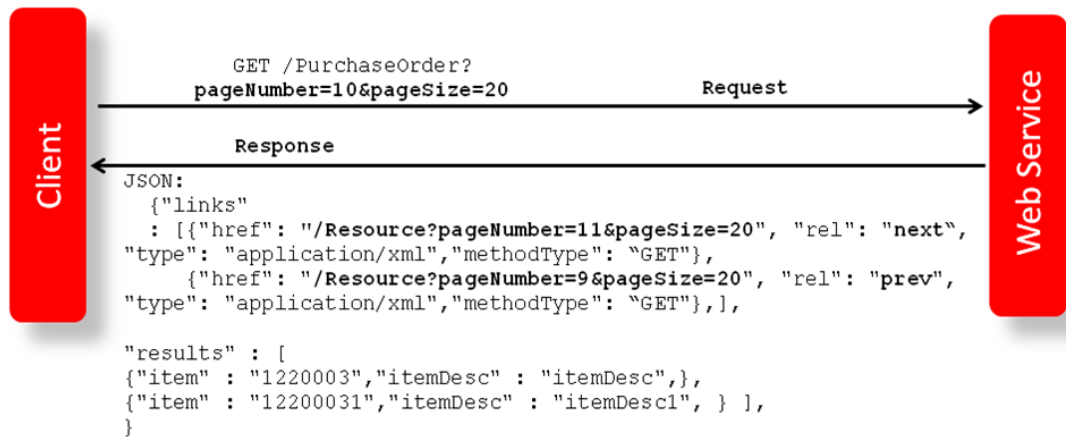
- **Total Record Count:** Displays the number of all records matching the service input criteria.
- **Next Page URL:** Shows the service URL with same input parameters, but with the pageNumber plus 1, when more records exist.

- **Previous Page URL:** Shows the service URL with same input parameters and the pageNumber input value minus 1, when page number is not 1.

Next or previous page URL is not provided when:

- No records are returned.
- Previous page is not returned, when the page number is 1.
- Next page is not returned, when the record reaches the last page.

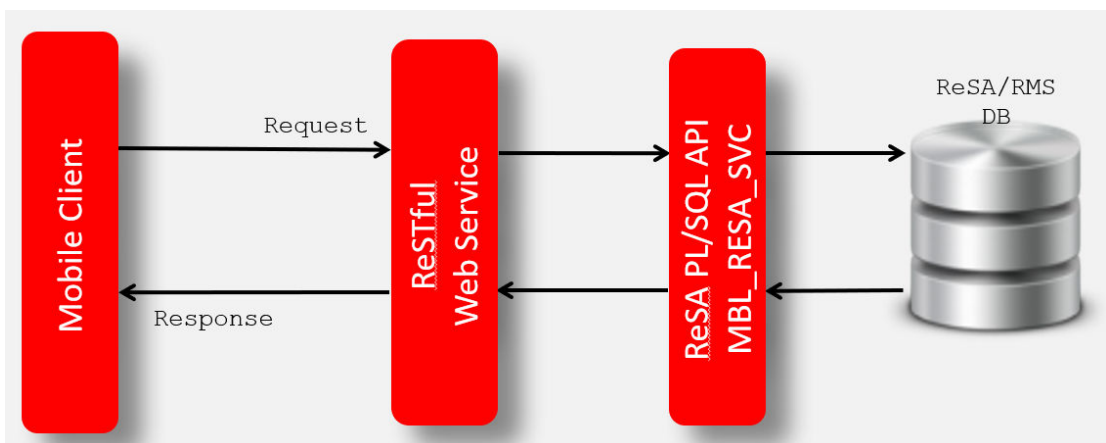
Figure 5-2 Javascript for Paging Information in Merchandising Web Services



Web Service APIs Process Flow

The diagram shows the Web Service API process flow.

Figure 5-3 Web Service APIs Process Flow



Merchandising ReSTful Web Services

Merchandising Common Services

This section describes the GA account validation service.

Functional Area

Foundation

Business Overview

The primary role of this service is to provide access to cross-functional Merchandising data.

Vdate

Business Overview

Retrieve Merchandising Vdate.

Service Type

Get

ReST URL

/Common/vDate

Input Parameters

NA

Output

Vdate in Long and Date Format

Parameter Name	Data Type
Vdate	Long
Vdate	Date

JSON Structure:

```
"{  
  "vdateDisplay": "01-Jul-2019",  
  "vdate": 1561939200000,  
  "links": [],  
  "hyperMediaContent": {  
    "linkRDO": []  
  }  
}"
```

Table Impact

TABLE	SELECT	INSERT	UPDATE	DELETE
PERIOD	Yes	No	No	No
JOB_AUDIT_LOGS	No	Yes	No	No
JOB_AUDIT_PAYLOAD	No	Yes	No	No

Procurement Unit Options

Business Overview

Retrieve Merchandising's Procurement Unit Options.

Service Type

Get

ReST URL

/Common/POSysOps

Input Parameters

NA

Output

ProcurementUnitOptionsRDO

Parameter Name	Data Type
backpostRcaRualnd	String
calcNegativeIncome	String
copyPoCurrRate	String
costLevel	String
creditMemoLevel	String
dealAgePriority	String
dealLeadDays	BigDecimal
dealTypePriority	String
deptLevelOrders	String
ediCostOverrideInd	String
expiryDelayPreIssue	BigDecimal
genConsignmentInvcFreq	String
genConInvcItnSupLocInd	String
latestShipDays	BigDecimal
ordApprCloseDelay	BigDecimal
ordApprAmtCode	String
ordAutoClosePartRcvdInd	String
ordPartRcvdCloseDelay	BigDecimal
orderBeforeDays	BigDecimal
orderExchInd	String

Parameter Name	Data Type
otbSystemInd	String
rcvCostAdjType	String
reclassApprOrderInd	String
redistFactor	BigDecimal
softContractInd	String
wacRecalcAdjInd	String

JSON Structure:

```
{
  "links": [],
  "backpostRcaRuaInd": "N",
  "billToLoc": "1000",
  "calcNegativeIncome": "N",
  "copyPoCurrRate": null,
  "costLevel": "DNN",
  "creditMemoLevel": "D",
  "dealAgePriority": "O",
  "dealLeadDays": 1,
  "dealTypePriority": "P",
  "deptLevelOrders": "N",
  "ediCostOverrideInd": "Y",
  "expiryDelayPreIssue": 30,
  "genConsignmentInvcFreq": "M",
  "genConInvcItmSupLocInd": "I",
  "latestShipDays": 30,
  "ordApprCloseDelay": 1,
  "ordApprAmtCode": "C",
  "ordAutoClosePartRcvdInd": "N",
  "ordPartRcvdCloseDelay": 1,
  "orderBeforeDays": 5,
  "orderExchInd": "N",
  "otbSystemInd": "N",
  "rcvCostAdjType": "F",
  "reclassApprOrderInd": "Y",
  "redistFactor": 2,
  "softContractInd": "Y",
  "wacRecalcAdjInd": "N",
  "hyperMediaContent": {
    "linkRDO": []
  }
}
```

Table Impact

TABLE	SELECT	INSERT	UPDATE	DELETE
PROCUREMENT_UNIT_OPTIONS	Yes	No	No	No
JOB_AUDIT_LOGS	No	Yes	No	No
JOB_AUDIT_PAYLOAD	No	Yes	No	No

Functional Config Options

Business Overview

Retrieve Merchandising's Functional Config Options.

Service Type

Get

ReST URL

/Common/FuncSysOps

Input Parameters

NA

Output

FunctionalConfigRDO

Parameter Name	Data Type
importInd	String
orgUnitInd	String
supplierSitesInd	String
contractInd	String
elcInd	String

JSON Structure:

```
{
  "links": [],
  "importInd": "Y",
  "orgUnitInd": "Y",
  "supplierSitesInd": "Y",
  "contractInd": "Y",
  "elcInd": "Y",
  "hyperMediaContent": {
    "linkRDO": []
  }
}
```

Table Impact

TABLE	SELECT	INSERT	UPDATE	DELETE
FUNCTIONAL_CONFIG_OPTIONS	Yes	No	No	No
JOB_AUDIT_LOGS	No	Yes	No	No

TABLE	SELECT	INSERT	UPDATE	DELETE
JOB_AUDIT_PAYLOAD	No	Yes	No	No

Inventory Movement Unit Options

Business Overview

Retrieve Merchandising's Inventory Movement Unit Options.

Service Type

Get

ReST URL

/Common/InvMovSysOps

Input Parameters

NA

Output

InvMoveUnitOptRDO

Parameter Name	Data Type
allocMethod	String
applyProfPresStock	String
autoRcvStore	String
closeOpenShipDays	BigDecimal
costMoney	BigDecimal
costOutStorage	BigDecimal
costOutStorageMeas	String
costOutStorageUom	String
costWhStorage	BigDecimal
costWhStorageMeas	String
costWhStorageUom	String
defaultAllocChrgInd	String
defaultOrderType	String
defaultSizeProfile	String
deptLevelTransfers	String
distributionRule	String
duplicateReceivingInd	String
increaseTsfQtyInd	String
intercompanyTransferBasis	String
invHistLevel	String
locActivityInd	String
locDivryInd	String

Parameter Name	Data Type
lookAheadDays	BigDecimal
maxWeeksSupply	BigDecimal
ordWorksheetCleanUpDelay	BigDecimal
racRtvTsfInd	BigDecimal
rejectStoreOrdInd	String
replOrderDays	String
rtvNadLeadTime	BigDecimal
rtvUnitCostInd	BigDecimal
shipRcvStore	String
shipRcvWh	String
storageType	String
storePackCompRcvInd	String
wfDefaultWh	String
targetRoi	BigDecimal
tsfAutoCloseStore	BigDecimal
tsfAutoCloseWh	String
tsfCloseOverdue	String
simForceCloseInd	String
tsfForceCloseInd	String
tsfOverReceiptInd	String
tsfMdStoreToStoreSndRcv	String
tsfMdStoreToWhSndRcv	String
tsfMdWhToStoreSndRcv	String
tsfMdWhToWhSndRcv	String
tsfPriceExceedWacInd	String
ssAutoCloseDays	String
wsAutoCloseDays	BigDecimal
swAutoCloseDays	BigDecimal
wwAutoCloseDays	BigDecimal
wfOrderLeadDays	BigDecimal
whCrossLinkInd	BigDecimal
wrongStReceiptInd	String

JSON Structure:

```
{
  "links": [],
  "allocMethod": "P",
  "applyProfPresStock": "N",
  "autoRcvStore": "Y",
  "closeOpenShipDays": 3,
  "costMoney": 7.5,
  "costOutStorage": 1.5,
  "costOutStorageMeas": "P",
```

```

    "costOutStorageUom": null,
    "costWhStorage": 1.5,
    "costWhStorageMeas": "P",
    "costWhStorageUom": null,
    "defaultAllocChrgInd": "Y",
    "defaultOrderType": "WAVE",
    "defaultSizeProfile": "N",
    "deptLevelTransfers": "Y",
    "distributionRule": "PRORAT",
    "duplicateReceivingInd": "N",
    "increaseTsfQtyInd": "N",
    "intercompanyTransferBasis": "T",
    "invHistLevel": "A",
    "locActivityInd": "Y",
    "locDlvryInd": "Y",
    "lookAheadDays": 7,
    "maxScalingIterations": null,
    "maxWeeksSupply": 5,
    "ordWorksheetCleanupDelay": 1,
    "racRtvTsfInd": "A",
    "rejectStoreOrdInd": "N",
    "replOrderDays": 3,
    "rtvNadLeadTime": 1,
    "rtvUnitCostInd": "A",
    "shipRcvStore": "Y",
    "shipRcvWh": "Y",
    "storageType": "W",
    "storePackCompRcvInd": "Y",
    "wfDefaultWh": 1212,
    "targetRoi": 7,
    "tsfAutoCloseStore": "Y",
    "tsfAutoCloseWh": "Y",
    "tsfCloseOverdue": "Y",
    "simForceCloseInd": "NL",
    "tsfForceCloseInd": "SL",
    "tsfOverReceiptInd": "NL",
    "tsfMdStoreToStoreSndRcv": "S",
    "tsfMdStoreToWhSndRcv": "S",
    "tsfMdWhToStoreSndRcv": "S",
    "tsfMdWhToWhSndRcv": "S",
    "tsfPriceExceedWacInd": "Y",
    "ssAutoCloseDays": 1,
    "wsAutoCloseDays": 1,
    "swAutoCloseDays": 1,
    "wwAutoCloseDays": 1,
    "wfOrderLeadDays": null,
    "whCrossLinkInd": "Y",
    "wrongStReceiptInd": "Y",
    "hyperMediaContent": {
      "linkRDO": []
    }
  }
}

```

Table Impact

TABLE	SELECT	INSERT	UPDATE	DELETE
INV_MOVE_UNIT_OPTIONS	Yes	No	No	No
JOB_AUDIT_LOGS	No	Yes	No	No

TABLE	SELECT	INSERT	UPDATE	DELETE
JOB_AUDIT_PAYLOAD	No	Yes	No	No

Currencies

Business Overview

Retrieve Merchandising's Currencies table records.

Service Type

Get

ReST URL

/Common/Currencies

Input Parameters

NA

Output

MblCurrenciesRDO

Parameter Name	Data Type
currencyCode	String
currencyDescription	String
currencyCostFormat	String
currencyRetailFormat	String
currencyCostDecimal	BigDecimal
currencyRetailDecimal	BigDecimal

JSON Structure:

```
[
  {
    "links": [],
    "currencyCode": "AED",
    "currencyDescription": "U.A.E. Dirham",
    "currencyCostFormat": "FM9G999G999G999G990D9099PR",
    "currencyRetailFormat": "FM9G999G999G999G990D90PR",
    "currencyCostDecimal": 4,
    "currencyRetailDecimal": 2,
    "hyperMediaContent": {
      "linkRDO": []
    }
  },
  {
    "links": [],
    "currencyCode": "ALL",
    "currencyDescription": "UNKNOWN",
    "currencyCostFormat": "FMD0",
    "currencyRetailFormat": "FMD90",
    "currencyCostDecimal": 2,
    "currencyRetailDecimal": 2,
  }
]
```

```

        "hyperMediaContent": {
            "linkRDO": []
        }
    },
    .....

```

Table Impact

TABLE	SELECT	INSERT	UPDATE	DELETE
CURRENCIES	Yes	No	No	No
JOB_AUDIT_LOGS	No	Yes	No	No
JOB_AUDIT_PAYLOAD	No	Yes	No	No

Department Search

Business Overview

This service retrieves departments with ID or name matching search string.

Service Type

Get

ReST URL

/Common/departmentSearch?
searchString={searchString}&pageSize={pageSize}&pageNumber={pageNumber}

Input Parameters

Parameter Name	Required	Description	Valid values
searchString	Yes	search string for department Id or Name	
PageSize	No	Maximum number of records to retrieve per page	
PageNumber	No	Result page to retrieve	

Output

MerchHierDeptRDO

Parameter Name	Data Type
department	BigDecimal
departmentName	String

PagedResultsRDO

Parameter Name	Data Type
totalRecordCount	BigDecimal
Next Page URL	String
Previous Page URL	String

JSON Structure:

```
"{
  "type": "paginationRDO",
  "totalRecordCount": 3512,
  "hyperMediaContent": {},
  "links" : [],
  "results": [{
    "departmentId": 3252,
    "departmentDescription": "some description"
  }]
}"
```

Table Impact

TABLE	SELECT	INSERT	UPDATE	DELETE
V_DEPS	Yes	No	No	No
JOB_AUDIT_LOGS	No	Yes	No	No
JOB_AUDIT_PAYLOAD	No	Yes	No	No

Department Load

Business Overview

This service retrieves departments' name of input IDs.

Service Type

Get

ReST URL

/Common/departmentLoad?departments={departments}

Input Parameters

Parameter Name	Required	Description	Valid values
departments	Yes	Comma separated values for Departments	NA

Output

MerchHierDeptRDO

Parameter Name	Data Type
department	BigDecimal
departmentName	String

JSON Structure:

```
"{
  "departmentId": 3252,
  "departmentDescription": "some description"
}"
```

Table Impact

TABLE	SELECT	INSERT	UPDATE	DELETE
V_DEPS	Yes	No	No	No
JOB_AUDIT_LOGS	No	Yes	No	No
JOB_AUDIT_PAYLOAD	No	Yes	No	No

Allocation Detail Service

This section describes the Allocation Detail service.

Business Overview

Allocation Detail service allows user to retrieve Allocation information for a selected allocation number.

Service Type

Get

ReST URL

Alloc/allocDetail?allocNumber={allocationNumber}

Input Parameters

Parameter Name	Required	Description
allocNumber	Yes	Allocation Number

Output

RestAllocRecRDO

Parameter Name	Data Type
alloc_no	BigDecimal
order_no	BigDecimal
wh	BigDecimal
item	String
status	String
alloc_desc	String
po_type	String
alloc_method	String
release_date	Date
order_type	String
doc	String
doc_type	String

Parameter Name	Data Type
origin_ind	String
close_date	Date
alloc_detail	List<RestAllocDetailRecRDO>

RestAllocDetailRecRDO

Parameter Name	Data Type
to_loc	BigDecimal
to_loc_type	String
qty_transferred	BigDecimal
qty_allocated	BigDecimal
qty_prescaled	BigDecimal
qty_distro	BigDecimal
qty_selected	BigDecimal
qty_cancelled	BigDecimal
qty_received	BigDecimal
qty_reconciled	BigDecimal
po_rcvd_qty	BigDecimal
non_scale_ind	String
in_store_date	Date
wf_order_no	BigDecimal
rush_flag	String

JSON Structure:

```
[
  {
    "docType": null,
    "allocDetail": [
      {
        "qtyTransferred": null,
        "rushFlag": null,
        "wfOrderNo": null,
        "inStoreDate": null,
        "qtyAllocated": null,
        "nonScaleInd": null,
        "toLoc": null,
        "qtyPrescaled": null,
        "toLocType": null,
        "qtyDistro": null,
        "qtySelected": null,
        "qtyReceived": null,
        "qtyCancelled": null,
        "qtyReconciled": null,
        "poRcvdQty": null,
        "links": [],
        "hyperMediaContent": {
          "linkRDO": []
        }
      }
    ]
  }
]
```

```

    }
  ],
  "doc": null,
  "originInd": null,
  "allocNo": null,
  "wh": null,
  "allocMethod": null,
  "allocDesc": null,
  "poType": null,
  "item": null,
  "status": null,
  "orderNo": null,
  "orderType": null,
  "releaseDate": null,
  "closeDate": null,
  "links": [],
  "hyperMediaContent": {
    "linkRDO": []
  }
}
]

```

Table Impact

TABLE	SELECT	INSERT	UPDATE	DELETE
ALLOC_HEADER	Yes	No	No	No
ALLOC_DETAIL	Yes	No	No	No
JOB_AUDIT_LOGS	No	Yes	No	No
JOB_AUDIT_PAYLOAD	No	Yes	No	No

Background Process Configuration

This section describes the Background Process Configuration.

Business Overview

This service is used to update the configuration for each background jobs in Merchandising.

Service Type

Post

ReST URL

processes/update/process_config/execution

Input Parameters

Parameter Name	Required	Description
JobName	Yes	Job Name

Parameter Name	Required	Description
numThreads	No	Maximum number of threads the job will execute
numDataToProcess	No	Number of records a jobs will process each run.
commitMaxCtr	No	Max number of records processed before a commit is issued.
archiveInd	No	This field will be used to determine if associated tables for this job needs to be archived to history or not.

Output

NA

Table Impact

TABLE	SELECT	INSERT	UPDATE	DELETE
B8D_PROCESS_CONFIG	No	No	Yes	No
JOB_AUDIT_LOGS	No	Yes	No	No
JOB_AUDIT_PAYLOAD	No	Yes	No	No

Book Transfer ReSTful Web Service

This section describes the Book Transfer ReSTful Web Service

Functional Area

Transfer and Customer Order

Business Overview

This web service will be built to virtually move inventory from one location to the other for the purposes of attributing the sale to a location different from the location that is fulfilling the order physically. For example, if the order is being fulfilled via shipment from a physical store, some retailers will want to actually have the sale processed against the e-commerce store. This service also accept a customer order number and fulfillment order number to be associated with the transfer when it is created as a cross reference.

Service Type

Post

ReST URL

/Transfer/customerOrderBookTransfer

Input Parameters

The Book Transfer web service has the following parameters:

Parameter Name	Required	Data Type	Description
FromLocation	Yes	BigDecimal	Transfer source location
ToLocation	Yes	BigDecimal	Transfer destination
CustomerOrderNumber	No	String	Customer order identification
FulfillOrderNumber	No	String	Fulfillment order identification
UpdateCustomerReservedQty	No	String	Indicates if any of the customer reserved quantity should be update either the source or destination location or both. Valid values: <ul style="list-style-type: none"> • B update both the source and destination location • S update only the source location • R update only the destination location • N or NULL no update
ItemsDetail	Yes		Collection of itemsDetail RDO
ApprovalDate	No	Date	Approval date of the transfer (Format: 'DD-MON-YYYY')
Userld	No	String	Indicates the user that performed the transaction

ItemDetail RDO

The Book Transfer web service has the following parameters:

Parameter Name	Required	Data Type	Description
Item	Yes	String	Item identification
Quantity	Yes	BigDecimal	Item quantity to be transferred

Example JSON Input

```
[
  {
    "toLocation":null,
    "fromLocation":null,
    "customerOrderNumber":null,
    "fulfillOrderNumber":null,
    "updateCustomerReservedQty":null,
    "itemsDetail":[
      {
        "item":null,
        "quantity":null
      },
      {
        "item":null,
        "quantity":null
      }
    ]
  }
]
```

```

    },
    {
      "item":null,
      "quantity":null
    }
  ],
  "approvalDate":null
  "userId":null
},
{
  "fromLocation":null,
  "toLocation":null,
  "customerOrderNumber":null,
  "fulfillOrderNumber":null,
  "updateCustomerReservedQty":null,
  "itemsDetail":[
    {
      "item":null,
      "quantity":null
    }
  ],
  "approvalDate":null
  "userId":null
}
]

```

Output

RestCobtsfStatuRDO

Parameter Name	Data Type
successCobtsfCount	Big Decimal
successCobtsfTbl	List< successCobtsfRDO>
failCobtsfCount	BigDecimal
failCobtsfTable	List<failCobtsfRDO>

SuccessCobtsfRDO

Parameter Name	Data Type
FromLocation	Big Decimal
ToLocation	Big Decimal
TransferNumber	BigDecimal

FailCobtsfRDO

Parameter Name	Data Type
FromLocation	Big Decimal
ToLocation	Big Decimal
errorMessage	BigDecimal

JSON Structure

```

{
  "successCobtsfCount": 2,
  "successCobtsfTable": [
    {
      "fromLocation ": 123,
      "toLocation ": 987,
      "transferNumber ": 123456789
      "links": [],
      "hyperMediaContent": {
        "linkRDO": []
      }
    },
    {
      "fromLocation ": 456,
      "toLocation ": 654,
      "transferNumber ": 987654321
      "links": [],
      "hyperMediaContent": {
        "linkRDO": []
      }
    }
  ],
  "failCobtsfCount": 1,
  "failCobtsfTable": [
    {
      "orderNumber": 123,
      "errorMessage": "Invalid Item.",
      "links": [],
      "hyperMediaContent": {
        "linkRDO": []
      }
    }
  ],
  "links": [],
  "hyperMediaContent": {
    "linkRDO": []
  }
}

```

Table Impact

The following tables are affected:

TABLE	SELECT	INSERT	UPDATE	DELETE
CURRENCIES	Yes	No	No	No
DEPS	Yes	No	No	No
ITEM_LOC	Yes	Yes	No	No
ITEM_LOC_SOH	Yes	Yes	Yes	No
ITEM_MASTER	Yes	No	No	No
ITEM_SUPP_COUNTRY	Yes	No	No	No
MONTH_DATA	Yes	No	No	No
MV_CURRENCY_CONVERSION_RATES	Yes	No	No	No

TABLE	SELECT	INSERT	UPDATE	DELETE
ORDCUST	Yes	Yes	No	No
STORE	Yes	No	No	No
TRAN_DATA	No	Yes	No	No
TSFDETAIL	No	Yes	No	No
TSFHEAD	No	No	No	No
UOM_CLASS	Yes	No	No	No
V_PACKSKU_QTY	Yes	No	No	No
WEEK_DATA	Yes	No	No	No
WH	Yes	No	No	No
JOB_AUDIT_LOGS	No	Yes	No	No
JOB_AUDIT_PAYLOAD	No	Yes	No	No

Cancel Customer Order Fulfillment Service

Functional Area

Inventory

Business Overview

This service is used to process Customer Order Fulfillment cancellation requests from an order management system (OMS). Merchandising supports three integration methods for processing Customer Order Fulfillment messages from OMS - either through RIB, SOAP service or REST service. At implementation time, you should decide on either RIB or web service for the integration method, but not both. However, if you choose web service, you can choose to call the SOAP service or the REST service or do a load share by splitting service calls between REST and SOAP calls. The same core logic is used to validate and persist customer orders in all three methods. In a web service implementation, the REST service is used to **cancel** a customer order in Merchandising.

- Success response: **200**
- Error response: **400**

This service:

- Accepts a collection of fulfillment orders as input. If one order fails, the entire service call fails and no orders will be created.
- Returns `Failure` status as part of the response object in the web service call if customer orders are not cancelled due to validation errors.

In a web service implementation, confirmation messages will be sent in a collection as part of the response object. This is the web service version of the same logic supported in the RIB version of the API. See [Customer Order Fulfillment Subscription API in RIB Subscription Designs](#) for more information.

Service Type

Put

ReST URL

/inventory/cancelFulfillOrders

Input Parameters

Table 5-1 fulfillOrdColRef - Collection of customer fulfillment order cancellation request

Parameter Name	Required	Data Type	Description
fulfillOrderReferences	Yes	Collection of Record	References a collection of cancellation records

Table 5-2 fulfillOrderReferences – Customer fulfillment order Reference

Parameter Name	Required	Data Type	Description
customerOrderNo	Yes	String	Holds the master customer order number from OMS.
fulfillOrderNo	Yes	String	Unique number from OMS related to the fulfillment details. One or more fulfillment orders could relate back to a single customer order in OMS.
sourceLocType	No	String	This would be either 'SU' for supplier, 'ST' for store, or 'WH' for warehouse. This would only be populated for vendor, warehouse or multi-site fulfillment orders.
sourceLocId	No	Number	Indicates the supplier, store or warehouse number associated with sourcing the customer order. Only populated for vendor, warehouse or multi-site fulfillment orders.
fulfillLocType	No	String	This would be either 'S' (for physical store) or 'V' (for virtual store).
fulfillLocId	Yes	Number	Indicates the store or warehouse number associated with fulfilling the customer order. For this release, this should always be populated with a virtual or physical store number.
orderDetails	Yes	Collection of Record	Indicates the customer order fulfillment cancellation detail.

Table 5-3 orderDetails – Customer fulfillment order details

Parameter Name	Required	Data Type	Description
item	Yes	String	Indicates the item ordered by the customer.
referenceItem	No	String	Indicates the reference item ordered by the customer. Used only if a specific UPC is ordered. This is supported for vendor drop-ships orders only as we don't support transfer requests for a specific reference item.
cancelQtySuom	Yes	Number	Indicates the quantity that should be cancelled from the order in item's standard unit of measure.
standardUom	No	String	Indicates item's standard unit of measure.
transactionUom	No	String	Indicates the original transaction unit of measure the order is placed in.
itemLineNo	No	Number	Indicates the detail item line number on the order.

Sample Input Message

```
{
  "fulfillOrderReferences": [
    {
      "customerOrderNo": "123252156",
      "fulfillOrderNo": "1231213",
      "sourceLocType": "ST",
      "sourceLocId": 6000,
      "fulfillLocType": "S",
      "fulfillLocId": 1311,
      "orderDetails": [
        {
          "item": "101450060",
          "referenceItem": "",
          "cancelQtySuom": 3,
          "standardUom": "EA",
          "transactionUom": "EA",
          "itemLineNo": 1
        }
      ]
    }
  ]
}
```

OutputResponse Code: **200 – Success**

Table 5-4 invocationSuccess – Webservice invocation success response

Parameter Name	Required	Data Type	Description
message	Yes	String	Message indicating the call was successful

Example

```
{
  "message": "string"
}
```

Output

Response Code: **400 – Error**

Table 5-5 RetailRestValidationException – Exception Record

Parameter Name	Required	Data Type	Description
message	Yes	String	Error Message
errors	Yes	Collection of String	Details of the errors

Example

```
{
  "message": "string",
  "errors": [
    "Additional information 1",
    "Additional information 2"
  ]
}
```

Code Detail Service

This section describes the Code Detail Service.

Business Overview

Code Detail service allows user to retrieve code details for a selected code and code type.

Service Type

Get

ReST URL

CodeDetail/codeDetails?code={ }&codeType={ }

Input Parameters

Parameter Name	Required	Description	Valid values
Code	Yes	Code	NA
Code Type	Yes	Code Type	NA

Output

Table 5-6 RestCodeDetailRecRDO

Parameter Name	Data Type
codeDesc	String
requiredInd	String
codeSeq	BigDecimal
codeType	String
codeTypeDesc	String
code	String

JSON Structure

```
{
  "codeDesc": null,
  "requiredInd": null,
  "codeSeq": null,
  "codeType": null,
  "codeTypeDesc": null,
  "code": null,
  "links": [],
  "hyperMediaContent": {
    "linkRDO": []
  }
}
```

Table Impact

TABLE	SELECT	INSERT	UPDATE	DELETE
CODE_HEAD	Yes	No	No	No
CODE_DETAIL	Yes	No	No	No
JOB_AUDIT_LOGS	No	Yes	No	No
JOB_AUDIT_PAYLOAD	No	Yes	No	No

Create Customer Order Fulfillment Service

Functional Area

Inventory

Business Overview

This service is used to process Customer Order Fulfillment requests from an order management system (OMS). Merchandising supports three integration methods for processing Customer Order Fulfillment messages from OMS - either through RIB, SOAP service or REST service. At implementation time, you should decide on either RIB or web service as the integration method, but not both. However, if you choose web service, you can choose to call the SOAP service, or the REST service, or do a load share by splitting service calls between REST and SOAP calls. The same core logic is used to validate and persist customer orders in all three methods. In a web service implementation, this REST service is used to create a customer order in Merchandising.

- Success response: **200**
- Error response: **400**

This service:

- Accepts a collection of fulfillment orders as input. If one order fails, the entire service call fails and no orders will be created.
- Returns `Failure` status as part of the response object in the web service call if customer orders are not created due to validation errors.
- Returns `Success` status and a confirmation message as part of the response object of type:
 - `X` if customer orders are not created due to lack of inventory
 - `P` if customer orders are partially created due to insufficient inventory
 - `C` if customer orders are completely created, when sufficient inventory is available

In a web service implementation, confirmation messages will be sent in a collection as part of the response object.

This is the web service version of the same logic as is supported in the RIB version of the API. See [Customer Order Fulfillment Subscription API](#) in [RIB Subscription Designs](#) for more information.

Service Type

Post

ReST URL

`/inventory/createFulfillOrders`

Input Parameters

Table 5-7 fulfilOrdColDesc - Collection of Customer order fulfillment order records

Parameter Name	Required	Data Type	Description
customerOrderNo	Yes	String	Holds the master customer order number from OMS.
fulfillOrderNo	Yes	String	Unique number from OMS related to the fulfillment details. One or more fulfillment orders could relate back to a single customer order in OMS.
sourceLocType	No	String	This would be either 'SU' for supplier, 'ST' for store, or 'WH' for warehouse. This would only be populated for vendor, warehouse or multi-site fulfillment orders.
sourceLocId	No	Number	Indicates the supplier, store or warehouse number associated with sourcing the customer order. Only populated for vendor, warehouse or multi-site fulfillment orders.
fulfillLocType	No	String	This would be either 'S' (for physical store) or 'V' (for virtual store).
fulfillLocId	Yes	Number	Indicates the store or warehouse number associated with fulfilling the customer order. For this release, this should always be populated with a virtual or physical store number.
partialDeliveryInd	Yes	String	Indicates if the order can be picked and shipped partially ('N') or if it should be shipped only when complete ('Y').
deliveryType	No	String	Indicates the fulfillment method - ship to customer or store pickup. Expected values are 'S' (ship direct) and 'C' (customer pickup).
carrierCode	No	String	Indicates the carrier the order is to be shipped with, if specified on the order.
carrierServiceCode	No	String	Indicates the method that was selected for shipping by the customer placing the order (e. g. Standard Shipping, Overnight, etc.).
consumerDeliveryDate	Yes	Date	Indicates the desired date the delivery is required by the customer. This will be the GMT time. The format should be yyyy-mm-dd
consumerDeliveryTime	No	Date	Indicates the desired time the delivery is required by the customer. This will be the GMT time. The format should be yyyy-mm-ddThh:mi:ssZ
deliveryCharges	No	Number	Contains the delivery charges on drop ship. Mostly relevant for Brazil.
deliveryChargesCurr	No	String	Contains the currency of the delivery charges.

Table 5-7 (Cont.) fulfilOrdColDesc - Collection of Customer order fulfillment order records

Parameter Name	Required	Data Type	Description
comments	No	String	Any comments sent by OMS about the order.
customerDetail	No	Record	Reference object for customer information related to the order.
orderDetails	Yes	Collection of Record	References a collection of customer order fulfillment details.
orderPlacedStore	No	Number	Indicates the store number associated with the location that the customer order was placed. For on-line orders this will contain the virtual store number associated with the on-line store. For orders captured in a physical store this will contain the store number for the physical store.

Table 5-8 orderDetails – Customer fulfillment order details

Parameter Name	Required	Data Type	Description
item	Yes	String	Indicates the item ordered by the customer.
referenceItem	No	String	Indicates the reference item ordered by the customer. Used only if a specific UPC is ordered. This is supported for vendor drop-ships orders only as we don't support transfer requests for a specific reference item.
orderQtySuom	Yes	Number	Indicates the quantity of the item ordered by the customer in item's standard unit of measure.
standardUom	No	String	Indicates item's standard unit of measure.
transactionUom	No	String	Indicates the original transaction unit of measure the order is placed in.
substituteInd	Yes	String	Indicates if substitutes are allowed on a customer order. This will only be used by orders passed to SIM.
unitRetail	No	Number	Indicates the unit sales retail of item on the customer order. This will only be used by Brazil orders in case of Warehouse Fulfillment or vendor sourced POs shipped directly to the customer. It is needed for Sales NF generation.

Table 5-8 (Cont.) orderDetails – Customer fulfillment order details

Parameter Name	Required	Data Type	Description
retailCurrency	No	String	Indicates currency of unit retail and total retail. This will only be used by Brazil orders in case of Warehouse Fulfillment or vendor sourced POs shipped directly to the customer. It is needed for Sales NF generation.
comments	No	String	Used to indicate any special instructions for the item, such as services (monograms, engrave, etc).
itemLineNo	No	Number	Indicates the detail item line number on the order.

Table 5-9 customerDetail – Customer information record

Parameter Name	Required	Data Type	Description
customerNo	No	String	Indicates the number that uniquely identifies the customer in OMS.
deliverFirstName	No	String	Contains the first name for the delivery address on the order.
deliverPhoneticFirst	No	String	Contains the phonetic first name for the delivery address on the order.
deliverLastName	No	String	Contains the last name for the delivery address on the order.
deliverPhoneticLast	No	String	Contains the phonetic last name for the delivery address on the order.
deliverPreferredName	No	String	Contains the preferred name for the delivery address on the order.
deliverCompanyName	No	String	Contains the company name for the delivery address on the order.
deliverAddress1	No	String	First line of the delivery address for the customer.
deliverAddress2	No	String	Second line of the delivery address for the customer.
deliverAddress3	No	String	Third line of the delivery address for the customer.
deliverCounty	No	String	County portion of the delivery address.
deliverCity	No	String	City portion of the delivery address.
deliverState	No	String	State portion of the delivery address.
deliverCountryId	No	String	Country portion of the delivery address.
deliverPost	No	String	Postal code portion of the delivery address.
deliverJurisdiction	No	String	Identifies the jurisdiction code for the delivery country-state relationship.
deliverPhone	No	String	Contains the delivery phone number.

Table 5-9 (Cont.) customerDetail – Customer information record

Parameter Name	Required	Data Type	Description
deliverEmail	No	String	Contains the delivery email.
billFirstName	No	String	Contains the first name for the billing address on the order.
billPhoneticFirst	No	String	Contains the phonetic first name for the billing address on the order.
billLastName	No	String	Contains the last name for the billing address on the order.
billPhoneticLast	No	String	Contains the phonetic last name for the billing address on the order.
billPreferredName	No	String	Contains the preferred name for the billing address on the order.
billCompanyName	No	String	Contains the company name for the billing address on the order.
billAddress1	No	String	First line of the billing address for the customer.
billAddress2	No	String	Second line of the billing address for the customer.
billAddress3	No	String	Third line of the billing address for the customer.
billCounty	No	String	County portion of the billing address.
billCity	No	String	City portion of the billing address.
billState	No	String	State portion of the billing address.
billCountryId	No	String	Country portion of the billing address.
billPost	No	String	Postal code portion of the billing address.
billJurisdiction	No	String	Identifies the jurisdiction code for the billing country-state relationship.
billPhone	No	String	Contains the billing phone number.
billEmail	No	String	Contains the billing email.

Sample Input Message

```
{
  "fulfillOrders": [
    {
      "customerOrderNo": "12342132156",
      "fulfillOrderNo": "123123213",
      "sourceLocType": "ST",
      "sourceLocId": 6000,
      "fulfillLocType": "S",
      "fulfillLocId": 1311,
      "partialDeliveryInd": "N",
      "deliveryType": "S",
      "carrierCode": "30",
      "carrierServiceCode": "D",
    }
  ]
}
```

```
"consumerDeliveryDate": "2022-12-31",
"consumerDeliveryTime": "2022-12-31T23:59:59.000Z",
"deliveryCharges": 3,
"deliveryChargesCurrency": "USD",
"comments": "commentDesc",
"customerDetail": {
  "customerNo": "123456",
  "deliverFirstName": "First",
  "deliverPhoneticFirst": "First",
  "deliverLastName": "Last",
  "deliverPhoneticLast": "Last",
  "deliverPreferredName": "Preferred Name",
  "deliverCompanyName": "Company Name",
  "deliverAddress1": "101 First Street",
  "deliverAddress2": "201 Second Street",
  "deliverAddress3": "301 Third Street",
  "deliverCounty": "Some County",
  "deliverCity": "Minneapolis",
  "deliverState": "MN",
  "deliverCountryId": "US",
  "deliverPost": "55555-1234",
  "deliverJurisdiction": "",
  "deliverPhone": "1800800800",
  "deliverEmail": "someone@email.com",
  "billFirstName": "First",
  "billPhoneticFirst": "First",
  "billLastName": "Last",
  "billPhoneticLast": "Last",
  "billPreferredName": "billPreferredName",
  "billCompanyName": "billCompanyName",
  "billAddress1": "101 First Street",
  "billAddress2": "201 Second Street",
  "billAddress3": "301 Third Street",
  "billCounty": "Some County",
  "billCity": "Minneapolis",
  "billState": "MN",
  "billCountryId": "US",
  "billPost": "55555-1234",
  "billJurisdiction": "",
  "billPhone": "1800800800",
  "billEmail": "someone@email.com"
},
"orderDetails": [
  {
    "item": "101450060",
    "referenceItem": "",
    "orderQtySuom": 3,
    "standardUom": "EA",
    "transactionUom": "EA",
    "substituteInd": "N",
    "unitRetail": 12,
    "retailCurrency": "USD",
    "comments": "commentDesc",
    "itemLineNo": 1
  }
]
```

```

    ],
    "orderPlacedStore": 1311
  }
]
}

```

Output

Response Code: **200 – Success**

Table 5-10 fulfillOrdCfmCol - Customer fulfillment order confirmation collection

Parameter Name	Required	Data Type	Description
collectionSize	No	Number	Number of records in the collection
fulfillOrderConfirmations	No	Collection of Record	Array of fulfillment order confirmations response

Table 5-11 fulfillOrderConfirmations – Customer fulfillment order confirmation record

Parameter Name	Required	Data Type	Description
customerOrderNo	Yes	String	Holds the master customer order number from OMS.
fulfillOrderNo	Yes	String	Unique number from OMS related to the fulfillment details. One or more fulfillment orders could relate back to a single customer order in OMS.
confirmType	Yes	String	Contains the confirmation type. 'P' - Order Partially Created; 'X' - Order Could not be Created; 'C' - Order Completely Created.
confirmNo	No	Number	Contains the PO or Transfer number in RMS related to the fulfillment order line. Required if fulfillment status is 'P'.
orderDetails	No	Collection of Record	References a collection of customer order confirmation item details.
fulfillLocId	No	Number	Contains the fulfillment location for the order.
systemCode	No	String	The code identifying the system associated with the location.

Table 5-12 orderDetails - Customer order confirmation item details

Parameter Name	Required	Data Type	Description
item	Yes	String	Indicates the item ordered by the customer.

Table 5-12 (Cont.) orderDetails - Customer order confirmation item details

Parameter Name	Required	Data Type	Description
referenceItem	No	String	Indicates the reference item ordered by the customer. Used only if a specific UPC is ordered. This is supported for vendor drop-ships orders only as we don't support transfer requests for a specific reference item.
confirmQty	Yes	Number	Indicates the quantity of the item that can be sourced or fulfilled on the order.
confirmQtyUom	Yes	String	Indicates the unit of measure of the confirmation quantity.
itemLineNo	No	Number	Indicates the detail item line number on the order.

Success Output Example

```
{
  "collectionSize": 1,
  "fulfillOrderConfirmations": [
    {
      "customerOrderNo": "123456",
      "fulfillOrderNo": "123456",
      "confirmType": "P",
      "confirmNo": 9999999,
      "orderDetails": [
        {
          "item": "100100076",
          "referenceItem": "",
          "confirmQty": 3,
          "confirmQtyUom": "EA",
          "itemLineNo": 1
        }
      ],
      "fulfillLocId": 6000,
      "systemCode": "INV"
    }
  ]
}
```

OutputResponse Code: **400 – Error****Table 5-13 RetailRestValidationException – Exception Record**

Parameter Name	Required	Data Type	Description
message	Yes	String	Error Message

Table 5-13 (Cont.) RetailRestValidationException – Exception Record

Parameter Name	Required	Data Type	Description
errors	Yes	Collection of String	Details of the errors

Error Output Example

```
{
  "message": "string",
  "errors": [
    "Additional information 1",
    "Additional information 2"
  ]
}
```

Create Franchise Orders

Functional Area

Franchise Management

Business Overview

This service is used from an external source, usually an order management system, to create franchise orders in Merchandising. This service accepts a collection of franchise orders and will return success or failure through the service response object. The franchise order uploaded through this service will be created with an order type of EDI. A linked transfer, PO, or store order will be created for the approved franchise orders.

Franchise orders created through this service are systematically approved if the customer is set up for automatic approval, provided the customer has valid credit. If the order does not meet the criteria for auto-approval, Merchandising will create the franchise order in `Input` or `Require Credit Approval` status and will provide the reason for the approval failure in the response object. Franchise orders from customers that are not identified for automatic approval are uploaded into Merchandising in `Input` status. Such orders will need to be manually approved in Merchandising to be considered active.

The service allows the upload of one or more franchise orders in a single service call. Each request is treated as a single unit of work and if there are no validation errors or business validation errors, all the franchise orders in the input payload will be created and success will be returned. In case of a validation error for any record, all the orders in the message will be created and the service will return success through the service response object. In case of one or more validation errors, the service call will be rejected, and the error response will contain the details of the validation errors.

Item Validation

- Packs are allowed for warehouse-sourced or supplier-sourced franchise orders only. If the pack added is a buyer pack with an 'order as' type of eaches, then the pack is exploded into its components.
- Deposit-container items cannot be added to franchise orders. When the deposit content item is added to an order, the associated container item is added as well. Deposit item content and container items must have the same costing location.
- Transformable, orderable items can be added to an order; however, transformable, sellable items are not allowed, as these are not inventoried.
- There cannot be multiple detail record for the same source location/item/customer location combination within a franchise order.
- Multiple order detail lines with the same source location, item and customer location combination in the franchise order is not allowed.

Source Validations

- For supplier-sourced orders, the supplier must be an active supplier and allow direct-store deliveries.
- For supplier-sourced franchise orders, items must be `Active` at the franchise store to be included on the order.
- The supplier must belong to the same organizational unit as the costing location.
- For warehouse-sourced or company-store-sourced order, the items must not be in `Delete` status at the source location and must be in either `Active` or `Discontinued` status at the franchise store.
- If the source for a franchise order is not specified, then the costing location for the item and franchise store will be used as the source location.
- If the source location is a physical warehouse, then the service will use distribution rules to determine from which virtual warehouse the inventory will be pulled, similar to distribution rules used for non-franchise transfers.

Customer Location Validation

- If an item/franchise store relationship does not exist, the franchise store is automatically ranged as part of franchise order creation.
- For source location as a store, the customer store cannot be non-stockholding.
- When a franchise order is uploaded through this service, having a future cost record for the item/franchise store combination is not necessary. Whether or not the cost template relationship is defined for the item/franchise store, the order can still be approved based on the provided fixed cost. However, if the fixed cost is also not provided, then the order creation fails.

Inventory Validations

The requested quantity is validated against the available inventory at the location for the warehouse-sourced or store-sourced franchise orders. For the warehouse-sourced orders, if the requested quantity exceeds the available stock at the warehouse, then Merchandising checks whether the item is on the Store Order replenishment at the same sourcing warehouse for the franchise store. If so, then an order is placed for the sourcing warehouse from the supplier for the unavailable quantity and the sourcing for the franchise location is

handled through transfers created through Merchandising replenishment process. When the source location is a store, requested inventory must be available at the location.

Order Date Validations

- Multiple franchise orders with the same source location, item, customer location and need date combination are not allowed in Merchandising.
- For the supplier-sourced franchise orders, the not-after date must be greater than supplier lead time.
- For the warehouse-sourced franchise orders, if the franchise order has a need date that is less than the order lead days in the future, the franchise order is fulfilled with the available warehouse inventory. Any remaining is fulfilled through Store Order replenishment if the item is on Store Order replenishment at the same sourcing warehouse for the franchise store.
- For store-sourced franchise orders, the need date must be within order lead days.

Service Type

POST

ReST URL

`/services/private/inventory/createFranchiseOrder`

Input Payload Details

Table 5-14 Create - Object. See list of elements for detail

Element Name	Required	Data Type	Description
items	Yes	Collection of Object	Collection

Table 5-15 Items - Object. See list of elements for detail

Element Name	Required	Data Type	Description
customerId	Yes	Number (10)	This value should be a valid customer ID.
orderReferenceNo	Yes	String (20)	This is an external reference number provided by the franchisee for their tracking purposes.
currencyCode	Yes	String (3)	This field represents the currency of the order, which may or may not be different from the primary currency in the system. Valid values for this field are based on the currency codes held in the CURRENCIES table.

Table 5-15 (Cont.) Items - Object. See list of elements for detail

Element Name	Required	Data Type	Description
exchangeRate	No	Number (20,10)	This field represents the exchange rate between the primary currency and the franchise order currency. If this is not provided, it is defaulted based on the conversion type set at system level.
freight	No	Number (20,4)	This field represents any freight charges associated with the franchise order.
otherCharges	No	Number (20,4)	This field represents other miscellaneous charges associated with the franchise order.
defaultBillingLocation	No	Number (10)	A customer's location where the billing for the entire order is sent. If blank, each location is billed.
billToAddressType	No	String (2)	This field represents the address type for the default billing location. This field is defaulted to Invoice address.
comments	No	String (2000)	Free form comments associated with the franchise order.
details	Yes	Collection of Object	Details of the orders. At least one detail item is mandatory.

Table 5-16 Details - Object. See list of elements for detail

Element Name	Required	Data Type	Description
item	Yes	String (25)	This is the item on the Franchise Order. This should be an approved, inventory, orderable, and transaction-level item. The item should be ranged to both customer location and source location and must not be a consignment/concession item.
customerLocation	Yes	Number (10)	This field holds a valid franchise location number.
sourceLocationType	No	String (2)	Contains the source entity type from which the items will originate. Valid values are <i>ST</i> - Store, <i>WH</i> - Warehouse, <i>SU</i> - Supplier. If this field is populated, the source location should also be populated.
sourceLocation	No	Number (10)	Contains the location ID from where this item will be sourced from. The ID will correspond to a company store, physical or virtual warehouse, or supplier depending on the source type value.

Table 5-16 (Cont.) Details - Object. See list of elements for detail

Element Name	Required	Data Type	Description
requestedQuantity	Yes	Number (12,4)	This field represents the quantity of the item on this record being ordered. This value is always written in the standard UOM for the item.
unitOfPurchase	No	String (3)	This field contains the unit of purchase information. Must be the standard unit of measure or a valid pallet name/case name/inner name for the item/supplier.
needDate	Yes	date	This date represents the initial date by which the franchisee wants the item on this order. The date format should be YYYY-MM-DD. The need date can be business date (VDATE) or a date in future.
notAfterDate	Yes	date	This date represents the last date by which the franchisee will accept the item on this order. The date format should be YYYY-MM-DD. This date should be after the need date.
fixedCost	No	Number (20,4)	This is a user defined cost which will override the customer cost for the item on this order if populated. This should be a positive numeric value. This must be provided if there are no cost templates associated with the item.

Table 5-17 CreateError - Object. See list of elements for detail

Element Name	Required	Data Type	Description
customerId	Yes	Number (10)	Input Customer ID
orderReferenceNo	Yes	String (20)	Input order reference number
item	No	String (25)	Input item
customerLocation	No	Number (10)	Input franchise location number
sourceLocationType	No	String (2)	Input source location type
sourceLocation	Yes	Number (10)	Input source location ID
errors	Yes	Array of String	List of errors identified during business data processing of the request

Sample Input Message

```
{
  "items": [
    {
      "customerId": 1001,
      "orderReferenceNo": "1001-A",
      "currencyCode": "USD",
      "exchangeRate": 52.5,
      "freight": 23.5,
      "otherCharges": 2.58,
      "defaultBillingLocation": 100123,
      "billToAddressType": "01",
      "comments": "Franchise order 1001-A",
      "details": [
        {
          "item": "104300083",
          "customerLocation": 100123,
          "sourceLocationType": "ST",
          "sourceLocation": 909090,
          "requestedQuantity": 9000,
          "unitOfPurchase": "EA",
          "needDate": "2001-12-31",
          "notAfterDate": "2001-12-31",
          "fixedCost": 95
        }
      ]
    }
  ]
}
```

Response Code: 200 (Success)

Table 5-18 CreateResponse - Object. See list of elements for detail

Element Name	Required	Data Type	Description
items	No	Collection of Object	Collection

Table 5-19 CreateResponse.Items - Object. See list of elements for detail

Element Name	Required	Data Type	Description
franchiseOrder	Yes	Number (10)	This is the unique identifier for the franchise order that is generated by Merchandising.
customerId	Yes	Number (10)	Input Customer ID
orderReferenceNo	Yes	String (20)	Input order reference number
status	No	String (1)	This contains the status in which the Franchise order was created. This can be I - Input or A - Approved.

Table 5-19 (Cont.) CreateResponse.Items - Object. See list of elements for detail

Element Name	Required	Data Type	Description
autoApproveErrors	No	Array of String	This will be populated with the reason (like customer fails credit check) why the order could not be approved and ended up being created in <code>Input</code> status. Using the UI, the franchise order should be approved after fixing the issue.
transactions	No	Collection of Object	References a collection of transfers or purchase orders created.

Table 5-20 Transactions - Object. See list of elements for detail

Element Name	Required	Data Type	Description
customerLocation	Yes	Number (10)	This field holds a valid franchise location number.
sourceLocation	Yes	Number (10)	Contains the location ID this item will be sourced from. The ID will correspond to a company store, physical or virtual warehouse, or supplier depending on the source type value.
sourceLocationType	Yes	String (2)	Contains the source entity type from which the items will originate. Valid values are <code>ST</code> - Store, <code>WH</code> - Warehouse, <code>SU</code> - Supplier. If this field is populated, the source location should also be populated.
needDate	No	date	This date represents the initial date by which the franchisee wants the item on this order. The date format should be <code>YYYY-MM-DD</code> . The need date can be a business date (<code>VDATE</code>) or a date in future.
documentNo	No	Number (12)	This contains the purchase order or transfer number created.
documentType	No	String (1)	This contains the document type generated. This can be <code>P</code> - Purchase Order or <code>T</code> - Transfer.

Sample Response Message

```
{
  "items": [
    {
      "franchiseOrder": 30001,
      "customerId": 1001,
      "orderReferenceNo": "1001-A",
```

```

    "status": "A",
    "autoApproveErrors": [
      "null"
    ],
    "transactions": [
      {
        "customerLocation": 100123,
        "sourceLocation": 909090,
        "sourceLocationType": "ST",
        "needDate": "2001-12-31",
        "documentNo": 22145453,
        "documentType": "T"
      }
    ]
  }
]
}

```

Response Code: 400 (Error)

In case of error, the following standard error response is returned. The element `validationErrors` is present when input payload or input parameters do not match the schema definition for this service. The element `businessError` element is present if the payload passes schema validation but an exception is caught while processing the business logic.

Table 5-21 CreateError - Object. See list of elements for detail

Element Name	Required	Data Type	Description
customerId	Yes	Number (10)	Input Customer ID
orderReferenceNo	Yes	String (20)	Input order reference number
item	No	String (25)	Input item
customerLocation	No	Number (10)	Input franchise location number
sourceLocationType	No	String (2)	Input source location type
sourceLocation	Yes	Number (10)	Input source location ID
errors	Yes	Array of String	List of errors identified during business data processing of the request

Sample Error Message

```

{
  "status": "ERROR",
  "message": "Error found in validation of input payload",
  "validationErrors": [
    {
      "error": "must be one of Y, N",
      "field": "createRecord.arg0.approveInd",
      "inputValue": "X"
    }
  ],
}

```

```

    "businessError": [
      {
        "customerId": 1001,
        "orderReferenceNo": "1001-A",
        "item": "104300083",
        "customerLocation": 100123,
        "sourceLocationType": "ST",
        "sourceLocation": 909090,
        "errors": [
          "Customer Location passed in franchise order is not a valid
franchise store."
        ]
      }
    ]
  }
}

```

Create Inventory Transfer Services

This section describes the inventory transfer services.

Functional Area

Inventory Movement

Business Overview

The primary role of these services is to create transfers and send them to Merchandising.

Transfer Number

Business Overview

Retrieves the next transfer number from Merchandising.

Service Type

Get

ReST URL

/Transfer/transferId

Input Parameters

No input

Output

...RDO

Parameter Name	Data Type
transfer_no	Long

```
JSON Structure:
"{
  "links": [],
  "transfer_no": 100000029403,
  "hyperMediaContent": {
    "linkRDO": []
  }
}"
```

Table Impact

NA

Search Items

Business Overview

This service retrieves items applicable for inventory transfer. Item can be searched either by Item or VPN. To search the item, enter an item number, a partial item description, or a VPN in the search string.

- When search type is ITEM, the search string can be an item number, a partial item number, an item description, or partial item description. In this case, the query returns all items which match the item description or partial description, or which match the item number entered.
- When search type is VPN, the search string can be a VPN or partial VPN, the API should return all items with that VPN.

The items returned are constrained by the following criteria:

- Approved status.
- Transaction-level items.
- Inventory items.
- When From Location is sent as an input, then only the following items are returned:
 - With available inventory at the From Location.
 - Packs with Receive as Type as Each are filtered out when, from location is a virtual warehouse.
- If the System Option for DEPT_LEVEL_TRANSFERS is set as "Y" and a Department ID is sent as input, then only the input department items are returned.

Service Type

Get

ReST URL

/Transfer/item?

itemSearchType={itemSearchType}&searchString={searchString}&dept={dept}&fromLocation={fromLocation}&pageSize={pageSize}&pageNumber={pageNumber}

Input Parameters

Parameter Name	Required	Description	Valid values
itemSearchType	Yes	Search type item or VPN.	ITEM, VPN
searchString	Yes	Search string for items ID or Name.	NA
dept	No	Selected items' department ID.	NA
fromLocation	No	Selected from location ID.	NA
PageSize	No	Maximum number of items to retrieve per page.	NA
PageNumber	No	Result page to retrieve.	NA

Output

TsflItemSearchRDO

Parameter Name	Data Type
item	String
itemDesc	String
dept	BigDecimal
availQty	BigDecimal
averageCost	BigDecimal
unitRetail	BigDecimal
currencyCode	String
standardUnitOfMeasure	String
suppPackSize	BigDecimal
innerPackSize	BigDecimal
itemImageUrl	String

PagedResultsRDO

Parameter Name	Data Type
totalRecordCount	BigDecimal
Next Page URL	String
Previous Page URL	String

JSON Structure:

```

{
  "links": [
    {
      "href": "/Transfer/item?
itemSearchType=ITEM&searchString=Black&fromLocation=363640301&pageSize=1&pageNumber=3",
      "rel": "next",
      "type": "GET",
      "methodType": null
    },
    {

```



```

        ""href"": ""/Transfer/item?
itemSearchType=ITEM&searchString=Black&fromLocation=363640301&pageSize=1&pageNumber=1""
    ,
        ""rel"": ""prev"",
        ""type"": ""GET"",
        ""methodType"": null
    }
    ],
    ""totalRecordCount"": 51,
    ""results"": [
        {
            ""links"": [],
            ""item"": ""100001406"",
            ""itemDesc"": ""DIT Test 11:Black:Extra Small"",
            ""dept"": 1102,
            ""availQty"": 100,
            ""averageCost"": 5,
            ""unitRetail"": 7.26,
            ""currencyCode"": ""USD"",
            ""standardlUnitOfMeasure"": ""CKG"",
            ""suppPackSize"": 1,
            ""innerPackSize"": 1,
            ""itemImageUrl"": null,
            ""hyperMediaContent"": {
                ""linkRDO"": []
            }
        }
    ],
    ""hyperMediaContent"": {
        ""linkRDO"": [
            {
                ""href"": ""/Transfer/item?
itemSearchType=ITEM&searchString=Black&fromLocation=363640301&pageSize=1&pageNumber=3""
            ,
                ""rel"": ""next"",
                ""type"": ""GET"",
                ""methodType"": null
            },
            {
                ""href"": ""/Transfer/item?
itemSearchType=ITEM&searchString=Black&fromLocation=363640301&pageSize=1&pageNumber=1""
            ,
                ""rel"": ""prev"",
                ""type"": ""GET"",
                ""methodType"": null
            }
        ]
    }
}"""

```

Table Impact

TABLE	SELECT	INSERT	UPDATE	DELETE
ALLOC_HEADER	Yes	No	No	No
ALLOC_DETAIL	Yes	No	No	No
DAILY_PURGE	Yes	No	No	No
ITEM_LOC	Yes	No	No	No

TABLE	SELECT	INSERT	UPDATE	DELETE
ITEM_LOC_SOH	Yes	No	No	No
ITEM_IMAGE	Yes	No	No	No
ITEM_SUPPLIER	Yes	No	No	No
ITEM_SUPP_COUNTRY	Yes	No	No	No
ORDHEAD	Yes	No	No	No
STORE	Yes	No	No	No
V_ITEM_MASTER	Yes	No	No	No
WH	Yes	No	No	No
JOB_AUDIT_LOGS	No	Yes	No	No
JOB_AUDIT_PAYLOAD	No	Yes	No	No

Load Items

This section describes the load items.

Business Overview

Load items service allows the user to refresh item records information for already selected items.

Service Type

Get

ReST URL

/Transfer/item/load?items={items}&fromLocation={fromLocation}

Input Paramters

Parameter Name	Required	Description
items	Yes	Comma Separated values for selected items' ID.
fromLocation	No	Selected from location ID.

Output

TsfltemSearchRDO

Parameter Name	Data Type
item	String
itemDesc	String
dept	BigDecimal
availQty	BigDecimal
averageCost	BigDecimal
unitRetail	BigDecimal

Parameter Name	Data Type
currencyCode	String
standardUnitOfMeasure	String
suppPackSize	BigDecimal
innerPackSize	BigDecimal
itemImageUrl	String

JSON Structure:

```
"[
  {
    "links": [],
    "item": "100001887",
    "itemDesc": "DIT Test 12:Black:Medium",
    "dept": 1102,
    "availQty": 100,
    "averageCost": 5,
    "unitRetail": 7.26,
    "currencyCode": "USD",
    "standardUnitOfMeasure": "CKG",
    "suppPackSize": 1,
    "innerPackSize": 1,
    "itemImageUrl": null,
    "hyperMediaContent": {
      "linkRDO": []
    }
  }
]"
```

Table Impact

TABLE	SELECT	INSERT	UPDATE	DELETE
ALLOC_HEADER	Yes	No	No	No
ALLOC_DETAIL	Yes	No	No	No
DAILY_PURGE	Yes	No	No	No
ITEM_LOC	Yes	No	No	No
ITEM_LOC_SOH	Yes	No	No	No
ITEM_IMAGE	Yes	No	No	No
ITEM_SUPPLIER	Yes	No	No	No
ITEM_SUPP_COUNTRY	Yes	No	No	No
ORDHEAD	Yes	No	No	No
STORE	Yes	No	No	No
V_ITEM_MASTER	Yes	No	No	No
WH	Yes	No	No	No
JOB_AUDIT_LOGS	No	Yes	No	No
JOB_AUDIT_PAYLOAD	No	Yes	No	No

Search From Location

This section describes the Search From Location service.

Business Overview

This service retrieves locations applicable for inventory transfer. Location can be searched by either 'S'tore or 'W'arehouse. Then enter a location number, a partial location number, a location description, or a partial location description in the search string.

The locations returned are constrained by the following criteria:

- When search type is warehouse only virtual warehouses are returned.
- Only stockholding location.
- When search type is store then only open stores are returned.
- When items are sent as input then only locations with available inventory are returned.
- When To Location is sent as input then:
 - It cannot be the same as the To Location.
 - When transfer type is Manual Requisition, then only locations with the same Transfer Entity/Set of Books as the To Location are returned in the search results.
 - When the transfer type is Intercompany, then only locations with a different Transfer Entity/Set of Books to the To Location are returned in the search results.
 - Only locations in the same transfer zone are returned in the search results.

Service Type

Get

ReST URL

/Transfer/fromLocation?
locationType={locationType}&searchString={searchString}&tsfType={tsfType}&toLocation={toLocation}&items={items}&pageSize={pageSize}&pageNumber={pageNumber}

Input Parameters

Parameter Name	Required	Description	Valid values
LocationType	Yes	Location type Store or warehouse	S, W
SearchString	Yes	search string for locations Id or Name	NA
tsfType	Yes	Transfer type	IC, MR
toLocation	No	Selected to location ID	NA

Parameter Name	Required	Description	Valid values
items	No	Comma Separated values for selected items	NA
PageSize	No	Maximum number of locations to retrieve per page	NA
PageNumber	No	Result page to retrieve	NA

Output

TsfLocSearchResultRDO

Parameter Name	Data Type
location	BigDecimal
locType	String
locName	String
locCurrencyCode	String
entity	BigDecimal
entityDesc	String
tsfLocitemSearchRes	List<TsfLocitemSearchResRDO>

TsfLocitemSearchResRDO

Parameter Name	Data Type
item	String
availQty	BigDecimal
averageCost	BigDecimal
unitRetail	BigDecimal
currencyCode	String

PagedResultsRDO

Parameter Name	Data Type
totalRecordCount	BigDecimal
Next Page URL	String
Previous Page URL	String

JSON Structure:

```
{
  "links": [],
  "totalRecordCount": 1,
  "results": [
    {
      "links": [],
      "location": 5991,
      "locType": "S",

```

```

    ""locName"": ""DIT Company Stockholding Store"",
    ""locCurrencyCode"": ""USD"",
    ""entity"": 1000,
    ""entityDesc"": ""Regular Stores"",
    ""tsfLocitemSearchRes"": [
      {
        ""links"": [],
        ""item"": ""100054006"",
        ""availQty"": 100,
        ""averageCost"": 0,
        ""unitRetail"": 181.82,
        ""currencyCode"": ""USD"",
        ""hyperMediaContent"": {
          ""linkRDO"": []
        }
      },
      {
        ""links"": [],
        ""item"": ""100040051"",
        ""availQty"": 998,
        ""averageCost"": 1,
        ""unitRetail"": 1.54,
        ""currencyCode"": ""USD"",
        ""hyperMediaContent"": {
          ""linkRDO"": []
        }
      }
    ],
    ""hyperMediaContent"": {
      ""linkRDO"": []
    }
  },
  ""hyperMediaContent"": {
    ""linkRDO"": []
  }
}

```

Table Impact

TABLE	SELECT	INSERT	UPDATE	DELETE
ALLOC_HEADER	Yes	No	No	No
ALLOC_DETAIL	Yes	No	No	No
ITEM_LOC	Yes	No	No	No
ITEM_LOC_SOH	Yes	No	No	No
ITEM_MASTER	Yes	No	No	No
MV_LOC_SOB	Yes	No	No	No
ORDHEAD	Yes	No	No	No
ORG_UNIT	Yes	No	No	No
STORE	Yes	No	No	No
TRANSFER_LOC	Yes	No	No	No
TSF_ENTITY	Yes	No	No	No
V_STORE	Yes	No	No	No

TABLE	SELECT	INSERT	UPDATE	DELETE
V_TRANSFER_FROM_LOC	Yes	No	No	No
V_TRANSFER_TO_LOC	Yes	No	No	No
V_WH	Yes	No	No	No
JOB_AUDIT_LOGS	No	Yes	No	No
JOB_AUDIT_PAYLOAD	No	Yes	No	No

Search To Location

This section describes the Search To Location service.

Business Overview

This service retrieves locations applicable for inventory transfer. Location can be searched by either 'S'tore or 'W'arehouse. Then enter a location number, a partial location number, a location description, or a partial location description in the search string.

The locations returned are constrained by the following criteria:

- When search type is warehouse only virtual warehouses are returned.
- Internal finishers are filtered out.
- Only stockholding location.
- When search type is Store then only open stores are returned.
- When items are sent as input then only locations with available inventory are returned.
- When From Location is sent as input then:
 - To Location cannot be the same as the From Location.
 - When Transfer Type is set as a manual request, then only locations with the same Transfer Entity/Set of Books as the From Location are returned in the search results.
 - When the Transfer Type is Intercompany, then only locations with a different Transfer Entity/Set of Books to the From Location are returned in the search results.
 - Only locations in the same transfer zone are returned in the search results.

Service Type

Get

ReST URL

/Transfer/toLocation?

locationType={locationType}&searchString={searchString}&tsfType={tsfType}&fromLocation={fromLocation}&pageSize={pageSize}&pageNumber={pageNumber}")

Input Parameters

Parameter Name	Required	Description	Valid values
LocationType	Yes	Location type Store or warehouse	S, W

Parameter Name	Required	Description	Valid values
SearchString	Yes	search string for locations Id or Name	NA
tsfType	Yes	Transfer type	IC, MR
fromLocation	No	Selected from location ID	NA
PageSize	No	Maximum number of locations to retrieve per page	NA
PageNumber	No	Result page to retrieve	NA

Output

TsfLocSearchResultRDO

Parameter Name	Data Type
location	BigDecimal
locType	String
locName	String
locCurrencyCode	String
entity	BigDecimal
entityDesc	String
tsfLocitemSearchRes	List<TsfLocitemSearchResRDO>

TsfLocitemSearchResRDO

Parameter Name	Data Type
item	String
availQty	BigDecimal
averageCost	BigDecimal
unitRetail	BigDecimal
currencyCode	String

PagedResultsRDO

Parameter Name	Data Type
totalRecordCount	BigDecimal
Next Page URL	String
Previous Page URL	String

JSON Structure:

```
{
  "links": [],
  "totalRecordCount": 1,
  "results": [
    {
      "links": [],
      "location": 5991,

```



```

        "locType": "S",
        "locName": "DIT Company Stockholding Store",
        "locCurrencyCode": "USD",
        "entity": 1000,
        "entityDesc": "Regular Stores",
        "tsfLocitemSearchRes": [],
        "hyperMediaContent": {
            "linkRDO": []
        }
    },
    "hyperMediaContent": {
        "linkRDO": []
    }
}

```

Table Impact

TABLE	SELECT	INSERT	UPDATE	DELETE
ALLOC_HEADER	Yes	No	No	No
ALLOC_DETAIL	Yes	No	No	No
ITEM_LOC	Yes	No	No	No
ITEM_LOC_SOH	Yes	No	No	No
ITEM_MASTER	Yes	No	No	No
MV_LOC_SOB	Yes	No	No	No
ORDHEAD	Yes	No	No	No
ORG_UNIT	Yes	No	No	No
STORE	Yes	No	No	No
TRANSFER_LOC	Yes	No	No	No
TSF_ENTITY	Yes	No	No	No
V_STORE	Yes	No	No	No
V_TRANSFER_FROM_LOC	Yes	No	No	No
V_TRANSFER_TO_LOC	Yes	No	No	No
V_WH	Yes	No	No	No
JOB_AUDIT_LOGS	No	Yes	No	No
JOB_AUDIT_PAYLOAD	No	Yes	No	No

Load Locations

This section describes the Load Locations service.

Business Overview

Load locations Web service allows user to refresh selected locations records.

Service Type

Get

ReST URL

/Transfer/loadLocations?fromLocation={fromLocation}&toLocation={toLocation}

Input Parameters

Parameter Name	Required	Description
FromLocation	No	Selected from location ID.
ToLocation	No	Selected to location ID.

Output

TsfLocSearchResultRDO

Parameter Name	Data Type
location	BigDecimal
locType	String
locName	String
locCurrencyCode	String
entity	BigDecimal
entityDesc	String
tsfLocitemSearchRes	List<TsfLocitemSearchResRDO>

TsfLocitemSearchResRDO

Parameter Name	Data Type
item	String
availQty	BigDecimal
averageCost	BigDecimal
unitRetail	BigDecimal
currencyCode	String

JSON Structure:

```
[
  {
    "links": [],
    "location": 5991,
    "locType": "S",
    "locName": "DIT Company Stockholding Store",
    "locCurrencyCode": "USD",
    "entity": 1000,
    "entityDesc": "Regular Stores",
    "tsfLocitemSearchRes": [],
    "hyperMediaContent": {
      "linkRDO": []
    }
  }
],
```

```

    {
      "links": [],
      "location": 12310101,
      "locType": "W",
      "locName": "test",
      "locCurrencyCode": "USD",
      "entity": 1000,
      "entityDesc": "Regular Stores",
      "tsfLocitemSearchRes": [],
      "hyperMediaContent": {
        "linkRDO": []
      }
    }
  ]"

```

Table Impact

TABLE	SELECT	INSERT	UPDATE	DELETE
ALLOC_HEADER	Yes	No	No	No
ALLOC_DETAIL	Yes	No	No	No
ITEM_LOC	Yes	No	No	No
ITEM_LOC_SOH	Yes	No	No	No
ITEM_MASTER	Yes	No	No	No
MV_LOC_SOB	Yes	No	No	No
ORDHEAD	Yes	No	No	No
ORG_UNIT	Yes	No	No	No
STORE	Yes	No	No	No
TRANSFER_LOC	Yes	No	No	No
TSF_ENTITY	Yes	No	No	No
V_STORE	Yes	No	No	No
V_TRANSFER_FROM_LOC	Yes	No	No	No
V_TRANSFER_TO_LOC	Yes	No	No	No
V_WH	Yes	No	No	No
JOB_AUDIT_LOGS	No	Yes	No	No
JOB_AUDIT_PAYLOAD	No	Yes	No	No

Create Transfer

This section describes the Create Transfer service.

Business Overview

The Web service calls the existing Merchandising XTSE API directly with input parameters. For more information on Merchandising XTSE API, see Store Order Subscription API and Transfer Subscription API sections.

Service Type

Post

ReST URL

/Transfer

Input Parameters

Example json RDO input:

```
{
  "links" : [ ],
  "tsfdtlRDOs" : [ {
    "links" : [ ],
    "item" : null,
    "tsfQty" : null,
    "suppPackSize" : null,
    "invStatus" : null,
    "unitCost" : null,
    "hyperMediaContent" : {
      "linkRDO" : [ ]
    }
  } ],
  "tsfNo" : null,
  "fromLocType" : null,
  "fromLoc" : null,
  "toLocType" : null,
  "toLoc" : null,
  "deliveryDate" : null,
  "dept" : null,
  "routingCode" : null,
  "freightCode" : null,
  "tsfType" : null,
  "status" : null,
  "userId" : null,
  "commentDesc" : null,
  "contextType" : null,
  "contextValue" : null,
  "hyperMediaContent" : {
    "linkRDO" : [ ]
  }
}
```

Output

NA

Table Impact

For more information on the Merchandising XTSE API, see the Store Order Subscription API and Transfer Subscription API sections.

Create Purchase Order Services

This section describes the Create Purchase Order Services section.

Functional Area

Procurement

Business Overview

The primary role of this service is to create purchase orders and send them to Merchandising.

Order Number

This section describes the Order Number.

Business Overview

Retrieves the next order number from Merchandising.

Service Type

Get

ReST URL

/PurchaseOrders/order/id

Input Parameters

NA

Output

OrderNoRDO

Parameter Name	Data Type
order_no	Long

JSON Structure:

```
"{"  
  "links": [],  
  "order_no": 100000047120,  
  "hyperMediaContent": {  
    "linkRDO": []  
  }  
}"
```

Table Impact

NA

Terms

This section describes the valid terms.

Business Overview

Retrieves all valid terms; valid terms are enabled with flag set to Yes and within the start and end active date.

Service Type

Get

ReST URL

/PurchaseOrders/supplier/terms

Input Parameters

NA

Output

PoSupTermsRDO

Parameter Name	Data Type
terms	String
terms_code	String
terms_desc	String

JSON Structure:

```
{
  "links": [],
  "terms": "108",
  "terms_code": "108",
  "terms_desc": "02 001.00% 010 000",
  "rank": null,
  "hyperMediaContent": {
    "linkRDO": []
  }
},"
```

Table Impact

TABLE	SELECT	INSERT	UPDATE	DELETE
TERMS_HEAD	Yes	No	No	No
TERMS_DETAIL	Yes	No	No	No
JOB_AUDIT_LOGS	No	Yes	No	No
JOB_AUDIT_PAYLOAD	No	Yes	No	No

Search Supplier

This section describes the Search Supplier service.

Business Overview

Supplier search can be, by entering either full or partial supplier site ID (numeric) or by a full or partial supplier site description in the search string.

Returned suppliers are constrained by the following criteria:

- Only active supplier sites are returned.
- When items are sent as input, then only supplier sites that are common amongst the items are returned.
- When locations are sent as input, then only suppliers that are valid for the Org Units associated with the input locations are returned.

Service Type

Get

ReST URL

```
/PurchaseOrders/supplier?
supplierSearchString={supplierSearchString}&locations={locations}&items={items}&pageSize
={pageSize}&pageNumber={pageNumber}
```

Input Parameters

Parameter Name	Required	Description
SupplierSearchString	Yes	Search string for Supplier's ID or Name.
Item	No	Comma Separated values for items.
Locations	No	Comma Separated values for locations.
PageSize	No	Maximum number of suppliers to retrieve per page.
PageNumber	No	Result page to retrieve.

Output

PoSupSearchResultRDO

Parameter Name	Data Type
supplier	BigDecimal
supplierName	String
supplierCurrency	String
terms	String
defaultItemLeadTime	BigDecimal
supplierSearchItemRDO	List<PoSupItemResultRDO>
supplierSearchItemLocRDO	List<PoSupItemLocResultRDO>

PoSupItemResultRDO

Parameter Name	Data Type
item	String
originCountryId	String
leadTime	BigDecimal

PoSupItemLocResultRDO

Parameter Name	Data Type
item	String
location	BigDecimal
pickupLeadTime	BigDecimal

PagedResultsRDO

Parameter Name	Data Type
totalRecordCount	BigDecimal
Next Page URL	String
Previous Page URL	String

JSON Structure:

```
{
  "links": [],
  "totalRecordCount": 1,
  "results": [
    {
      "links": [],
      "supplier": 2200,
      "supplierName": "Our Supplier",
      "supplierCurrency": "USD",
      "terms": "04",
      "defaultItemLeadTime": 2,
      "supplierSearchItemRDO": [
        {
          "links": [],
          "item": "100001887",
          "originCountryId": "US",
          "leadTime": 2,
          "hyperMediaContent": {
            "linkRDO": []
          }
        }
      ],
      "supplierSearchItemLocRDO": [
        {
          "links": [],
          "item": "100001887",
          "location": 363640301,
          "pickupLeadTime": null,
          "hyperMediaContent": {
            "linkRDO": []
          }
        }
      ],
      "hyperMediaContent": {
        "linkRDO": []
      }
    }
  ],
  "hyperMediaContent": {
    "linkRDO": []
  }
}
```


Table Impact

TABLE	SELECT	INSERT	UPDATE	DELETE
ITEM_SUPPLIER	Yes	No	No	No
ITEM_SUPP_COUNTRY	Yes	No	No	No
ITEM_SUPP_COUNTRY_LOC	Yes	No	No	No
STORE	Yes	No	No	No
SUPS	Yes	No	No	No
V_SUPS	Yes	No	No	No
WH	Yes	No	No	No
JOB_AUDIT_LOGS	No	Yes	No	No
JOB_AUDIT_PAYLOAD	No	Yes	No	No

Load Supplier

This section describes the load supplier service.

Business Overview

Loading supplier Web service allows a user to refresh the selected supplier records.

Service Type

Get

ReST URL

/PurchaseOrders/supplier/load?suppliers={suppliers}&locations={locations}&items={items}

Input Parameters

Parameter Name	Required	Description
Supplier	Yes	Supplier's ID.
Item	No	Comma Separated values for items.
Locations	No	Comma Separated values for locations.

Output

PoSupSearchResultRDO

Parameter Name	Data Type
supplier	BigDecimal
supplierName	String
supplierCurrency	String
terms	String

Parameter Name	Data Type
defaultItemLeadTime	BigDecimal
supplierSearchItemRDO	List<PoSupItemResultRDO>
supplierSearchItemLocRDO	List<PoSupItemLocResultRDO>

PoSupItemResultRDO

Parameter Name	Data Type
item	String
originCountryId	String
leadTime	BigDecimal

PoSupItemLocResultRDO

Parameter Name	Data Type
item	String
location	BigDecimal
pickupLeadTime	BigDecimal

JSON Structure:

```

{
  "links": [],
  "totalRecordCount": 1,
  "results": [
    {
      "links": [],
      "supplier": 2200,
      "supplierName": "Our Supplier",
      "supplierCurrency": "USD",
      "terms": "04",
      "defaultItemLeadTime": 2,
      "supplierSearchItemRDO": [
        {
          "links": [],
          "item": "100001887",
          "originCountryId": "US",
          "leadTime": 2,
          "hyperMediaContent": {
            "linkRDO": []
          }
        }
      ],
      "supplierSearchItemLocRDO": [
        {
          "links": [],
          "item": "100001887",
          "location": 363640301,
          "pickupLeadTime": null,
          "hyperMediaContent": {
            "linkRDO": []
          }
        }
      ]
    }
  ]
}

```

```

    ],
    "hyperMediaContent": {
      "linkRDO": []
    }
  }
],
"hyperMediaContent": {
  "linkRDO": []
}
}"

```

Table Impact

TABLE	SELECT	INSERT	UPDATE	DELETE
ITEM_SUPPLIER	Yes	No	No	No
ITEM_SUPP_COUNTRY	Yes	No	No	No
ITEM_SUPP_COUNTRY_LOC	Yes	No	No	No
STORE	Yes	No	No	No
SUPS	Yes	No	No	No
V_SUPS	Yes	No	No	No
WH	Yes	No	No	No
JOB_AUDIT_LOGS	No	Yes	No	No
JOB_AUDIT_PAYLOAD	No	Yes	No	No

Search Items

This section describes the Search Items service.

Business Overview

This service retrieves items applicable for Purchase Order. Item can be searched by either Item or VPN. Enter an item number, a partial item description, or a VPN in the search string.

1. When search type is ITEM, the search string can be an item number, a partial item number, an item description, or partial item description
2. When search type is VPN, the search string can be a VPN or partial VPN.

The items returned are constrained by the following criteria:

- Approved status.
- Transaction-level items.
- Orderable items.
- Pack items with Order Type as Each are filtered out.
- Only items belonging to Normal Merchandise Purchase Type as Department are returned.
- When a supplier is sent as input then:
 - Only items supplied by the input supplier are returned.
 - The item information is based on the Item/Supplier/Primary Origin Country.

- When supplier is not sent as input, then item information is based on the primary supplier and primary origin country.
- If the system_options.dept_level_orders is set to "Y" and the Department ID is sent as input, then only the input department items are returned.
- Items set for deletion are filtered out.

Service Type

Get

ReST URL

```
/PurchaseOrders/item?
itemSearchType={itemSearchType}&searchString={searchString}&dept={dept}&supplier={supplier}&locations={locations}&pageSize={pageSize}&pageNumber={pageNumber}
```

Input Parameters

Parameter Name	Required	Description	Valid values
itemSearchType	Yes	Search Type item or VPN.	ITEM, VPN
searchString	Yes	Search string for items Id or Name.	NA
dept	No	Selected items' department ID.	NA
supplier	No	Selected Supplier ID.	NA
Locations	No	Comma Separated values for selected locations' ID.	NA
PageSize	No	Maximum number of items to retrieve per page.	NA
PageNumber	No	Result page to retrieve.	NA

Output

PoltemSearchResultRDO

Parameter Name	Data Type
item	String
itemDesc	String
supplier	BigDecimal
originCountry	String
suppPackSize	BigDecimal
unitCost	BigDecimal
supplierCurrency	String
baseUnitRetail	BigDecimal
retailCurrency	String
baseRetailUnitOfMeasure	String
itemImageUrl	String

Parameter Name	Data Type
dept	BigDecimal
itemSearchLocRDO	List<PoltemSearchRstLocRDO>

PoltemSearchRstLocRDO

Parameter Name	Data Type
location	BigDecimal
locationType	String
unitRetail	BigDecimal
retailCurrency	String
unitRetailUnitOfMeasure	String
itemLocStatus	String

PagedResultsRDO

Parameter Name	Data Type
totalRecordCount	BigDecimal
Next Page URL	String
Previous Page URL	String

JSON Structure:

```
{
  "links": [],
  "totalRecordCount": 1,
  "results": [
    {
      "links": [],
      "item": "100001887",
      "itemDesc": "DIT Test 12:Black:Medium",
      "supplier": 2200,
      "originCountry": "US",
      "suppPackSize": 1,
      "unitCost": 5,
      "supplierCurrency": "USD",
      "baseUnitRetail": 7.26,
      "retailCurrency": "USD",
      "baseRetailUnitOfMeasure": "EA",
      "itemImageUrl": null,
      "dept": 1102,
      "itemSearchLocRDO": [
        {
          "links": [],
          "location": 363640301,
          "locationType": "W",
          "unitRetail": 7.26,
          "retailCurrency": "USD",
          "unitRetailUnitOfMeasure": "CKG",
          "itemLocStatus": "A",
          "hyperMediaContent": {
            "linkRDO": []
          }
        }
      ]
    }
  ]
}
```

```

        }
      },
      "hyperMediaContent": {
        "linkRDO": []
      }
    },
    "hyperMediaContent": {
      "linkRDO": []
    }
  }
}

```

Table Impact

TABLE	SELECT	INSERT	UPDATE	DELETE
DAILY_PURGE	Yes	No	No	No
DEPS	Yes	No	No	No
ITEM_LOC	Yes	No	No	No
ITEM_IMAGE	Yes	No	No	No
ITEM_MASTER	Yes	No	No	No
ITEM_SUPPLIER	Yes	No	No	No
ITEM_SUPP_COUNTRY	Yes	No	No	No
MV_CURRENCY_CONVERSION_RATES	Yes	No	No	No
RPM_MERCH_RETAIL_DEF_EXPL	Yes	No	No	No
RPM_ZONE	Yes	No	No	No
V_ITEM_MASTER	Yes	No	No	No
V_PACKSKU_QTY	Yes	No	No	No
V_SUPS	Yes	No	No	No
WH	Yes	No	No	No
JOB_AUDIT_LOGS	No	Yes	No	No
JOB_AUDIT_PAYLOAD	No	Yes	No	No

Load Items

This section describes the load items.

Business Overview

The primary use of loading items Web service is to refresh already selected PO items records.

Service Type

Get

ReST URL

/PurchaseOrders/item/load?item=item&supplier={supplier}&locations={locations}

Input Parameters

Parameter Name	Required	Description
Items	Yes	Comma Separated values for selected items' ID.
Supplier	No	Selected Supplier ID.
Locations	No	Comma Separated values for selected locations' ID.

Output

PoltemSearchResultRDO

Parameter Name	Data Type
item	String
itemDesc	String
supplier	BigDecimal
originCountry	String
suppPackSize	BigDecimal
unitCost	BigDecimal
supplierCurrency	String
baseUnitRetail	BigDecimal
retailCurrency	String
baseRetailUnitOfMeasure	String
itemImageUrl	String
dept	BigDecimal
itemSearchLocRDO	List<PoltemSearchRstLocRDO>

PoltemSearchRstLocRDO

Parameter Name	Data Type
location	BigDecimal
locationType	String
unitRetail	BigDecimal
retailCurrency	String
unitRetailUnitOfMeasure	String
itemLocStatus	String

PagedResultsRDO

Parameter Name	Data Type
totalRecordCount	BigDecimal
Next Page URL	String
Previous Page URL	String

JSON Structure:

```
{
  "links": [],
  "totalRecordCount": 1,
  "results": [
    {
      "links": [],
      "item": "100001887",
      "itemDesc": "DIT Test 12:Black:Medium",
      "supplier": 2200,
      "originCountry": "US",
      "suppPackSize": 1,
      "unitCost": 5,
      "supplierCurrency": "USD",
      "baseUnitRetail": 7.26,
      "retailCurrency": "USD",
      "baseRetailUnitOfMeasure": "EA",
      "itemImageUrl": null,
      "dept": 1102,
      "itemSearchLocRDO": [
        {
          "links": [],
          "location": 363640301,
          "locationType": "W",
          "unitRetail": 7.26,
          "retailCurrency": "USD",
          "unitRetailUnitOfMeasure": "CKG",
          "itemLocStatus": "A",
          "hyperMediaContent": {
            "linkRDO": []
          }
        }
      ],
      "hyperMediaContent": {
        "linkRDO": []
      }
    }
  ],
  "hyperMediaContent": {
    "linkRDO": []
  }
}
```

Table Impact

TABLE	SELECT	INSERT	UPDATE	DELETE
DAILY_PURGE	Yes	No	No	No
DEPS	Yes	No	No	No
ITEM_LOC	Yes	No	No	No
ITEM_IMAGE	Yes	No	No	No
ITEM_MASTER	Yes	No	No	No
ITEM_SUPPLIER	Yes	No	No	No
ITEM_SUPP_COUNTRY	Yes	No	No	No
MV_CURRENCY_CONVERSION_RATES	Yes	No	No	No
RPM_MERCH_RETAIL_DEF_EXPL	Yes	No	No	No

TABLE	SELECT	INSERT	UPDATE	DELETE
RPM_ZONE	Yes	No	No	No
V_ITEM_MASTER	Yes	No	No	No
V_PACKSKU_QTY	Yes	No	No	No
V_SUPS	Yes	No	No	No
WH	Yes	No	No	No
JOB_AUDIT_LOGS	No	Yes	No	No
JOB_AUDIT_PAYLOAD	No	Yes	No	No

Search Locations

This section describes the Search Location service.

Business Overview

The Web service enables location search applicable for PO. Location can be searched by either 'S'tore or 'W'arehouse. Enter a location number, a partial location number, a location description, or a partial location description in the search string.

The locations returned are constrained by the following criteria:

- Only stockholding locations are returned.
- When search type is Warehouse then:
 - Only virtual warehouses are returned.
 - Internal finishers are filtered out.
- When search type is store then only the following stores are returned:
 - Company stores.
 - Open stores.
- When system_options.org_unit_ind is set as 'Y' then:
 - When supplier is sent as input then only locations with same org_unit_id are returned.
 - When Org Unit ID is sent as input then only locations with same org_unit_id are returned.

Service Type

Get

ReST URL

```
/PurchaseOrders/location?
locationType={locationType}&searchString={searchString}&supplier={supplier}&orgUnitId={or
gUnitId}&pageSize={pageSize}&pageNumber={pageNumber}
```

Input Parameters

Parameter Name	Required	Description	Valid values
LocationType	Yes	Location type Store or warehouse.	S, W
SearchString	Yes	Search string for locations Id or Name.	NA
Supplier	No	Selected Supplier ID.	NA
OrgUnitId	No	Selected locations' Org unit ID.	NA
PageSize	No	Maximum number of locations to retrieve per page.	NA
PageNumber	No	Result page to retrieve.	NA

Output

PoLocSearchResultRDO

Parameter Name	Data Type
location	BigDecimal
locType	String
locName	String
locationCurrency	String
orgUnitId	BigDecimal

PagedResultsRDO

Parameter Name	Data Type
totalRecordCount	BigDecimal
Next Page URL	String
Previous Page URL	String

JSON Structure:

```
{
  "links": [
    {
      "href": "/PurchaseOrders/location?searchString=e&pageSize=1&pageNumber=2",
      "rel": "next",
      "type": "GET",
      "methodType": null
    }
  ],
  "totalRecordCount": 100,
  "results": [
    {
      "links": [],
      "location": 292919862,
```

```

        "locType": "S",
        "locName": "ALLOC_FD_Store_1_292919862",
        "locationCurrency": "USD",
        "orgUnitId": 111111111,
        "hyperMediaContent": {
            "linkRDO": []
        }
    },
    "hyperMediaContent": {
        "linkRDO": [
            {
                "href": "/PurchaseOrders/location?searchString=e&pageSize=1&pageNumber=2",
                "rel": "next",
                "type": "GET",
                "methodType": null
            }
        ]
    }
}
}
}
}
"

```

Table Impact

TABLE	SELECT	INSERT	UPDATE	DELETE
PARTNER_ORG_UNIT	Yes	No	No	No
V_STORE	Yes	No	No	No
V_WH	Yes	No	No	No
WH	Yes	No	No	No
JOB_AUDIT_LOGS	No	Yes	No	No
JOB_AUDIT_PAYLOAD	No	Yes	No	No

Load Locations

This section describes the Load Locations service.

Business Overview

This Web service allows the user to refresh already selected PO locations records.

Service Type

Get

ReST URL

`/PurchaseOrders/location/load?locations={locations}&supplier={supplier}`

Input Parameters

Parameter Name	Required	Description
Locations	Yes	Comma Separated values for selected locations' ID.

Parameter Name	Required	Description
Supplier	No	Selected Supplier ID.

Output

PoLocSearchResultRDO

Parameter Name	Data Type
location	BigDecimal
locType	String
locName	String
locationCurrency	String
orgUnitId	BigDecimal

JSON Structure:

```
"
{
  ""links"": [],
  ""location"": 292919862,
  ""locType"": ""S"",
  ""locName"": ""ALLOC_FD_Store_1_292919862"",
  ""locationCurrency"": ""USD"",
  ""orgUnitId"": 111111111,
  ""hyperMediaContent"": {
    ""linkRDO"": []
  }
}
```

Table Impact

TABLE	SELECT	INSERT	UPDATE	DELETE
PARTNER_ORG_UNIT	Yes	No	No	No
V_STORE	Yes	No	No	No
V_WH	Yes	No	No	No
WH	Yes	No	No	No
JOB_AUDIT_LOGS	No	Yes	No	No
JOB_AUDIT_PAYLOAD	No	Yes	No	No

Create Purchase Order

This section describes the Create Purchase Order service.

Business Overview

This Web service calls the existing Merchandising XOrder API directly with input parameters. For more information on Merchandising XOrder API, see the sections addressing both the Store Order Subscription API and the PO Subscription API.

Service Type

Post

ReST URL

/PurchaseOrders

Input Parameters

Example json RDO input:

```
{
  "links" : [ ],
  "itemRDOs" : [ {
    "links" : [ ],
    "item" : null,
    "location" : null,
    "unitCost" : null,
    "referenceItem" : null,
    "originCountryId" : null,
    "suppPackSize" : null,
    "qtyOrdered" : null,
    "locationType" : null,
    "cancelInd" : null,
    "reInstateInd" : null,
    "hyperMediaContent" : {
      "linkRDO" : [ ]
    }
  } ],
  "orderNo" : null,
  "supplier" : null,
  "currencyCode" : null,
  "terms" : null,
  "notBeforeDate" : null,
  "notAfterDate" : null,
  "status" : "A",
  "writtenDate" : null,
  "origInd" : null,
  "user_id" : null,
  "dept" : null,
  "exchangeRate" : null,
  "includeOnOrdInd" : null,
  "ediPoInd" : null,
  "preMarkInd" : null,
  "comment" : null,
  "otbEowDate" : null,
  "hyperMediaContent" : {
    "linkRDO" : [ ]
  }
}
```

Output

NA

Table Impact

For more information on Merchandising XOrder API, see the Store Order Subscription API and the PO Subscription API sections.

Currency Rates Service

This section describes the Currency Rate service.

Business Overview

This service is used to retrieve all currencies and currency conversion rates. The conversion rate is the value used to convert to the primary currency.

Service Type

Get

ReST URL

CurrencyRates/currencyRates

Input Parameters

NA

Output

RestCurrencyRatesRecRDO

Parameter Name	Data Type
exchangeRate	BigDecimal
effectiveDate	Timestamp
currencyCode	String
exchangeType	String

JSON Structure:

```
[
  {
    "exchangeRate": null,
    "effectiveDate": null,
    "currencyCode": null,
    "exchangeType": null,
    "links": [],
    "hyperMediaContent": {
      "linkRDO": []
    }
  }
]
```

Table Impact

TABLE	SELECT	INSERT	UPDATE	DELETE
CURRENCY_RATES	Yes	No	No	No
JOB_AUDIT_LOGS	No	Yes	No	No
JOB_AUDIT_PAYLOAD	No	Yes	No	No

Data Privacy Access Service

This section describes the Data Privacy Access service for Merchandising.

Business Overview

This query service provides access to data stored in Merchandising that contain personally identifiable information.

Service Type

GET

ReSTURL

`https://<host:port>/RetailAppsDataPrivServicesRESTApp/rest/privatedata/getPersonalInfo?customer_id={entityName}::{entityType}::{entityId}::{fullName}:: {phone}::{email}`

Accept

- application/json
- application/xml

Query Parameters

- customer_id (required): The customer ID string containing the parameters to be used in looking up data. The format of this string is as follows:
 - entity name}::{entity type}::{entity id}::{full name}::{phone}::{email}

Path Parameters

Parameter	Description
Entity Name	The query group type for which data is to be retrieved. The available group types for Merchandising are: <ul style="list-style-type: none"> • BUYER (Buyer) • MERCHANT (Merchant) • STORE (Store) • WAREHOUSE (Warehouse) • SUPPLIER (Supplier) • PARTNER (Partner) • OUTLOC (Outside Location) • ORDER CUSTOMER (Merchandising Customer)
Entity Type	Used if the entity name is PARTNER or OUTLOC. The value here should indicate the type of partner or outside location being queried. Valid values for this input can be found on the Codes table for each type: <ul style="list-style-type: none"> • Partner Types = PTNR • Outside Location Types = LOCT
Entity ID	The ID of the entity being queried. For example, the supplier ID.
Full Name	The full name to be searched for the selected entity.
Phone	The phone number to be searched for the selected entity.
Email	The email to be searched for the selected entity.

Default Response

The response will return all instances of the data being searched that occur in the requested entity. For example, if the entity requested was BUYER, all instances where the buyer, name, and phone match the data sent will be returned. If any of these parameters are not sent (e.g. buyer), then it will not be used as part of the search. The following data is included in the response:

Parameter	Description
Entity Name	Valid values are: <ul style="list-style-type: none"> • BUYER (Buyer) • MERCHANT (Merchant) • STORE (Store) • WAREHOUSE (Warehouse) • SUPPLIER (Supplier) • PARTNER (Partner) • OUTLOC (Outside Location) • ORDER CUSTOMER (Merchandising Customer)
Entity Type	If the entity name is PARTNER or OUTLOC, the value here indicates the type of partner or outside location being queried. Valid values for this input can be found on the Codes table for each type: <ul style="list-style-type: none"> • Partner Types = PTNR • Outside Location Types = LOCT For other entity types, this will be null.
Entity ID	The ID of the entity where the data was found.
Full Name	The name associated with the entity.

Parameter	Description
Phone	The phone number associated with the entity.
Fax	The fax number associated with the entity.
Telex	The telex number associated with the entity.
Pager	The pager number associated with the entity.
Email	The email address associated with the entity.

Sample Response

```
{
  "Personal Information": {
    "list": [],
    "Get Personal Information": {
      "list": [
        [
          {
            "ENTITY_NAME": "BUYER",
            "ENTITY_TYPE": "null",
            "ENTITY_ID": "1002",
            "FULL_NAME": "Matt Wilsman",
            "PHONE": "6125251034",
            "FAX": "6125259800",
            "TELEX": "null",
            "PAGER": "null",
            "EMAIL": "null"
          }
        ]
      ]
    }
  }
}
```

Response Codes and Error Messages

- 200 - Success
- 400 - Bad Request - for the following situations:
 - Customer ID does not match the required format
 - Invalid input type
 - Missing customer ID
 - Invalid jsonFormat
- 500 - Internal Server Errors - for all other types of errors (for example, configuration errors, SQL errors, and so on)

Success Payloads

- When Accept=application/json, this API will return data in JSON format
- When Accept=application/xml, this API will return data formatted as an HTML page

Data Privacy Forget Service

This section describes the Data Privacy Forget service for Merchandising.

Business Overview

This service supports updating personal information stored in Merchandising. When the service is invoked with mask strings as inputs, it overwrites the fields with mask strings, which effectively removes the personal information from the system.

Service Type

DELETE

ReSTURL

```
https://<host:port>/RetailAppsDataPrivServicesRESTApp/rest/privatedata/  
updatePersonalInfo?customer_id={entityName}::{entityType}::{entityId}::{fullName}::  
{phone}::{fax}::{telex}::{pager}::{email}::{addr1}::{addr2}::{addr3}::{county}::{city}::  
{state}::{countryId}::{postalCode}
```

Accept

- application/json
- application/xml

Query Parameters

- customer_id (required): The customer ID string containing the parameters to be used in updating data. The format of this string is as follows:
 - {entityName}::{entityType}::{entityId}::{fullName}::{phone}::{fax}::{telex}::
{pager}::{email}::{addr1}::{addr2}::{addr3}::{county}::{city}::{state}::{countryId}::
{postalCode}

Path Parameters

Parameter	Description
Entity Name (required)	The group type for which data is to be updated. The available group types for Merchandising are: <ul style="list-style-type: none"> • BUYER (Buyer) • MERCHANT (Merchant) • STORE (Store) • WAREHOUSE (Warehouse) • SUPPLIER (Supplier) • PARTNER (Partner) • OUTLOC (Outside Location) • ORDER CUSTOMER (Merchandising Customer)

Parameter	Description
Entity Type	Required if the entity name is PARTNER or OUTLOC. The value here should indicate the type of partner or outside location. Valid values for this input can be found on the Codes table for each type: <ul style="list-style-type: none">Partner Types = PTNROutside Location Types = LOCT
Entity ID (required)	The ID of the entity to be updated. For example, the supplier ID.
Full Name	The value to update the full name with. If the value is null and this is a required field in the entity, 'XXXXX' will be used.
Phone	The value to update the phone number with. If the value is null and this is a required field in the entity, 'XXXXX' will be used.
Fax	The value to update the fax number with.
Telex	The value to update the telex number with.
Pager	The value to update the pager number with.
Email	The value to update the email address with.
Addr1	The value to update the address 1 with.
Addr2	The value to update the address 2 with.
Addr3	The value to update the address 3 with.
County	The value to update the county with.
City	The value to update the city with.
State	The value to update the state with.
Country	The value to update the country with.
Postal Code	The value to update the postal code with.

Default Response

This service only returns a response code to signify if the request is successful or not. If no record is updated, the service returns an error.

Response Codes and Error Messages

- 200 - Success
- 400 - Bad Request - for the following situations:
 - Customer ID does not match the required format
 - Invalid input type
 - Missing customer ID
 - Invalid jsonFormat
- 500 - Internal Server Errors - for all other types of errors (e.g. config errors, sql errors, etc).

Success Payloads

N/A

Diff Detail Service

This section describes the Diff Detail service.

Business Overview

Diff Detail service allows user to retrieve Diff description for a selected Diff Id.

Service Type

Get

ReST URL

DiffIds/diffIdDetail?diffId={diffId}

Input Parameters

Parameter Name	Required	Description
Diff_Id	Yes	Diff ID

Output

RestDiffIdsRecRDO

Parameter Name	Data Type
industrySubgroup	String
diffGroupDesc	String
diffType	String
diffDesc	String
industryCode	String
diffGroupId	String
diffTypeDesc	String

JSON Structure

```

{
  "industrySubgroup": null,
  "diffGroupDesc": null,
  "diffType": null,
  "diffDesc": null,
  "industryCode": null,
  "diffGroupId": null,
  "diffTypeDesc": null,
  "links": [],
  "hyperMediaContent": {
    "linkRDO": []
  }
}

```

Table Impact

TABLE	SELECT	INSERT	UPDATE	DELETE
DIFF_IDS	Yes	No	No	No
JOB_AUDIT_LOGS	No	Yes	No	No
JOB_AUDIT_PAYLOAD	No	Yes	No	No

EOW Run Signal Service

Business Overview

This service is used to determine whether the Vdate is an End of Week Date or Not. If Vdate is at EOW, it returns 'Y', else 'N'.

Service Type

Get

ReST URL

EowRunSignal/EowRunSignalDetail

Input Parameters

N/A

Output

RestEowRDO

Parameter Name	Data Type
eow	Char('Y' or 'N')

JSON Structure:

```
{
  "eow": "N",
  "links": [],
  "hyperMediaContent": {
    "linkRDO": []
  }
}
```

Table Impact

TABLE	SELECT	INSERT	UPDATE	DELETE
PERIOD	Yes	No	No	No
SYSTEM_VARIABLES	Yes	No	No	No

Half Data Budget Service

Business Overview

The primary role of this service is to modify half data budgets and send them to Merchandising.

Functional Area

Financials

Modify Half Data Budget

Business Overview

This service modifies half data budget by calling the SVCPROV_HDB package to load input data to the staging tables and then calling the core half data budget package to validate and insert data to the Merchandising tables.

Service Type

Post

Rest URL:

financials/HalfDataBudgetREST/modifyHdb

Input Parameters

SvcprovHdbdescRecRDO

Parameter Name	Data Type
dept	BigDecimal
halfNo	BigDecimal
locType	String
location	BigDecimal
setOfBooksId	BigDecimal
cumMarkonPct	BigDecimal
shrinkagePct	BigDecimal

Parameter Name	Data Type
markdownPct	BigDecimal

JSON Structure:

```
[
  {
    "dept": null,
    "halfNo": null,
    "locType": null,
    "location": null,
    "setOfBooksId": null,
    "cumMarkonPct": null,
    "shrinkagePct": null,
    "markdownPct": null
  }
]
```

Output

SvcprovHdbStatusRecRDO

Parameter Name	Data Type
statusMsg	String
hdbErrTbl	List< SvcprovFailHdbRecRDO>

SvcprovFailHdbRecRDO

Parameter Name	Data Type
dept	BigDecimal
halfNo	BigDecimal
locType	String
location	BigDecimal
setOfBooksId	BigDecimal
errorMsg	String

The output will contain the status of the request including validation errors, if any.

JSON Structure:

```
{
  "statusMsg": null,
  " hdbErrTbl ":
  [
    {
      "dept": null,
      "halfNo": null,
      "locType": null,
      "location": null,
      "setOfBooksId": null,
      "errorMsg": null,
      "links": [],
      "hyperMediaContent": {
        "linkRDO": []
      }
    }
  ]
}
```

```

    }
  }
],
"links": [],
"hyperMediaContent": {
  "linkRDO": []
}
}

```

Table Impact

TABLE	SELECT	INSERT	UPDATE	DELETE
HALF_DATA_BUGET	Yes	Yes	Yes	No
SVC_PROCESS_TRACKER	Yes	Yes	Yes	No
SVC_ADMIN_UPLD_ER	Yes	Yes	No	No
SVC_HALF_DATA_BUDGET	Yes	Yes	No	Yes
CODE_DETAIL	Yes	No	No	No
JOB_AUDIT_LOGS	No	Yes	No	No
JOB_AUDIT_PAYLOAD	No	Yes	No	No

Inventory Adjustment Service

This section describes the Inventory Adjustment service.

Functional Area

Inventory

Business Overview

The primary role of this service is to create inventory adjustment and send them to Merchandising.

Inventory Adjustment

Business Overview

This service creates inventory adjustment by calling the package SVCPROV_INVADJ to load input data to the staging tables and then calling the core inventory adjustment package to validate and insert data to the Merchandising tables.

Service Type

Post

ReST URL

Invadj/createInvadj

Input Parameters

SvcprovInvadjdescRecRDO

Parameter Name	Data Type
location	BigDecimal
Invadjdtl	List<SvcprovInvadjdescdtlRecRDO>

SvcprovInvadjdescdtlRecRDO

Parameter Name	Data Type
unitQty	BigDecimal
toDisposition	String
adjReasonCode	BigDecimal
docType	String
toWipCode	String
item	String
poNbr	String
auxReasonCode	String
weight	BigDecimal
toTroubleCode	String
fromWipCode	String
weightUom	String
unitCost	BigDecimal
fromTroubleCode	String
transshipmentNumber	String
fromDisposition	String
transactionCode	BigDecimal
adjTranDate	Date (Format: 'YYYY-MM-DD')
userId	String

JSON Structure:

```
[
  {
    "location":null,
    "invadjdtl":[
      {
        "unitQty":null,
        "toDisposition":null,
        "adjReasonCode":null,
        "docType":null,
        "toWipCode":null,
        "item":null,
        "poNbr":null,
        "auxReasonCode":null,
        "weight":null,
        "toTroubleCode":null,
        "fromWipCode":null,

```

```

        "weightUom":null,
        "unitCost":null,
        "fromTroubleCode":null,
        "transshipmentNumber":null,
        "fromDisposition":null,
        "transactionCode":null,
        "adjTranDate":null,
        "userId":null
    }
  ]
}
]

```

Output

SvcprovInvadjStatusRecRDO

Parameter Name	Data Type
statusMsg	String
invadjErrTbl	List< SvcprovFailInvadjRecRDO >

SvcprovFailInvadjRecRDO

Parameter Name	Data Type
location	BigDecimal
unitQty	BigDecimal
adjReasonCode	BigDecimal
item	String
errorMsg	String

The output will contain the status of the request including validation errors, if any.

JSON Structure:

```

{
  "statusMsg": null,
  " invadjErrTbl ": [
    {
      " location": null,
      "unitQty": null,
      " adjReasonCode": null,
      " item": null,
      "errorMsg": null,
      "links": [],
      "hyperMediaContent": {
        "linkRDO": []
      }
    }
  ],
  "links": [],
  "hyperMediaContent": {
    "linkRDO": []
  }
}

```

Table Impact

TABLE	SELECT	INSERT	UPDATE	DELETE
INV_ADJ	Yes	Yes	No	No
ITEM_LOC_SOH	Yes	Yes	Yes	No
SVC_PROCESS_TRACKER	Yes	Yes	Yes	No
SVC_INV_ADJ	Yes	Yes	Yes	Yes
JOB_AUDIT_LOGS	No	Yes	No	No
JOB_AUDIT_PAYLOAD	No	Yes	No	No

Item Detail Service

This section describes the Item Detail service.

Business Overview

Item Detail service allows user to retrieve Item information for a selected item.

Service Type

Get

ReST URL

Item/itemDetail?item={itemNumber}

Input Parameters

Parameter Name	Required	Description
Item	Yes	Item number

Output

RestItemRecRDO

Parameter Name	Data Type
itemGrandparent	String
itemParent	String
item	String
itemDesc	String
shortDesc	String
packInd	String
status	String
itemLevel	BigDecimal

Parameter Name	Data Type
tranLevel	BigDecimal
dept	BigDecimal
classAttribute	BigDecimal
subclass	BigDecimal
diff1	String
diff2	String
diff3	String
diff4	String
primaryReflItemInd	String
originalRetail	BigDecimal
sellableInd	String
orderableInd	String
inventoryInd	String
packitemBreakout	List<RestPackitemBreakoutRecRDO>
itemSupplier	List<RestItemSupplierRecRDO>
itemSupplierCountry	List<RestItemSupplierCountryRecRDO>
vatItem	List<RestVatItemRecRDO>

RestPackitemBreakoutRecRDO

Parameter Name	Data Type
item	String
seqNo	BigDecimal
packItemQty	BigDecimal

RestItemSupplierRecRDO

Parameter Name	Data Type
supplier	BigDecimal
vpn	String
primarySupplInd	String
directShipInd	String

RestItemSupplierCountryRecRDO

Parameter Name	Data Type
originCountryId	String
primaryCountryInd	String
unitCost	BigDecimal
suppPackSize	BigDecimal
innerPackSize	BigDecimal

Parameter Name	Data Type
leadTime	BigDecimal
pickupLeadTime	BigDecimal

RestVatItemRecRDO

Parameter Name	Data Type
vatRegion	BigDecimal
vatType	String
vatCode	String
vatRate	BigDecimal
activeDate	Timestamp

JSON Structure:

```
{
  "itemGrandparent": null,
  "itemParent": null,
  "item": null,
  "itemDesc": null,
  "shortDesc": null,
  "packInd": null,
  "status": null,
  "itemLevel": null,
  "tranLevel": null,
  "dept": null,
  "classAttribute": null,
  "subclass": null,
  "diff1": null,
  "diff2": null,
  "diff3": null,
  "diff4": null,
  "primaryRefItemInd": null,
  "originalRetail": null,
  "sellableInd": null,
  "orderableInd": null,
  "inventoryInd": null,
  "packitemBreakout": [],
  "itemSupplier": [
    {
      "primarySuppInd": null,
      "itemSupplierCountry": [
        {
          "unitCost": null,
          "leadTime": null,
          "suppPackSize": null,
          "originCountryId": null,
          "primaryCountryInd": null,
          "pickupLeadTime": null,
          "innerPackSize": null,
          "links": [],
          "hyperMediaContent": {
            "linkRDO": []
          }
        }
      ]
    }
  ],
}
```

```

        "supplier": null,
        "vpn": null,
        "directShipInd": null,
        "links": [],
        "hyperMediaContent": {
            "linkRDO": []
        }
    }
},
"vatItem": [
    {
        "vatRegion": null,
        "activeDate": null,
        "vatType": null,
        "vatCode": null,
        "vatRate": null,
        "links": [],
        "hyperMediaContent": {
            "linkRDO": []
        }
    }
],
"links": [],
"hyperMediaContent": {
    "linkRDO": []
}
}

```

Table Impact

TABLE	SELECT	INSERT	UPDATE	DELETE
V_ITEM_MASTER	Yes	No	No	No
PACKITEM_BREAKOUT	Yes	No	No	No
ITEM_SUPPLIER	Yes	No	No	No
ITEM_SUPP_COUNTRY	Yes	No	No	No
VAT_ITEM	Yes	No	No	No
JOB_AUDIT_LOGS	No	Yes	No	No
JOB_AUDIT_PAYLOAD	No	Yes	No	No

Item Loc Inventory Detail Service

This section describes the Item Loc Inventory Detail service.

Business Overview

Item Loc Inventory Detail service allows user to retrieve Item Location and Item Location Stock on Hand information for a selected item and location. If location and location type are not specified, all locations for the item will be retrieved. If location type is specified but not the location, all locations for the item and location type will be retrieved.

Service Type

Get

ReST URL

ItemlocInvDtl/itemlocInvDetail?
item={itemNumber}&location={locationNumber}&locationType={locationType}

Input Parameters

Parameter Name	Required	Description
Item	Yes	Item ID
Location	No	Location ID.
Location Type	No	Location Type.

Output

RestItemlocInvDtlRecRDO

Parameter Name	Data Type
item	String
itemParent	String
loc	BigDecimal
locType	String
unitRetail	BigDecimal
sellingUom	String
clearInd	String
taxableInd	String
localItemDesc	String
status	String
primarySupp	BigDecimal
primaryCntry	String
avCost	BigDecimal
unitCost	BigDecimal
stockOnHand	BigDecimal
sohUpdateDatetime	Timestamp
inTransitQty	BigDecimal
packCompSoh	BigDecimal
packCompResv	BigDecimal
packCompExp	BigDecimal
rtvQty	BigDecimal
customerResv	BigDecimal

Parameter Name	Data Type
sellingUnitRetail	BigDecimal
localShortDesc	String
packCompIntran	BigDecimal
tsfReservedQty	BigDecimal
tsfExpectedQty	BigDecimal
nonSellableQty	BigDecimal
customerBackorder	BigDecimal
packCompCustResv	BigDecimal
packCompCustBack	BigDecimal
packCompNonSellable	BigDecimal
firstReceived	Timestamp
lastReceived	Timestamp

JSON Structure:

```
{
  "item": null,
  "itemParent": null,
  "loc": null,
  "locType": null,
  "unitRetail": null,
  "sellingUom": null,
  "clearInd": null,
  "taxableInd": null,
  "localItemDesc": null,
  "status": null,
  "primarySupp": null,
  "primaryCntry": null,
  "avCost": null,
  "unitCost": null,
  "stockOnHand": null,
  "sohUpdateDatetime": null,
  "inTransitQty": null,
  "packCompSoh": null,
  "packCompResv": null,
  "packCompExp": null,
  "rtvQty": null,
  "customerResv": null,
  "sellingUnitRetail": null,
  "localShortDesc": null,
  "packCompIntran": null,
  "tsfReservedQty": null,
  "tsfExpectedQty": null,
  "nonSellableQty": null,
  "customerBackorder": null,
  "packCompCustResv": null,
  "packCompCustBack": null,
  "packCompNonSellable": null,
  "firstReceived": null,
  "lastReceived": null,
  "links": [],
  "hyperMediaContent": {
    "linkRDO": []
  }
}
```



```

    }
}

```

Table Impact

TABLE	SELECT	INSERT	UPDATE	DELETE
V_ITEM_LOC	Yes	No	No	No
ITEM_LOC_SOH	Yes	No	No	No
JOB_AUDIT_LOGS	No	Yes	No	No
JOB_AUDIT_PAYLOAD	No	Yes	No	No

MerchHierarchy Detail Service

This section describes the MerchHierarchy Detail service.

Business Overview

MerchHierarchyDetail service allows user to retrieve full merchandising hierarchy information.

Service Type

Get

ReST URL

/MerchHierarchy/merchHierarchy

Input Parameters

NA

Output

RestMerchHierarchyRecRDO

Parameter Name	Data Type
profitCalcType	BigDecimal
deptVatInclInd	String
classAttribute	BigDecimal
division	BigDecimal
classVatInd	String
subclass	BigDecimal
buyer	BigDecimal
dept	BigDecimal
className	String
subName	String

Parameter Name	Data Type
groupNo	BigDecimal
otbCalcType	String
groupName	String
divName	String
purchaseType	BigDecimal
merch	BigDecimal
deptName	String

JSON Structure

```
{
  "profitCalcType": null,
  "deptVatInclInd": null,
  "classAttribute": null,
  "division": null,
  "classVatInd": null,
  "subclass": null,
  "buyer": null,
  "dept": null,
  "className": null,
  "subName": null,
  "groupNo": null,
  "otbCalcType": null,
  "groupName": null,
  "divName": null,
  "purchaseType": null,
  "merch": null,
  "deptName": null,
  "links": [],
  "hyperMediaContent": {
    "linkRDO": []
  }
}
```

Table Impact

TABLE	SELECT	INSERT	UPDATE	DELETE
V_MERCH_HIERARCHY	Yes	No	No	No
JOB_AUDIT_LOGS	No	Yes	No	No
JOB_AUDIT_PAYLOAD	No	Yes	No	No

Obligations Upload Services

This section describes the Obligations Upload Services:

Function Area

Import export

Business Overview

The primary role of these services is to support requests from trading partners or suppliers for bulk uploads of obligations to send to Merchandising.

Create Obligation

Business Overview

This service creates obligations by calling the SVCPROV_OBLIGATION package and then calling the core obligation package to validate and insert data into the Merchandising tables.

This service creates obligations with required fields. At least one component is needed to create a successful obligation. If the obligation level is 'PO' or 'POIT', then component locations can be added. Reallocation to ALC will be done after obligation creation. If an obligation is created in approved status, then an invoice will be created.

Service Type

Post

ReST URL

/ObligationUpload/createObligation

Input Parameters

RestObligationRecRDO

Parameter Name	Data Type
obligationKey	BigDecimal
obligationLevel	String
keyValue1	String
keyValue2	String
keyValue3	String
keyValue4	String
keyValue5	String
keyValue6	String
status	String
partnerType	String
partnerId	String
supplierSite	String
invoice	String
invoiceDate	Timestamp
qty	BigDecimal
qtyUom	String
currency	String
exchangeRate	BigDecimal

Parameter Name	Data Type
checkAuthNo	String
paidAmt	BigDecimal
paidDate	Timestamp
comments	String
obligationCompTbl	List<RestObligationCompRecRDO>

RestObligationCompRecRDO

Parameter Name	Data Type
compId	String
allocateToAlc	String
allocationType	String
allocationBasisUom	String
amt	BigDecimal
rate	BigDecimal
perCount	BigDecimal
perCountUom	String
obligationCompLocTbl	List<RestObligationCompLocRecRDO>

RestObligationCompLocRecRDO

Parameter Name	Data Type
action	String
locType	String
locId	BigDecimal
qty	BigDecimal
amt	BigDecimal

JSON Structure

invoiceDate and paidDate will take input in strings the 'YYYY-MM-DD' format that will be converted to Timestamp format and inserted into tables.

```
[{
  "obligationKey": null,
  "obligationLevel":null ,
  "keyValue1":null,
  "keyValue2":null,
  "keyValue3":null,
  "keyValue4":null,
  "keyValue5":null,
  "keyValue6":null,
  "status":null,
  "partnerType":null,
  "partnerId":null,
  "supplierSite":null,
  "invoice":null,
```

```

    "invoiceDate":null,
    "qty":null,
    "qtyUom":null,
    "exchangeRate":null,
    "currency":null,
    "paymentMethod":null,
    "checkAuthNo":null,
    "paidAmt":null,
    "paidDate":null,
    "comments":null,
    "obligationCompTbl" : [{
        "compId":null,
        "allocateToAic":null,
        "allocationType":null,
        "allocationBasisUom": null,
        "amt":null,
        "rate":null,
        "perCount":null,
        "perCountUom":null,
        "obligationCompLocTbl":[{
            "action":null,
            "locType":null,
            "locId": null,
            "qty":null,
            "amt":null}}
    ]
}

```

Output

RestObligationStatusRecRDO

Parameter Name	Data Type
successObligationTbl	List<RestObligationSuccessRecRDO>
successObligationCount	BigDecimal
failObligationTbl	List<RestObligationFailRecRDO>
failObligationCount	BigDecimal

RestObligationSuccessRecRDO

Parameter Name	Data Type
obligationKey	BigDecimal
status	String
invoice	String

RestObligationFailRecRDO

Parameter Name	Data Type
obligationKey	BigDecimal
status	String
invoice	String
errorMessage	String

The output will contain the status of the request including validation error, if any.

It will insert data into the obligation, obligation_comp and obligation_comp_loc tables, based on obligation levels. It will also insert data to ALC tables. For an approved obligation, it will insert data to INVC tables.

For Success:

```
{
  "successObligationTbl": [
    {
      "status": null,
      "invoice": null,
      "obligationKey": null,
      "links": [],
      "hyperMediaContent": {
        "linkRDO": []
      }
    }
  ],
  "successObligationCount": null,
  "failObligationTbl": [],
  "failObligationCount": null,
  "links": [],
  "hyperMediaContent": {
    "linkRDO": []
  }
}
```

For Failure:

```
{
  "successObligationTbl": [],
  "successObligationCount": null,
  "failObligationTbl": [
    {
      "errorMessage": null,
      "status": null,
      "invoice": null,
      "obligationKey": null,
      "links": [],
      "hyperMediaContent": {
        "linkRDO": []
      }
    }
  ],
  "failObligationCount": null,
  "links": [],
  "hyperMediaContent": {
    "linkRDO": []
  }
}
```

Table Impact

TABLE	SELECT	INSERT	UPDATE	DELETE
OBLIGATION	No	Yes	No	No
OBLIGATION_COMP	No	Yes	No	No
OBLIGATION_COMP_LOC	No	Yes	No	No

TABLE	SELECT	INSERT	UPDATE	DELETE
ORDHEAD	Yes	No	No	No
ORDSKU	Yes	No	No	No
ORDLOC	Yes	No	No	No
TRANSPORTATION	Yes	No	No	No
SHIPMENT	Yes	No	No	No
SHIPSKU	Yes	No	No	No
V_PACKSKU_QTY	Yes	No	No	No
ITEM_SUPP_COUNTRY	Yes	No	No	No
ORDLOC_EXP	Yes	No	No	No
CE_HEAD	Yes	No	No	No
ELC_COMP	Yes	No	No	No
RTM_UNIT_OPTIONS	Yes	No	No	No
ALC_HEAD	Yes	Yes	Yes	No
ALC_COMP_LOC	Yes	Yes	Yes	No
INVC_HEAD	No	Yes	No	No
INVC_NON_MERCH	No	Yes	No	No
INVC_XREF	No	Yes	No	No

Create Obligation Component

Business Overview

This service creates obligation components for existing pending obligations by calling the SVCPROV_OBLIGATION package and then calling the core obligation package to validate and insert data into the Merchandising tables.

This service accepts an obligation key, or supplier/invoice, or partner type/partner/invoice combination to identify the existing obligation. Add component details which user wants to create. For obligation levels 'PO' and 'POIT', the user can also add location details. Based on validations, components will also be created. Reallocation to ALC will be done after component creation.

Service Type

Post

ReST URL

/ObligationUpload/createObligationComp

Input Parameters

RestObligationRecRDO

Parameter Name	Data Type
obligationKey	BigDecimal

Parameter Name	Data Type
obligationLevel	String
keyValue1	String
keyValue2	String
keyValue3	String
keyValue4	String
keyValue5	String
keyValue6	String
Status	String
partnerType	String
partnerId	String
supplierSite	String
invoice	String
invoiceDate	Timestamp
Qty	BigDecimal
qtyUom	String
Currency	String
exchangeRate	BigDecimal
checkAuthNo	String
paidAmt	BigDecimal
paidDate	Timestamp
Comments	String
obligationCompTbl	List<RestObligationCompRecRDO>

RestObligationCompRecRDO

Parameter Name	Data Type
compld	String
allocateToAlc	String
allocationType	String
allocationBasisUom	String
Amt	BigDecimal
Rate	BigDecimal
perCount	BigDecimal
perCountUom	String
obligationCompLocTbl	List<RestObligationCompLocRecRDO>

RestObligationCompLocRecRDO

Parameter Name	Data Type
Action	String

Parameter Name	Data Type
locType	String
locId	BigDecimal
Qty	BigDecimal
Amt	BigDecimal

JSON Structure: The same RDO 'RestObligationRecRDO' will be used for createObligationComp, but only the below parameters will be considered for the request. The rest of the parameters will be ignored.

```
{
  "obligationKey": null,
  "partnerType": null,
  "partnerId": null,
  "supplierSite": null,
  "invoice": null,
  "obligationCompTbl" : [
    {
      "compId": null,
      "allocateToAloc": null,
      "allocationType": null,
      "allocationBasisUom": null,
      "amt": null,
      "rate": null,
      "perCount": null,
      "perCountUom": null,
      "obligationCompLocTbl": [
        {
          "action": null,
          "locType": null,
          "locId": null,
          "qty": null,
          "amt": null}}
      ]
    }
  ]
}
```

Output

RestObligationStatusRecRDO

Parameter Name	Data Type
successObligationTbl	List<RestObligationSuccessRecRDO>
successObligationCount	BigDecimal
failObligationTbl	List<RestObligationFailRecRDO>
failObligationCount	BigDecimal

RestObligationSuccessRecRDO

Parameter Name	Data Type
obligationKey	BigDecimal
status	String

Parameter Name	Data Type
invoice	String

RestObligationFailRecRDO

Parameter Name	Data Type
obligationKey	BigDecimal
status	String
invoice	String
errorMessage	String

The output will contain the status of the request including validation errors, if any.

It will insert data into the tables `obligation_comp` and `obligation_comp_loc` (only for 'PO' and 'POIT'). It will also insert data to into ALC tables.

For Success:

```
{
  "successObligationTbl": [
    {
      "status": null,
      "invoice": null,
      "obligationKey": null,
      "links": [],
      "hyperMediaContent": {
        "linkRDO": []
      }
    }
  ],
  "successObligationCount": null,
  "failObligationTbl": [],
  "failObligationCount": null,
  "links": [],
  "hyperMediaContent": {
    "linkRDO": []
  }
}
```

For Failure:

```
{
  "successObligationTbl": [],
  "successObligationCount": null,
  "failObligationTbl": [
    {
      "errorMessage": null,
      "status": null,
      "invoice": null,
      "obligationKey": null,
      "links": [],
      "hyperMediaContent": {
        "linkRDO": []
      }
    }
  ],
}
```

```

    "failObligationCount": null,
    "links": [],
    "hyperMediaContent": {
      "linkRDO": []
    }
  }
}

```

Table Impact

TABLE	SELECT	INSERT	UPDATE	DELETE
OBLIGATION	Yes	No	No	No
OBLIGATION_COMP	Yes	Yes	No	No
OBLIGATION_COMP_LOC	Yes	Yes	No	No
ORDHEAD	Yes	No	No	No
ORDSKU	Yes	No	No	No
ORDLOC	Yes	No	No	No
SHIPMENT	Yes	No	No	No
SHIPSKU	Yes	No	No	No
V_PACKSKU_QTY	Yes	No	No	No
ITEM_SUPP_COUNTRY	Yes	No	No	No
ORDLOC_EXP	Yes	No	No	No
ELC_COMP	Yes	No	No	No
ALC_HEAD	Yes	Yes	Yes	No
ALC_COMP_LOC	Yes	Yes	Yes	No

Create Obligation Component Location

Business Overview

This service creates obligation component locations for existing pending obligations by calling the SVCPROV_OBLIGATION package and then calling the core obligation package to validate and insert data into the Merchandising tables.

This service accepts an obligation key, or supplier/invoice, or partner type/partner/invoice combination to identify the existing obligation. Add the component ID for which the user wants to create locations. Add valid location details. Reallocation to ALC will be done after successful location creation.

Service Type

Post

ReST URL

/ObligationUpload/createObligationCompLoc

Input Parameters

RestObligationRecRDO

Parameter Name	Data Type
obligationKey	BigDecimal
obligationLevel	String
keyValue1	String
keyValue2	String
keyValue3	String
keyValue4	String
keyValue5	String
keyValue6	String
status	String
partnerType	String
partnerId	String
supplierSite	String
invoice	String
invoiceDate	Timestamp
qty	BigDecimal
qtyUom	String
currency	String
exchangeRate	BigDecimal
checkAuthNo	String
paidAmt	BigDecimal
paidDate	Timestamp
comments	String
obligationCompTbl	List<RestObligationCompRecRDO>

RestObligationCompRecRDO

Parameter Name	Data Type
compId	String
allocateToAlc	String
allocationType	String
allocationBasisUom	String
amt	BigDecimal
rate	BigDecimal
perCount	BigDecimal
perCountUom	String
obligationCompLocTbl	List<RestObligationCompLocRecRDO>

RestObligationCompLocRecRDO

Parameter Name	Data Type
action	String
locType	String
locId	BigDecimal
qty	BigDecimal
amt	BigDecimal

JSON Structure: Same RDO 'RestObligationRecRDO' will be used for createObligationComp, but only the below parameters will be considered for the request. The rest of the parameters will be ignored.

```
[{
  "obligationKey": null,
  "partnerType":null,
  "partnerId":null,
  "supplierSite":null,
  "invoice":null,
  "obligationCompTbl" : [{
    "compId":null,
    "obligationCompLocTbl":[{
      "action":null,
      "locType":null,
      "locId": null,
      "qty":null,
      "amt":null}}
  ]
}]
```

Output

RestObligationStatusRecRDO

Parameter Name	Data Type
successObligationTbl	List<RestObligationSuccessRecRDO>
successObligationCount	BigDecimal
failObligationTbl	List<RestObligationFailRecRDO>
failObligationCount	BigDecimal

RestObligationSuccessRecRDO

Parameter Name	Data Type
obligationKey	BigDecimal
status	String
invoice	String

RestObligationFailRecRDO

Parameter Name	Data Type
obligationKey	BigDecimal

Parameter Name	Data Type
status	String
invoice	String
errorMessage	String

The output will contain the status of the request, including validation errors, if any.

For Success:

```
{
  "successObligationTbl": [
    {
      "status": null,
      "invoice": null,
      "obligationKey": null,
      "links": [],
      "hyperMediaContent": {
        "linkRDO": []
      }
    }
  ],
  "successObligationCount": null,
  "failObligationTbl": [],
  "failObligationCount": null,
  "links": [],
  "hyperMediaContent": {
    "linkRDO": []
  }
}
```

For Failure:

```
{
  "successObligationTbl": [],
  "successObligationCount": null,
  "failObligationTbl": [
    {
      "errorMessage": null,
      "status": null,
      "invoice": null,
      "obligationKey": null,
      "links": [],
      "hyperMediaContent": {
        "linkRDO": []
      }
    }
  ],
  "failObligationCount": null,
  "links": [],
  "hyperMediaContent": {
    "linkRDO": []
  }
}
```

Table Impact

TABLE	SELECT	INSERT	UPDATE	DELETE
OBLIGATION	Yes	No	No	No
OBLIGATION_COMP	Yes	No	No	No
OBLIGATION_COMP_LOC	Yes	Yes	No	No
ORDHEAD	Yes	No	No	No
ORDSKU	Yes	No	No	No
ORDLOC	Yes	No	No	No
SHIPMENT	Yes	No	No	No
SHIPSKU	Yes	No	No	No
V_PACKSKU_QTY	Yes	No	No	No
ITEM_SUPP_COUNTRY	Yes	No	No	No
ORDLOC_EXP	Yes	No	No	No
ALC_HEAD	Yes	Yes	Yes	No
ALC_COMP_LOC	Yes	Yes	Yes	No

Modify Obligation

Business Overview

This service updates obligation header records by calling the SVCPROV_OBLIGATION package and then calling the core obligation package to validate and update data in the Obligation table.

This service accepts an obligation key, or supplier/invoice, or partner type/partner/invoice combination to identify the existing obligation. The user can modify the mandatory fields below when the obligation is in pending status. Reallocation to ALC will be performed for pending obligations. If the user updates the status to approved, then an invoice will be created.

- Status
- Partner Type
- Partner ID
- Supplier Site
- Invoice
- Invoice Date
- Quantity
- Quantity UOM
- Currency
- Exchange Rate

The below fields, which are not mandatory, can be updated in pending as well as approved status.

- Payment Method

- Check Authorization No.
- Amount Paid
- Paid Date
- Comments

Service Type

Post

ReST URL

/ObligationUpload/modifyObligation

RestObligationRecRDO

Parameter Name	Data Type
obligationKey	BigDecimal
obligationLevel	String
keyValue1	String
keyValue2	String
keyValue3	String
keyValue4	String
keyValue5	String
keyValue6	String
status	String
partnerType	String
partnerId	String
supplierSite	String
invoice	String
invoiceDate	Timestamp
qty	BigDecimal
qtyUom	String
currency	String
exchangeRate	BigDecimal
checkAuthNo	String
paidAmt	BigDecimal
paidDate	Timestamp
comments	String
obligationCompTbl	List<RestObligationCompRecRDO>

RestObligationCompRecRDO

Parameter Name	Data Type
compld	String
allocateToAlc	String

Parameter Name	Data Type
allocationType	String
allocationBasisUom	String
amt	BigDecimal
rate	BigDecimal
perCount	BigDecimal
perCountUom	String
obligationCompLocTbl	List<RestObligationCompLocRecRDO>

RestObligationCompLocRecRDO

Parameter Name	Data Type
action	String
locType	String
locId	BigDecimal
qty	BigDecimal
amt	BigDecimal

JSON Structure: Same RDO 'RestObligationRecRDO' will be used for modifyObligation but only the below parameters will be considered for the request. The rest of the parameters will be ignored.

invoiceDate and paidDate will take input in a string with the 'YYYY-MM-DD' format, which will be converted to Timestamp format and insert into the tables.

```
[{
  "obligationKey": null,
  "status":null,
  "partnerType":null,
  "partnerId":null,
  "supplierSite":null,
  "invoice":null,
  "invoiceDate":null,
  "qty":null,
  "qtyUom":null,
  "exchangeRate":null,
  "currency":null,
  "paymentMethod":null,
  "checkAuthNo":null,
  "paidAmt":null,
  "paidDate":null,
  "comments":null
}]
```

Output

RestObligationStatusRecRDO

Parameter Name	Data Type
successObligationTbl	List<RestObligationSuccessRecRDO>

Parameter Name	Data Type
successObligationCount	BigDecimal
failObligationTbl	List<RestObligationFailRecRDO>
failObligationCount	BigDecimal

RestObligationSuccessRecRDO

Parameter Name	Data Type
obligationKey	BigDecimal
status	String
invoice	String

RestObligationFailRecRDO

Parameter Name	Data Type
obligationKey	BigDecimal
status	String
invoice	String
errorMessage	String

The output will contain the status of the request, including validation errors, if any.

NULL values will be ignored for mandatory fields. Non-mandatory fields can be updated to NULL.

For Success:

```
{
  "successObligationTbl": [
    {
      "status": null,
      "invoice": null,
      "obligationKey": null,
      "links": [],
      "hyperMediaContent": {
        "linkRDO": []
      }
    }
  ],
  "successObligationCount": null,
  "failObligationTbl": [],
  "failObligationCount": null,
  "links": [],
  "hyperMediaContent": {
    "linkRDO": []
  }
}
```

For Failure:

```
{
  "successObligationTbl": [],
```

```

    "successObligationCount": null,
    "failObligationTbl": [
      {
        "errorMessage": null,
        "status": null,
        "invoice": null,
        "obligationKey": null,
        "links": [],
        "hyperMediaContent": {
          "linkRDO": []
        }
      }
    ],
    "failObligationCount": null,
    "links": [],
    "hyperMediaContent": {
      "linkRDO": []
    }
  }
}

```

Table Impact

TABLE	SELECT	INSERT	UPDATE	DELETE
OBLIGATION	Yes	No	No	No
OBLIGATION_COMP	Yes	No	No	No
OBLIGATION_COMP_LOC	Yes	No	No	No
ORDHEAD	Yes	No	No	No
ORDSKU	Yes	No	No	No
ORDLOC	Yes	No	No	No
SHIPMENT	Yes	No	No	No
SHIPSKU	Yes	No	No	No
V_PACKSKU_QTY	Yes	No	No	No
ITEM_SUPP_COUNTRY	Yes	No	No	No
ORDLOC_EXP	Yes	No	No	No
ALC_HEAD	Yes	Yes	Yes	No
ALC_COMP_LOC	Yes	Yes	Yes	No
INVC_HEAD	No	Yes	No	No
INVC_NON_MERCH	No	Yes	No	No
INVC_XREF	No	Yes	No	No

Modify Obligation Comp

Business Overview

This service updates existing obligation component records which are in pending status by calling the SVCPROV_OBLIGATION package and then calling the core obligation package to validate and update data in the obligation_comp table.

This service accepts an obligation key, or supplier/invoice, or partner type/partner/invoice combination to identify the existing obligation. The user can modify component level fields. Reallocation to ALC will be done after update.

Service Type

Post

ReST URL

/ObligationUpload/modifyObligationComp

Input Parameters

RestObligationRecRDO

Parameter Name	Data Type
obligationKey	BigDecimal
obligationLevel	String
keyValue1	String
keyValue2	String
keyValue3	String
keyValue4	String
keyValue5	String
keyValue6	String
status	String
partnerType	String
partnerId	String
supplierSite	String
invoice	String
invoiceDate	Timestamp
qty	BigDecimal
qtyUom	String
currency	String
exchangeRate	BigDecimal
checkAuthNo	String
paidAmt	BigDecimal
paidDate	Timestamp
comments	String
obligationCompTbl	List<RestObligationCompRecRDO>

RestObligationCompRecRDO

Parameter Name	Data Type
compId	String
allocateToAlc	String
allocationType	String
allocationBasisUom	String
amt	BigDecimal

Parameter Name	Data Type
rate	BigDecimal
perCount	BigDecimal
perCountUom	String
obligationCompLocTbl	List<RestObligationCompLocRecRDO>

RestObligationCompLocRecRDO

Parameter Name	Data Type
action	String
locType	String
locId	BigDecimal
qty	BigDecimal
amt	BigDecimal

JSON Structure: The same RDO 'RestObligationRecRDO' will be used for modifyObligationComp, but only the below parameters will be considered for the request. The rest of the parameters will be ignored.

```
[{
  "obligationKey": null,
  "partnerType":null,
  "partnerId":null,
  "supplierSite":null,
  "invoice":null,
  "obligationCompTbl" : [{"compId":null,
    "allocateToAloc":null,
    "allocationType":null,
    "allocationBasisUom": null,
    "amt":null,
    "rate":null,
    "perCount":null,
    "perCountUom":null,
  }]}
]
```

Output

RestObligationStatusRecRDO

Parameter Name	Data Type
successObligationTbl	List<RestObligationSuccessRecRDO>
successObligationCount	BigDecimal
failObligationTbl	List<RestObligationFailRecRDO>
failObligationCount	BigDecimal

RestObligationSuccessRecRDO

Parameter Name	Data Type
obligationKey	BigDecimal
status	String
invoice	String

RestObligationFailRecRDO

Parameter Name	Data Type
obligationKey	BigDecimal
status	String
invoice	String
errorMessage	String

The output will contain the status of the request, including validation errors, if any.

For Success:

```
{
  "successObligationTbl": [
    {
      "status": null,
      "invoice": null,
      "obligationKey": null,
      "links": [],
      "hyperMediaContent": {
        "linkRDO": []
      }
    }
  ],
  "successObligationCount": null,
  "failObligationTbl": [],
  "failObligationCount": null,
  "links": [],
  "hyperMediaContent": {
    "linkRDO": []
  }
}
```

For Failure:

```
{
  "successObligationTbl": [],
  "successObligationCount": null,
  "failObligationTbl": [
    {
      "errorMessage": null,
      "status": null,
      "invoice": null,
      "obligationKey": null,
      "links": [],
      "hyperMediaContent": {
        "linkRDO": []
      }
    }
  ],
}
```

```

    "failObligationCount": null,
    "links": [],
    "hyperMediaContent": {
        "linkRDO": []
    }
}

```

Table Impact

TABLE	SELECT	INSERT	UPDATE	DELETE
OBLIGATION	Yes	No	No	No
OBLIGATION_COMP	Yes	No	No	No
OBLIGATION_COMP_LOC	Yes	No	No	No
ORDHEAD	Yes	No	No	No
ORDSKU	Yes	No	No	No
ORDLOC	Yes	No	No	No
SHIPMENT	Yes	No	No	No
SHIPSKU	Yes	No	No	No
V_PACKSKU_QTY	Yes	No	No	No
ITEM_SUPP_COUNTRY	Yes	No	No	No
ORDLOC_EXP	Yes	No	No	No
ALC_HEAD	Yes	Yes	Yes	No
ALC_COMP_LOC	Yes	Yes	Yes	No

Modify Obligation Component Location

Business Overview

This service updates existing obligation component records which are in pending status by calling the SVCPROV_OBLIGATION package and then calling the core obligation package to validate and update data in the obligation_comp table.

This service accepts an obligation key, or supplier/invoice, or partner type/partner/invoice combination to identify the existing obligation. The user can modify component level fields. Reallocation to ALC will be done after the update.

Service Type

Post

ReST URL

/ObligationUpload/modifyObligationCompLoc

Input Parameters

RestObligationRecRDO

Parameter Name	Data Type
obligationKey	BigDecimal

Parameter Name	Data Type
obligationLevel	String
keyValue1	String
keyValue2	String
keyValue3	String
keyValue4	String
keyValue5	String
keyValue6	String
status	String
partnerType	String
partnerId	String
supplierSite	String
invoice	String
invoiceDate	Timestamp
qty	BigDecimal
qtyUom	String
currency	String
exchangeRate	BigDecimal
checkAuthNo	String
paidAmt	BigDecimal
paidDate	Timestamp
comments	String
obligationCompTbl	List<RestObligationCompRecRDO>

RestObligationCompRecRDO

Parameter Name	Data Type
compld	String
allocateToAlc	String
allocationType	String
allocationBasisUom	String
amt	BigDecimal
rate	BigDecimal
perCount	BigDecimal
perCountUom	String
obligationCompLocTbl	List<RestObligationCompLocRecRDO>

RestObligationCompLocRecRDO

Parameter Name	Data Type
action	String

Parameter Name	Data Type
locType	String
locId	BigDecimal
qty	BigDecimal
amt	BigDecimal

JSON Structure: The same RDO 'RestObligationRecRDO' will be used for modifyObligationCompLoc but only the below parameters will be considered for the request. The rest of the parameters will be ignored.

```
[{
  "obligationKey": null,
  "partnerType":null,
  "partnerId":null,
  "supplierSite":null,
  "invoice":null,
  "obligationCompTbl" : [{
    "compId":null,
    "obligationCompLocTbl":[{
      "action":null,
      "locType":null,
      "locId": null,
      "qty":null,
      "amt":null}]
  }]
}]
```

Output

RestObligationStatusRecRDO

Parameter Name	Data Type
successObligationTbl	List<RestObligationSuccessRecRDO>
successObligationCount	BigDecimal
failObligationTbl	List<RestObligationFailRecRDO>
failObligationCount	BigDecimal

RestObligationSuccessRecRDO

Parameter Name	Data Type
obligationKey	BigDecimal
status	String
invoice	String

RestObligationFailRecRDO

Parameter Name	Data Type
obligationKey	BigDecimal
status	String

Parameter Name	Data Type
invoice	String
errorMessage	String

The output will contain the status of the request, including validation errors, if any.

The action field is mandatory here. Valid value for this are MOD/DEL. MOD is to update a location and DEL to delete a location.

For Success:

```
{
  "successObligationTbl": [
    {
      "status": null,
      "invoice": null,
      "obligationKey": null,
      "links": [],
      "hyperMediaContent": {
        "linkRDO": []
      }
    }
  ],
  "successObligationCount": null,
  "failObligationTbl": [],
  "failObligationCount": null,
  "links": [],
  "hyperMediaContent": {
    "linkRDO": []
  }
}
```

For Failure:

```
{
  "successObligationTbl": [],
  "successObligationCount": null,
  "failObligationTbl": [
    {
      "errorMessage": null,
      "status": null,
      "invoice": null,
      "obligationKey": null,
      "links": [],
      "hyperMediaContent": {
        "linkRDO": []
      }
    }
  ],
  "failObligationCount": null,
  "links": [],
  "hyperMediaContent": {
    "linkRDO": []
  }
}
```

Table Impact

TABLE	SELECT	INSERT	UPDATE	DELETE
OBLIGATION	Yes	No	No	No
OBLIGATION_COMP	Yes	No	Yes	No
OBLIGATION_COMP_LOC	Yes	No	Yes	No
ORDHEAD	Yes	No	No	No
ORDSKU	Yes	No	No	No
ORDLOC	Yes	No	No	No
SHIPMENT	Yes	No	No	No
SHIPSKU	Yes	No	No	No
V_PACKSKU_QTY	Yes	No	No	No
ITEM_SUPP_COUNTRY	Yes	No	No	No
ORDLOC_EXP	Yes	No	No	No
ALC_HEAD	Yes	No	Yes	No
ALC_COMP_LOC	Yes	No	Yes	No

Delete Obligation

Business Overview

This service deletes existing obligation records with pending status by calling the SVCPROV_OBLIGATION package and then calling the core obligation package to validate and delete entire obligation records from Merchandising tables.

This service accepts an obligation key, or supplier/invoice, or partner type/partner/invoice combination to identify the obligation. This service will delete the entire obligation record.

Service Type

Post

ReST URL

/ObligationUpload/deleteObligation

Input Parameters

RestObligationRecRDO

Parameter Name	Data Type
obligationKey	BigDecimal
obligationLevel	String
keyValue1	String
keyValue2	String
keyValue3	String
keyValue4	String

Parameter Name	Data Type
keyValue5	String
keyValue6	String
status	String
partnerType	String
partnerId	String
supplierSite	String
invoice	String
invoiceDate	Timestamp
qty	BigDecimal
qtyUom	String
currency	String
exchangeRate	BigDecimal
checkAuthNo	String
paidAmt	BigDecimal
paidDate	Timestamp
comments	String
obligationCompTbl	List<RestObligationCompRecRDO>

RestObligationCompRecRDO

Parameter Name	Data Type
compId	String
allocateToAlc	String
allocationType	String
allocationBasisUom	String
amt	BigDecimal
rate	BigDecimal
perCount	BigDecimal
perCountUom	String
obligationCompLocTbl	List<RestObligationCompLocRecRDO>

RestObligationCompLocRecRDO

Parameter Name	Data Type
action	String
locType	String
locId	BigDecimal
qty	BigDecimal
amt	BigDecimal

JSON Structure: The same RDO 'RestObligationRecRDO' will be used for deleteObligation but only the below parameters will be considered for the request. The rest of the parameters will be ignored.

```
[{
  "obligationKey": null,
  "partnerType": null,
  "partnerId": null,
  "supplierSite": null,
  "invoice": null }]
```

Output

RestObligationStatusRecRDO

Parameter Name	Data Type
successObligationTbl	List<RestObligationSuccessRecRDO>
successObligationCount	BigDecimal
failObligationTbl	List<RestObligationFailRecRDO>
failObligationCount	BigDecimal

RestObligationSuccessRecRDO

Parameter Name	Data Type
obligationKey	BigDecimal
status	String
invoice	String

RestObligationFailRecRDO

Parameter Name	Data Type
obligationKey	BigDecimal
status	String
invoice	String
errorMessage	String

The output will contain the status of the request, including validation errors, if any.

For Success:

```
{
  "successObligationTbl": [
    {
      "status": null,
      "invoice": null,
      "obligationKey": null,
      "links": [],
      "hyperMediaContent": {
        "linkRDO": []
      }
    }
  ],
  "successObligationCount": null,
```

```

    "failObligationTbl": [],
    "failObligationCount": null,
    "links": [],
    "hyperMediaContent": {
        "linkRDO": []
    }
}

```

For Failure:

```

{
    "successObligationTbl": [],
    "successObligationCount": null,
    "failObligationTbl": [
        {
            "errorMessage": null,
            "status": null,
            "invoice": null,
            "obligationKey": null,
            "links": [],
            "hyperMediaContent": {
                "linkRDO": []
            }
        }
    ],
    "failObligationCount": null,
    "links": [],
    "hyperMediaContent": {
        "linkRDO": []
    }
}

```

Table Impact

TABLE	SELECT	INSERT	UPDATE	DELETE
OBLIGATION	Yes	No	No	Yes
OBLIGATION_COMP	Yes	No	No	Yes
OBLIGATION_COMP_LOC	Yes	No	No	Yes
ALC_HEAD	Yes	No	No	Yes
ALC_COMP_LOC	Yes	No	No	Yes

Delete Obligation Component

Business Overview

This service deletes obligation components for existing obligations by calling the SVCPROV_OBLIGATION package and then calling the core obligation package to validate and delete data from Merchandising tables.

This service accepts obligation key, or supplier/invoice, or partner type/partner/invoice combination to identify the obligation. Pass the component ID which the user wants to delete. This service will delete the component. If locations are attached to the components, then that will be deleted as well. Reallocation to ALC will be done after deletion.

Service Type

Post

ReST URL

/ObligationUpload/deleteObligationComp

Input Parameters

RestObligationRecRDO

Parameter Name	Data Type
obligationKey	BigDecimal
obligationLevel	String
keyValue1	String
keyValue2	String
keyValue3	String
keyValue4	String
keyValue5	String
keyValue6	String
status	String
partnerType	String
partnerId	String
supplierSite	String
invoice	String
invoiceDate	Timestamp
qty	BigDecimal
qtyUom	String
currency	String
exchangeRate	BigDecimal
checkAuthNo	String
paidAmt	BigDecimal
paidDate	Timestamp
comments	String
obligationCompTbl	List<RestObligationCompRecRDO>

RestObligationCompRecRDO

Parameter Name	Data Type
compId	String
allocateToAlc	String
allocationType	String
allocationBasisUom	String
amt	BigDecimal

Parameter Name	Data Type
rate	BigDecimal
perCount	BigDecimal
perCountUom	String
obligationCompLocTbl	List<RestObligationCompLocRecRDO>

RestObligationCompLocRecRDO

Parameter Name	Data Type
action	String
locType	String
locId	BigDecimal
qty	BigDecimal
amt	BigDecimal

JSON Structure: The same RDO 'RestObligationRecRDO' will be used for deleteObligationComp, but only the below parameters will be considered for the request. The rest of the parameters will be ignored.

```
[{
  "obligationKey": null,
  "partnerType":null,
  "partnerId":null,
  "supplierSite":null,
  "invoice":null,
  "obligationCompTbl" : [{
    "compId":null
  }]
}]
```

Output

RestObligationStatusRecRDO

Parameter Name	Data Type
successObligationTbl	List<RestObligationSuccessRecRDO>
successObligationCount	BigDecimal
failObligationTbl	List<RestObligationFailRecRDO>
failObligationCount	BigDecimal

RestObligationSuccessRecRDO

Parameter Name	Data Type
obligationKey	BigDecimal
status	String
invoice	String

RestObligationFailRecRDO

Parameter Name	Data Type
obligationKey	BigDecimal
status	String
invoice	String
errorMessage	String

The output will contain the status of the request, including validation errors, if any.

For Success:

```
{
  "successObligationTbl": [
    {
      "status": null,
      "invoice": null,
      "obligationKey": null,
      "links": [],
      "hyperMediaContent": {
        "linkRDO": []
      }
    }
  ],
  "successObligationCount": null,
  "failObligationTbl": [],
  "failObligationCount": null,
  "links": [],
  "hyperMediaContent": {
    "linkRDO": []
  }
}
```

For Failure:

```
{
  "successObligationTbl": [],
  "successObligationCount": null,
  "failObligationTbl": [
    {
      "errorMessage": null,
      "status": null,
      "invoice": null,
      "obligationKey": null,
      "links": [],
      "hyperMediaContent": {
        "linkRDO": []
      }
    }
  ],
  "failObligationCount": null,
  "links": [],
  "hyperMediaContent": {
    "linkRDO": []
  }
}
```

Table Impact

TABLE	SELECT	INSERT	UPDATE	DELETE
OBLIGATION	Yes	No	No	Yes
OBLIGATION_COMP	Yes	No	No	Yes
OBLIGATION_COMP_LOC	Yes	No	No	Yes
ORDHEAD	Yes	No	No	Yes
ORDSKU	Yes	No	No	Yes
ORDLOC	Yes	No	No	Yes
SHIPMENT	Yes	No	No	Yes
SHIPSKU	Yes	No	No	Yes
V_PACKSKU_QTY	Yes	No	No	Yes
ITEM_SUPP_COUNTRY	Yes	No	No	Yes
ORDLOC_EXP	Yes	No	No	Yes
ALC_HEAD	Yes	No	Yes	Yes
ALC_COMP_LOC	Yes	No	Yes	Yes

Purchase Order Detail Service

This section describes the Purchase Order Detail Service.

Business Overview

Purchase Order Detail service allows user to retrieve purchase order information for a selected order.

Service Type

Get

ReST URL

Po/poDetail?orderNumber={orderNumber}

Input Parameters

Parameter Name	Required	Description
Order Number	Yes	Order Number

Output

RestPoRecRDO

Parameter Name	Data Type
orderNumber	BigDecimal

Parameter Name	Data Type
orderType	String
group	BigDecimal
division	BigDecimal
dept	BigDecimal
buyer	BigDecimal
supplier	BigDecimal
supplierStatus	String
locationType	String
location	BigDecimal
writtenDate	Date
notBeforeDate	Date
notAfterDate	Date
otbEndofWeekDate	Date
earliestShipDate	Date
latestShipDate	Date
closeDate	Date
terms	String
freightTerms	String
originIndicator	BigDecimal
shipmentMethod	String
purchaseType	String
status	String
currencyCode	String
masterPurchaseOrderNumber	BigDecimal
polItemTbl	List<RestPolItemRecRDO>

RestPolItemRecRDO

Parameter Name	Data Type
item	String
refernceltem	String
packItem	String
originCountryId	String
earliestShipDate	Date
latestShipDate	Date
supplierPackSize	BigDecimal
location	BigDecimal
locationType	String
physicalWarehouse	BigDecimal
unitRetail	BigDecimal

Parameter Name	Data Type
quantityOrdered	BigDecimal
quantityPrescaled	BigDecimal
quantityReceived	BigDecimal
lastReceivedQuantity	BigDecimal
lastRoundQuantity	BigDecimal
lastGroupRoundedQunatity	BigDecimal
quantityCancelled	BigDecimal
cancelCode	String
cancelDate	Date
unitCost	BigDecimal
costSource	String
nonScaleIndicator	String
estimatedStockDate	Date
restPoltemExpTbl	List<RestPoltemExpRecRDO>

RestPoltemExpRecRDO

Parameter Name	Data Type
item	String
packItem	String
location	BigDecimal
locationType	String
componentId	String
componentDecsipation	String
alwaysDefaultIndicator	String
componentRate	BigDecimal
componentCurrency	String
exchangeRate	BigDecimal
estimatedExpenValue	BigDecimal

JSON Structure:

```
{
  "orderNumber": null,
  "orderType": null,
  "group": null,
  "division": null,
  "dept": null,
  "buyer": null,
  "supplier": null,
  "supplierStatus": null,
  "locationType": null,
  "location": null,
  "writtenDate": null,
  "notBeforeDate": null,
  "notAfterDate": null,
}
```

```
"otbEndOfWeekDate": null,
"earliestShipDate": null,
"latestShipDate": null,
"closeDate": null,
"terms": null,
"freightTerms": null,
"originIndicator": null,
"shipmentmethod": null,
"purchaseType": null,
"status": null,
"currencyCode": null,
"masterPurchaseOrderNumber": null,
"poItemTbl": [
  {
    "item": null,
    "referenceItem": null,
    "originCountryId": null,
    "earliestShipDate": null,
    "latestShipDate": null,
    "supplierPackSize": null,
    "location": null,
    "locationType": null,
    "physicalWarehouse": null,
    "unitRetail": null,
    "quantityOrdered": null,
    "quantityPrescaled": null,
    "quantityReceived": null,
    "lastReceivedQuantity": null,
    "lastRoundQuantity": null,
    "lastGroupRoundedQuantity": null,
    "quantityCancelled": null,
    "cancelCode": null,
    "unitCost": null,
    "costSource": null,
    "nonScaleIndicator": null,
    "estimatedStockDate": null,
    "poItemExpTbl": [
      {
        "item": null,
        "packItem": null,
        "location": null,
        "locationType": null,
        "componentId": null,
        "componentDescription": null,
        "alwaysDefaultIndicator": null,
        "componentRate": null,
        "componentCurrency": null,
        "exchangeRate": null,
        "estimatedExpenValue": null,
        "links": [],
        "hyperMediaContent": {
          "linkRDO": []
        }
      }
    ]
  },
  "links": [],
  "hyperMediaContent": {
    "linkRDO": []
  }
}
"links": [],
```

```

    "hyperMediaContent": {
      "linkRDO": []
    }
  }
}

```

Table Impact

TABLE	SELECT	INSERT	UPDATE	DELETE
ORDHEAD	Yes	No	No	No
ORDLOC	Yes	No	No	No
ORDSKU	Yes	No	No	No
ORDLOC_EXPENSES	Yes	No	No	No
V_DEPS	Yes	No	No	No
SUPS	Yes	No	No	No
WH	Yes	No	No	No
ELC_COMP	Yes	No	No	No
JOB_AUDIT_LOGS	No	Yes	No	No
JOB_AUDIT_PAYLOAD	No	Yes	No	No

Recent Inventory Transfer Services

This section describes the Recent Inventory Transfer services.

Functional Area

Inventory Movement

Business Overview

The primary role of these services is to approve or reject Merchandising's transfers.

Transfer Location Search

This section describes the transfer location search service.

Business Overview

The web service enables location search applicable for Transfers. Locations can be searched by either 'S'tore or 'W'arehouse, with the subsequent entry of a location number, a partial location number, a location description, or a partial location description in the search string.

The locations returned are constrained by the following criteria:

- When search type is warehouse then:
 - Internal finishers are filtered out
- When search type is store then:
 - Only company stores are returned

- Only stockholding stores are returned

Service Type

Get

ReST URL

/Transfer/recent/transferLocSearch?
searchString={searchString}&locType={locType}&pageSize={pageSize}&pageNumber={page
Number}

Input Parameters

Parameter Name	Required	Description	Valid values
SearchString	No	search string for locations Id or Name	NA
LocType	No	Location type Store or warehouse	S, W
PageSize	No	Maximum number of locations to retrieve per page	NA
PageNumber	No	Result page to retrieve	NA

Output

PagedResultsRDO

Parameter Name	Data Type
totalRecordCount	BigDecimal
results	List<RtsfLocSearchResultRDO>

RtsfLocSearchResultRDO

Parameter Name	Data Type
location	BigDecimal
locationType	String
locationName	String
currency	String

JSON Structure:

```
{
  "totalRecordCount": 1,
  "results": [
    {
      "location": null,
      "locationType": null,
      "locationName": null,
      "currency": null,
      "links": [ ],
      "hyperMediaContent": {
```

```

        "linkRDO": [ ]
      }
    },
    "links": [ ],
    "hyperMediaContent": {
      "linkRDO": [ ]
    }
  }
}

```

Table Impact

TABLE	SELECT	INSERT	UPDATE	DELETE
V_STORE	Yes	No	No	No
V_WH	Yes	No	No	No
JOB_AUDIT_LOGS	No	Yes	No	No
JOB_AUDIT_PAYLOAD	No	Yes	No	No

Transfer Location Load

This section describes the Transfer Location Load service.

Business Overview

This web service allows the user to refresh already selected Transfer locations records.

Service Type

Get

ReST URL

/Transfer/recent/transferLocationLoad?locations={locations}

Input Parameters

Parameter Name	Required	Description
Locations	No	Comma Separated values for selected locations' ID

Output

RtsfLocSearchResultRDO

Parameter Name	Data Type
location	BigDecimal
locationType	String
locationName	String
currency	String

JSON Structure:

```
[
  {
    "location": null,
    "locationType": null,
    "locationName": null,
    "currency": null,
    "links": [ ],
    "hyperMediaContent": {
      "linkRDO": [ ]
    }
  }
]
```

Table Impact

TABLE	SELECT	INSERT	UPDATE	DELETE
V_STORE	Yes	No	No	No
V_WH	Yes	No	No	No
JOB_AUDIT_LOGS	No	Yes	No	No
JOB_AUDIT_PAYLOAD	No	Yes	No	No

Transfer Status List

This section describes the Transfer Status List service.

Business Overview

Retrieves all valid transfer statuses.

Service Type

Get

ReST URL

/ Transfer/recent/ transferStatusList

Input Parameters

No input.

Output

CodeDetailRDO

Parameter Name	Data Type
code	String
codeDescription	String
codeSequence	BigDecimal

```

JSON Structure:
[
  {
    "code": null,
    "codeDescription": null,
    "codeSequence": null,
    "links": [ ],
    "hyperMediaContent": {
      "linkRDO": [ ]
    }
  }
]

```

Table Impact

TABLE	SELECT	INSERT	UPDATE	DELETE
CODE_HEAD	Yes	No	No	No
CODE_DETAIL	Yes	No	No	No
JOB_AUDIT_LOGS	No	Yes	No	No
JOB_AUDIT_PAYLOAD	No	Yes	No	No

Transfer Type List

This section describes the Transfer Type List service.

Business Overview

Retrieves all valid transfer types.

Service Type

Get

ReST URL

/ Transfer/recent/transferTypeList

Input Parameters

No input.

Output

CodeDetailRDO

Parameter Name	Data Type
code	String
codeDescription	String
codeSequence	BigDecimal

```

JSON Structure:
[

```

```

    {
      "code": null,
      "codeDescription": null,
      "codeSequence": null,
      "links": [ ],
      "hyperMediaContent": {
        "linkRDO": [ ]
      }
    }
  ]

```

Table Impact

TABLE	SELECT	INSERT	UPDATE	DELETE
CODE_HEAD	Yes	No	No	No
CODE_DETAIL	Yes	No	No	No
JOB_AUDIT_LOGS	No	Yes	No	No
JOB_AUDIT_PAYLOAD	No	Yes	No	No

Search Transfer User IDs

This section describes the Search Transfer User IDs.

Business Overview

The Search Transfer User IDs service retrieves for all User IDs that created transfers.

Service Type

Get

ReST URL

/Transfer/recent/searchUserIds?
searchString={searchString}&pageSize={pageSize}&pageNumber={pageNumber}

Input Parameters

Parameter Name	Required	Description
SearchString	Yes	search string for User Id
PageSize	No	Maximum number of transfer user IDs to retrieve per page
PageNumber	No	Result page to retrieve

Output

PagedResultsRDO

Parameter Name	Data Type
totalRecordCount	BigDecimal

Parameter Name	Data Type
results	List<VarcharIdRDO>

VarcharIdRDO

Parameter Name	Data Type
id	String

JSON Structure:

```
{
  "totalRecordCount": null,
  "results": [
    {
      "id": null,
      "links": [ ],
      "hyperMediaContent": {
        "linkRDO": [ ]
      }
    }
  ],
  "links": [ ],
  "hyperMediaContent": {
    "linkRDO": [ ]
  }
}
```

Table Impact

TABLE	SELECT	INSERT	UPDATE	DELETE
V_TSFHEAD	Yes	No	No	No
JOB_AUDIT_LOGS	No	Yes	No	No
JOB_AUDIT_PAYLOAD	No	Yes	No	No

Transfer Search

This section describes the Transfer Search service.

Business Overview

The web services in this area enables search for applicable transfers. Transfers can be searched by their status, transfer types, transfer number, create date, delivery date, create ID, item department and/or locations. The transfers returned are constrained by the following criteria:

- Customer Orders and Book Transfers are filtered out.
- Only Transfers with transfer details are returned.

Service Type

Get

ReST URL

```
/Transfer/recent/transferSearch?
statuses={statuses}&transferTypes={transferTypes}&createIds={createIds}&startCreateDate
={startCreateDate}&endCreateDate={endCreateDate}&startDeliveryDate={startDeliveryDate
}&endDeliveryDate={endDeliveryDate}&transferNumber={transferNumber}&locations={locatio
ns}&departments={departments}&pageSize={pageSize}&pageNumber={pageNumber}
```

Input Parameters

Parameter Name	Required	Description	Valid values
Statuses	No	Comma Separated values for selected transfer statuses	
TransferTypes	No	Comma Separated values for selected transfer types	
CreateIds	No	Comma Separated values for selected transfer create ID	
StartCreateDate	No	Start of the range of transfer create dates	
EndCreateDate	No	End of the range of transfer create dates	
StartDeliveryDate	No	Start of the range of transfer create dates	
EndDeliveryDate	No	End of the range of transfer create dates	
TransferNumber	No	Transfer Number	
Locations	No	Comma Separated values for selected Location IDs	
Departments	No	Comma Separated values for selected Department IDs	
PageSize	No	Maximum number of locations to retrieve per page	
PageNumber	No	Result page to retrieve	

Output

PagedResultsRDO

Parameter Name	Data Type
totalRecordCount	BigDecimal
results	List<RtsfSearchResRDO>

RtsfSearchResRDO

Parameter Name	Data Type
transferNumber	BigDecimal
tsfType	String
fromLocation	BigDecimal
fromLocationType	String
fromLocationName	String

Parameter Name	Data Type
toLocation	BigDecimal
toLocationType	String
toLocationName	String
status	String
totalCost	BigDecimal
currency	String
deliveryDate	Long

JSON Structure:

```
{
  "totalRecordCount": null,
  "results": [
    {
      "transferNumber": null,
      "tsfType": null,
      "fromLocation": null,
      "fromLocationType": null,
      "fromLocationName": null,
      "toLocation": null,
      "toLocationType": null,
      "toLocationName": null,
      "status": null,
      "totalCost": null,
      "currency": null,
      "deliveryDate": null,
      "links": [ ],
      "hyperMediaContent": {
        "linkRDO": [ ]
      }
    }
  ],
  "links": [ ],
  "hyperMediaContent": {
    "linkRDO": [ ]
  }
}
```

Table Impact

TABLE	SELECT	INSERT	UPDATE	DELETE
ITEM_LOC_SOH	Yes	No	No	No
ITEM_MASTER	Yes	No	No	No
TSFDETAIL	Yes	No	No	No
TSFITEM_INV_FLOW	Yes	No	No	No
V_STORE	Yes	No	No	No
V_TSFDETAIL	Yes	No	No	No
V_TSFHEAD	Yes	No	No	No
V_WH	Yes	No	No	No

TABLE	SELECT	INSERT	UPDATE	DELETE
JOB_AUDIT_LOGS	No	Yes	No	No
JOB_AUDIT_PAYLOAD	No	Yes	No	No

Get Transfer Detail

This section describes the Get Transfer Detail service.

Business Overview

Get Transfer Detail service allow user to retrieve Transfer information for a selected transfer number.

Service Type

Get

ReST URL

/Transfer/recent/transferDetail?
transferNumber={transferNumber}&pageSize={pageSize}&pageNumber={pageNumber}

Input Parameters

Parameter Name	Required	Description
TransferNumber	Yes	Transfer Number ID
PageSize	No	Maximum number of items to retrieve per page
PageNumber	No	Result page to retrieve

Output

PagedResultsRDO

Parameter Name	Data Type
totalRecordCount	BigDecimal
results	List<RtsfTsfDtIRDO>

RtsfTsfDtIRDO

Parameter Name	Data Type
transferNumber	BigDecimal
status	String
fromLocation	BigDecimal
fromLocationName	String
finisher	BigDecimal

Parameter Name	Data Type
finisherName	String
toLocation	BigDecimal
toLocationName	String
transferType	String
totalCost	BigDecimal
totalRetail	BigDecimal
currency	String
deliveryDate	Long
createId	String
createDate	Long
transferItemsTable	List<RtsfTsfDtlItemRDO>

RtsfTsfDtlItemRDO

Parameter Name	Data Type
item	String
itemDescription	String
transferQuantity	BigDecimal

JSON Structure:

```
{
  "totalRecordCount": null,
  "results": [
    {
      "transferNumber": null,
      "status": null,
      "fromLocation": null,
      "fromLocationName": null,
      "finisher": null,
      "finisherName": null,
      "toLocation": null,
      "toLocationName": null,
      "transferType": null,
      "totalCost": null,
      "totalRetail": null,
      "currency": null,
      "deliveryDate": null,
      "createId": null,
      "createDate": null,
      "transferItemsTable": [
        {
          "item": null,
          "itemDescription": null,
          "transferQuantity": null,
          "links": [ ],
          "hyperMediaContent": {
            "linkRDO": [ ]
          }
        }
      ]
    }
  ],
}
```



```

        "links": [ ],
        "hyperMediaContent": {
            "linkRDO": [ ]
        }
    },
    "links": [ ],
    "hyperMediaContent": {
        "linkRDO": [ ]
    }
}

```

Table Impact

TABLE	SELECT	INSERT	UPDATE	DELETE
ITEM_LOC	Yes	No	No	No
ITEM_LOC_SOH	Yes	No	No	No
TSF_ITEM_INV_FLOW	Yes	No	No	No
V_EXTERNAL_FINISHES	Yes	No	No	No
V_INTERNAL_FINISHES	Yes	No	No	No
V_ITEM_MASTER	Yes	No	No	No
V_LOCATION	Yes	No	No	No
V_STORE	Yes	No	No	No
V_TSFDETAIL	Yes	No	No	No
V_TSFHEAD	Yes	No	No	No
V_WH	Yes	No	No	No
JOB_AUDIT_LOGS	No	Yes	No	No
JOB_AUDIT_PAYLOAD	No	Yes	No	No

Update Transfer Status

This section describes the Update Transfer Status service.

Business Overview

The web service approves or unapproves a transfer or a list of transfers.

Service Type

Post

ReST URL

/Transfer/recent/updateTransferStatus?
newStatus={newStatus}&transferNumbers={transferNumbers}

Input Parameters

Parameter Name	Required	Description	Valid values
NewStatus	Yes	New status of the transfer. May only be 'A'pproved or 'I'nput.	A, I
TransferNumbers	Yes	Comma Separated values for selected locations' ID	

Output

NA

Table Impact

TABLE	SELECT	INSERT	UPDATE	DELETE
TSFHEAD	Yes	No	Yes	No
TSFDETAIL	Yes	Yes	Yes	Yes
ITEM_LOC	Yes	No	No	No
ITEM_LOC_SOH	Yes	No	Yes	No
ITEM_MASTER	Yes	No	No	No
PACKITEM_BREAKOUT	Yes	No	No	No
STORE	Yes	No	No	No
SYSTEM_OPTIONS	Yes	No	No	No
V_TSFHEAD	Yes	No	No	No
WH	Yes	No	No	No
JOB_AUDIT_LOGS	No	Yes	No	No
JOB_AUDIT_PAYLOAD	No	Yes	No	No

Recent Purchase Order Services

This section describes the Recent Purchase Order services.

Functional Area

Procurement

Business Overview

The primary role of this service is to approve, reject, or cancel Merchandising's purchase orders.

Cancel Reason Code List

This section describes the Cancel Reason Code List service.

Business Overview

Retrieves all purchase order cancel reason codes.

Service Type

Get

ReST URL

/PurchaseOrders/recent/cancelReasonCodeList

Input Parameters

No input.

Output

CodeDetailRDO

Parameter Name	Data Type
code	String
codeDescription	String
codeSequence	BigDecimal

JSON Structure:

```
[
  {
    "code": null,
    "codeDescription": null,
    "codeSequence": null,
    "links": [ ],
    "hyperMediaContent": {
      "linkRDO": [ ]
    }
  }
]
```

Table Impact

TABLE	SELECT	INSERT	UPDATE	DELETE
CODE_HEAD	Yes	No	No	No
CODE_DETAIL	Yes	No	No	No
JOB_AUDIT_LOGS	No	Yes	No	No
JOB_AUDIT_PAYLOAD	No	Yes	No	No

Origin Code List

This section describes the Origin Code List service.

Business Overview

Retrieves all purchase order origin codes.

Service Type

Get

ReST URL

/PurchaseOrders/recent/originCodeList

Input Parameters

No input.

Output

CodeDetailRDO

Parameter Name	Data Type
code	String
codeDescription	String
codeSequence	BigDecimal

JSON Structure:

```
[
  {
    "code": null,
    "codeDescription": null,
    "codeSequence": null,
    "links": [ ],
    "hyperMediaContent": {
      "linkRDO": [ ]
    }
  }
]
```

Table Impact

TABLE	SELECT	INSERT	UPDATE	DELETE
CODE_HEAD	Yes	No	No	No
CODE_DETAIL	Yes	No	No	No
JOB_AUDIT_LOGS	No	Yes	No	No
JOB_AUDIT_PAYLOAD	No	Yes	No	No

Purchase Order Status List

This section describes the Purchase Order Status list.

Business Overview

Retrieves all valid purchase order statuses.

Service Type

Get

ReST URL

/PurchaseOrders/recent/purchaseOrderStatusList

Input Parameters

No input.

Output

CodeDetailRDO

Parameter Name	Data Type
code	String
codeDescription	String
codeSequence	BigDecimal

JSON Structure:

```
[
  {
    "code": null,
    "codeDescription": null,
    "codeSequence": null,
    "links": [ ],
    "hyperMediaContent": {
      "linkRDO": [ ]
    }
  }
]
```

Table Impact

TABLE	SELECT	INSERT	UPDATE	DELETE
CODE_HEAD	Yes	No	No	No
CODE_DETAIL	Yes	No	No	No
JOB_AUDIT_LOGS	No	Yes	No	No
JOB_AUDIT_PAYLOAD	No	Yes	No	No

Search Purchase Order User ID

This section describes the Search Purchase Order User ID.

Business Overview

This service retrieves a list of user IDs associated with creating a purchase order.

Service Type

Get

ReST URL

/PurchaseOrders/recent/searchUserIds?
searchString={searchString}&pageSize={pageSize}&pageNumber={pageNumber}

Input Parameters

- Search String - Required
- Page Size - Optional
- Page Number - Optional

Output

VarcharIdRDO

Parameter Name	Data Type
id	String

PagedResultsRDO

Parameter Name	Data Type
totalRecordCount	BigDecimal
Next Page URL	String
Previous Page URL	String

JSON Structure:

```
{
  "totalRecordCount": null,
  "results": [
    {
      "id": null,
      "links": [ ],
      "hyperMediaContent": {
        "linkRDO": [ ]
      }
    }
  ],
  "links": [ ],
  "hyperMediaContent": {
    "linkRDO": [ ]
  }
}
```

Table Impact

TABLE	SELECT	INSERT	UPDATE	DELETE
V_ORDHEAD	Yes	No	No	No
JOB_AUDIT_LOGS	No	Yes	No	No
JOB_AUDIT_PAYLOAD	No	Yes	No	No

Purchase Order Search

This section describes the Purchase Order Search service.

Business Overview

This service retrieves summary information on all none closed purchase orders that match input criteria.

Service Type

Get

ReST URL

/PurchaseOrders/recent/purchaseOrderSearch?
 statuses={statuses}&createIds={createIds}&startCreateDate={startCreateDate}&endCreateDate={endCreateDate}&orderNumber={orderNumber}&suppliers={suppliers}&originCodes={originCodes}&departments={departments}&pageSize={pageSize}&pageNumber={pageNumber}

Input Parameters

Parameter Name	Required	Description	Valid values
statuses	No	List of order status	A, S, W
createIds	No	List of user IDs who created the PO	
startCreateDate	No	Long format date for starting period	
endCreateDate	No	Long format date for end period	
orderNumber	No	Order number to retrieve	
suppliers	No	List of order suppliers	
originCodes	No	List of valid Origin codes	
departments	No	List of valid order/item departments	
pageSize		Maximum number of orders to retrieve per page	
pageNumber		Result page to retrieve	

Output

RpoSearchResRDO

Parameter Name	Data Type
orderNumber	BigDecimal
status	String
supplier	BigDecimal
supplierName	String
notBeforeDate	Long
notAfterDate	Long
totalCost	BigDecimal
currency	String
previouslyApprovedIndicator	String
editableIndicator	String

PagedResultsRDO

Parameter Name	Data Type
totalRecordCount	BigDecimal
Next Page URL	String
Previous Page URL	String

JSON Structure:

```
"{
  "type": "paginationRDO",
  "totalRecordCount": 252,
  "hyperMediaContent": {},
  "links": [],
  "results": [{
    "orderNumber": 12453253,
    "statusId" : "W",
    "supplierId": 124121,
    "supplierName": "Some Supplier Site",
    "notBeforeDate": 35235252,
    "notAfterDate": 325235252351,
    "totalCost": 243.231,
    "currencyCode": "USD"
  }]
}"
```

Table Impact

TABLE	SELECT	INSERT	UPDATE	DELETE
ITEM_MASTER	Yes	No	No	No
PRODUCT_CONFIG_OPTIONS	Yes	No	No	No
V_ORDHEAD	Yes	No	No	No
V_ORDSKU	Yes	No	No	No
V_SUPS	Yes	No	No	No
JOB_AUDIT_LOGS	No	Yes	No	No
JOB_AUDIT_PAYLOAD	No	Yes	No	No

Get Purchase Order Summary

This section describes the Get Purchase Order Summary service.

Business Overview

This service retrieves purchase order header detail with open to buy information.

Service Type

Get

ReST URL

/PurchaseOrders/recent/PurchaseOrderSummary?orderNumber={orderNumber}

Input Parameters

Order Number-Required

Output

RpoOrderSumRDO

Parameter Name	Data Type
orderNumber	BigDecimal
status	String
supplier	BigDecimal
supplierName	String
notBeforeDate	Long
notAfterDate	Long
otbEowDate	Long
terms	String
termsCode	String
termsDescription	String
totalCost	BigDecimal
totalRetail	BigDecimal
Currency	String
createld	String
writtenDate	Long
defaultDisplayLevel	String
previouslyApprovedIndicator	String
editableIndicator	String
otbTable	List<RpoOrderSumOtbRDO>

RpoOrderSumOtbRDO

Parameter Name	Data Type
department	BigDecimal
classId	BigDecimal
subclassId	BigDecimal
subclassName	String
orderAmount	BigDecimal
budgetAmount	BigDecimal
receivedAmount	BigDecimal
approvedAmount	BigDecimal
outstandingAmount	BigDecimal

JSON Structure:

```
{
  "orderNumber":12345,
  "statusId":"W",
  "supplierId":12345,
  "supplierName": "Supplier 12345",
  "notBeforeDate": 1234567,
  "notAfterDate": 236573,
  "terms":"01",
  "termsCode":"01234",
  "termsDescription":"Letter Of Credit",
  "totalCost": 123.45,
  "totalRetail": 456.78,
  "currencyCode": "CAD",
  "createdBy": "BUYER",
  "writtenDate": 1234567,
  "otbResults":
  [{
    "department" : 12345,
    "classId": 12345,
    "subClassId" : 12345,
    "subClassName": "subClassName"
    "budgetAmount": 12345.545,
    "orderAmount": 12345.545,
    "receivedAmount": 12345.545,
    "approvedAmount": 12345.545
  }]
}
```

Table Impact

TABLE	SELECT	INSERT	UPDATE	DELETE
COST_ZONE_GROUP_LOC	Yes	No	No	No
COUNTRY_ATTRIB	Yes	No	No	No
DEPS	Yes	No	No	No
ELC_COMP	Yes	No	No	No
ITEM_COST_HEAD	Yes	No	No	No
ITEM_EXP_DETAIL	Yes	No	No	No
ITEM_EXP_HEAD	Yes	No	No	No

TABLE	SELECT	INSERT	UPDATE	DELETE
ITEM_LOC	Yes	No	No	No
ITEM_MASTER	Yes	No	No	No
ITEM_SUPP_COUNTRY	Yes	No	No	No
MV_CURRENCY_CONVERSION_RATES	Yes	No	No	No
ORDHEAD	Yes	No	No	No
ORDLOC	Yes	No	No	No
ORDLOC_EXP	Yes	No	No	No
ORDSKU	Yes	No	No	No
ORDSKU HTS	Yes	No	No	No
ORDSKU HTS ASSESS	Yes	No	No	No
OTB	Yes	No	No	No
PERIOD	Yes	No	No	No
PRODUCT_CONFIG_OPTIONS	Yes	No	No	No
STORE	Yes	No	No	No
SUPS	Yes	No	No	No
V_ORDHEAD	Yes	No	No	No
V_PACKSKU_QTY	Yes	No	No	No
V_SUBCLASS_TL	Yes	No	No	No
V_SUPS	Yes	No	No	No
V_TERMS_HEAD_TL	Yes	No	No	No
WH	Yes	No	No	No
JOB_AUDIT_LOGS	No	Yes	No	No
JOB_AUDIT_PAYLOAD	No	Yes	No	No

Get Purchase Order Items

This section describes the Get Purchase Order Items service.

Business Overview

This service retrieves items details for an order number. Based on the display level, the items record aggregates to the level specified when applicable.

Service Type

Get

ReST URL

```
/PurchaseOrders/recent/PurchaseOrderItems?
orderNumber={orderNumber}&itemDisplayLevel={itemDisplayLevel}&pageSize={pageSize}&
pageNumber={pageNumber}
```

Input Parameters

Order Number - Required

Item Display Level - Optional - valid values PARENT_LEVEL, PARENT_DIFF_LEVEL, or TRAN_LEVEL

Page Size - Optional

Page Number - Optional

Output

RpoOrderSumItemRDO

Parameter Name	Data Type
item	String
ItemDescription	String
diff1	String
diff1Description	String
diff2	String
diff2Description	String
diff3	String
diff3Description	String
diff4	String
diff4Description	String
quantityOrdered	BigDecimal
totalCost	BigDecimal
currency	String
itemImageUrl	String

PagedResultsRDO

Parameter Name	Data Type
totalRecordCount	BigDecimal
Next Page URL	String
Previous Page URL	String

JSON Structure:

```
{
  "type": "paginationRDO",
  "totalRecordCount": 252,
  "hyperMediaContent": {},
  "links": [],
  "orderNumber": 1212131,
  "results": [{
    "itemId": 1234
    "itemDescription": "some item",
    "firstDiffId": 123424,
```

```

    "firstDiffDescription": "desc",
    "secondDiffId": 12345
    "secondDiffDescription" : "desc",
    "thirdDiffId": 1234324
    "thirdDiffDescription" : "desc",
    "fourthDiffId" : 1324,
    "fourthDiffDescription" : "desc",
    "quantityOrdered": 100,
    "totalCost" : 12345.353,
    "currencyCode": "USD",
    "itemImageUrl": "http://..."
  }
}
}

```

Table Impact

TABLE	SELECT	INSERT	UPDATE	DELETE
ITEM_IMAGE	Yes	No	No	No
ITEM_MASTER	Yes	No	No	No
ORDLOC	Yes	No	No	No
ORDSKU	Yes	No	No	No
ORDLOC_WKSHT	Yes	No	No	No
V_ITEM_MASTER	Yes	No	No	No
JOB_AUDIT_LOGS	No	Yes	No	No
JOB_AUDIT_PAYLOAD	No	Yes	No	No

Get Purchase Order Item Locations

This section describes the Get Purchase Order Item Locations service.

Business Overview

This service retrieves item location details for an order number. The location record aggregates based on the display level when applicable.

Service Type

Get

ReST URL

```

/PurchaseOrders/recent/PurchaseOrderItemLocations?
orderNumber={orderNumber}&item={item}&itemDisplayLevel={itemDisplayLevel}&diff1={diff1}
&diff2={diff2}&diff3={diff3}&diff4={diff4}&pageSize={pageSize}&pageNumber={pageNumber}

```

Input Parameters

Parameter Name	Required	Description	Valid values
orderNumber	Yes	Order number	
item	Yes	Item Id	

Parameter Name	Required	Description	Valid values
itemDisplayLevel	No	Item display level	PARENT_LEVEL PARENT_DIFF_LEVEL TRAN_LEVEL
diff1	No	Diff1 Id	
diff2	No	Diff2 Id	
diff3	No	Diff3 Id	
diff4	No	Diff4 Id	
pageSize	No	Maximum number of items to retrieve per page	
pageNumber	No	Result page to retrieve	

Output

RpoOrderItemLocRDO

Parameter Name	Data Type
location	BigDecimal
locationName	String
quantityOrdered	BigDecimal
totalCost	BigDecimal
currency	String

PagedResultsRDO

Parameter Name	Data Type
totalRecordCount	BigDecimal
Next Page URL	String
Previous Page URL	String

JSON Structure:

```
{
  "locations" : [
    {
      "locationId" : 12345,
      "locationName" : "some location",
      "orderedQuantity" : 1000,
      "totalCost" : 12345.234,
      "currencyCode" : "USD"
    },
    {
      "locationId" : 12345,
      "locationName" : "some location",
      "orderedQuantity" : 1000,
      "totalCost" : 12345.234,
      "currencyCode" : "USD"
    }
  ]
}
```

Table Impact

TABLE	SELECT	INSERT	UPDATE	DELETE
ITEM_MASTER	Yes	No	No	No
ORDLOC	Yes	No	No	No
V_STORE_TL	Yes	No	No	No
V_WH_TL	Yes	No	No	No
JOB_AUDIT_LOGS	No	Yes	No	No
JOB_AUDIT_PAYLOAD	No	Yes	No	No

Update Purchase Orders Date

This section describes the Update Purchase Orders Date service.

Business Overview

This service update list of purchase order dates. If no date is sent or sent as null then the assumption is there is no change on the current record date.

Service Type

Post

ReST URL

```
/PurchaseOrders/recent/updatePurchaseOrderDate?
notBeforeDate={notBeforeDate}&notAfterDate={notAfterDate}&otbEowDate={otbEowDate}&orderNumbers={orderNumbers}
```

Input Parameters

Order Numbers - Required - comma separated list

Not Before Date - Optional - in a long format

Not After Date - Optional - in a long format

OTB EWO Date - Optional - in a long format

Output

RpoStatusRDO

Parameter Name	Data Type
successOrdersCount	BigDecimal
successOrdersTable	List<BigDecimal>
failOrdersCount	BigDecimal
failOrdersTable	List<RpoFailRDO>

RpoFailRDO

Parameter Name	Data Type
orderNumber	BigDecimal
errorMessage	String

JSON Structure:

```
{
  "successOrdersCount": 0,
  "successOrdersTable": [],
  "failOrdersCount": 2,
  "failOrdersTable": [
    {
      "orderNumber": 123,
      "errorMessage": "Invalid Reason Code.",
      "links": [],
      "hyperMediaContent": {
        "linkRDO": []
      }
    },
    {
      "orderNumber": 987,
      "errorMessage": "Invalid Reason Code.",
      "links": [],
      "hyperMediaContent": {
        "linkRDO": []
      }
    }
  ],
  "links": [],
  "hyperMediaContent": {
    "linkRDO": []
  }
}
```

Table Impact

TABLE	SELECT	INSERT	UPDATE	DELETE
ALLOC_HEADER	No	No	Yes	No
CONTRACT_HEADER	Yes	No	No	No
DEAL_HEAD	Yes	No	Yes	No
ORDHEAD	Yes	No	Yes	No
OTB	No	No	Yes	No
SHIPMENT	Yes	No	No	No
SYSTEM_OPTIONS	Yes	No	No	No
JOB_AUDIT_LOGS	No	Yes	No	No
JOB_AUDIT_PAYLOAD	No	Yes	No	No

Cancel Purchase Orders

This section describes the Cancel Purchase Orders service.

Business Overview

This service cancels a list of purchase order.

Service Type

Post

ReST URL

/PurchaseOrders/recent/cancelPurchaseOrders?orderNumbers={orderNumbers}

Input Parameters

Order Number -Required-comma separated list

Output

RpoStatusRDO

Parameter Name	Data Type
successOrdersCount	BigDecimal
successOrdersTable	List<BigDecimal>
failOrdersCount	BigDecimal
failOrdersTable	List<RpoFailRDO>

RpoFailRDO

Parameter Name	Data Type
orderNumber	BigDecimal
errorMessage	String

JSON Structure:

```
{
  "successOrdersCount": 0,
  "successOrdersTable": [],
  "failOrdersCount": 2,
  "failOrdersTable": [
    {
      "orderNumber": 123,
      "errorMessage": "Invalid Reason Code.",
      "links": [],
      "hyperMediaContent": {
        "linkRDO": []
      }
    },
    {
      "orderNumber": 987,
      "errorMessage": "Invalid Order Number.",
      "links": [],
      "hyperMediaContent": {
        "linkRDO": []
      }
    }
  ]
}
```

```

    }
  ],
  "links": [],
  "hyperMediaContent": {
    "linkRDO": []
  }
}

```

Table Impact

TABLE	SELECT	INSERT	UPDATE	DELETE
ALLOC_DETAIL	Yes	No	Yes	No
ALLOC_HEADER	Yes	No	Yes	No
APPT_DETAIL	Yes	No	No	No
APPT_HEAD	Yes	No	No	No
CODE_DETAIL	Yes	No	No	No
DEAL_CALC_QUEUE	No	No	No	Yes
ORDHEAD	Yes	No	Yes	No
ORDLOC	Yes	No	Yes	No
OTB	No	No	Yes	No
SHIPMENT	Yes	No	Yes	No
SHIPSKU	Yes	No	Yes	No
SYSTEM_OPTIONS	Yes	No	No	No
WH	Yes	No	No	No
JOB_AUDIT_LOGS	No	Yes	No	No
JOB_AUDIT_PAYLOAD	No	Yes	No	No

Approve Purchase Orders

This section describes the Approve Purchase Orders service.

Business Overview

This service approves a list of purchase orders.

Service Type

Post

ReST URL

/PurchaseOrders/recent/cancelPurchaseOrders?orderNumbers={orderNumbers}

Input Parameters

Order Number -Required-comma separated list

Output

RpoStatusRDO

Parameter Name	Data Type
successOrdersCount	BigDecimal
successOrdersTable	List<BigDecimal>
failOrdersCount	BigDecimal
failOrdersTable	List<RpoFailRDO>

RpoFailRDO

Parameter Name	Data Type
orderNumber	BigDecimal
errorMessage	String

JSON Structure:

```
{
  "successOrdersCount": 0,
  "successOrdersTable": [],
  "failOrdersCount": 2,
  "failOrdersTable": [
    {
      "orderNumber": 123,
      "errorMessage": " Invalid Order Number.",
      "links": [],
      "hyperMediaContent": {
        "linkRDO": []
      }
    },
    {
      "orderNumber": 987,
      "errorMessage": "Invalid Order Number.",
      "links": [],
      "hyperMediaContent": {
        "linkRDO": []
      }
    }
  ],
  "links": [],
  "hyperMediaContent": {
    "linkRDO": []
  }
}
```

Table Impact

TABLE	SELECT	INSERT	UPDATE	DELETE
ALC_HEAD_TEMP	No	No	No	Yes
ALLOC_CHRG_TEMP	No	No	No	Yes
ALLOC_DETAIL	Yes	No	Yes	No

TABLE	SELECT	INSERT	UPDATE	DELETE
ALLOC_DETAIL_TEMP	No	No	No	Yes
ALLOC_HEADER	Yes	No	Yes	No
ALLOC_HEADER_TEMP	No	No	No	Yes
CONTRACT_COST_HIST	Yes	Yes	No	No
CONTRACT_DETAIL	Yes	No	Yes	No
CONTRACT_HEADER	Yes	No	Yes	No
DEAL_ACTUALS_FORECAST	No	No	No	Yes
DEAL_ACTUALS_ITEM_LOC	No	No	No	Yes
DEAL_COMP_PROM	No	No	No	Yes
DEAL_DETAIL	No	No	No	Yes
DEAL_HEAD	No	No	No	Yes
DEAL_HEAD_CFA_EXT	No	No	No	Yes
DEAL_ITEMLOC_DCS	No	No	No	Yes
DEAL_ITEMLOC_DIV_GRP	No	No	No	Yes
DEAL_ITEMLOC_ITEM	No	No	No	Yes
DEAL_ITEMLOC_PARENT_DIFF	No	No	No	Yes
DEAL_QUEUE	No	No	No	Yes
DEAL_THRESHOLD	No	No	No	Yes
DEAL_THRESHOLD_REV	No	No	No	Yes
DOC	Yes	No	No	No
DOC_LINK	Yes	No	No	No
ITEM_LOC	Yes	No	Yes	No
ITEM_MASTER	Yes	No	No	No
ITEM_SUPP_COUNTRY	Yes	No	No	No
ITEM_TICKET	Yes	No	No	No
LC_ACTIVITY	Yes	No	No	No
LC_AMENDMENTS	Yes	Yes	No	No
LC_DETAIL	Yes	Yes	No	No
LC_HEAD	Yes	No	Yes	No
LC_ORDAPPLY	No	Yes	No	Yes
ORD_INV_MGMT	Yes	No	No	Yes
ORD_LC_AMENDMENTS	Yes	No	No	No
ORDCUST	Yes	No	No	No
ORDCUST_DETAIL	Yes	Yes	No	Yes
ORDDIST_ITEM_TEMP	No	No	No	Yes
ORDHEAD	Yes	No	No	No
ORDHEAD_REV	No	Yes	No	No
ORDLC	Yes	No	Yes	No
ORDLOC	Yes	No	Yes	No

TABLE	SELECT	INSERT	UPDATE	DELETE
ORDLOC_DISCOUNT	No	No	No	Yes
ORDLOC_DISCOUNT_TEMP	No	No	No	Yes
ORDLOC_EXP_TEMP	No	No	No	Yes
ORDLOC_REV	No	Yes	No	No
ORDLOC_TEMP	No	No	No	Yes
ORDLOC_WKSHT	Yes	No	No	Yes
ORDSKU	Yes	No	No	No
ORDSKU HTS	Yes	No	No	No
ORDSKU HTS_ASSESS_TEMP	No	No	No	Yes
ORDSKU HTS_TEMP	No	No	No	Yes
ORDSKU_REV	No	Yes	No	No
ORDSKU_TEMP	No	No	No	Yes
OTB	Yes	Yes	Yes	No
OTB_CASCADE_STG	No	Yes	No	No
PARTNER_ORG_UNIT	Yes	No	No	No
POP_TERMS_DEF	No	No	No	Yes
POP_TERMS_FULFILLMENT	No	No	No	Yes
PROCUREMENT_UNIT_OPTIONS	Yes	No	No	No
REPL_RESULTS_TEMP	No	No	No	Yes
REQ_DOC	Yes	Yes	No	No
REQ_DOC_TEMP	No	No	No	Yes
REV_ORDERS	No	No	No	Yes
RTM_UNIT_OPTIONS	Yes	No	No	No
STORE	Yes	No	No	No
SUP_AVAIL	Yes	No	Yes	No
SUPS	Yes	No	No	No
SYSTEM_CONFIG_OPTIONS	Yes	No	No	No
TAX_CALC_EVENT	Yes	Yes	No	No
TAX_EVENT_RUN_TYPE	Yes	No	No	No
TICKET_REQUEST	No	Yes	No	No
TIMELINE_TEMP	No	No	No	Yes
TRANSIT_TIMES	Yes	No	No	No
V_PACKSKU_QTY	Yes	No	No	No
WH	Yes	No	No	No
WO_DETAIL_TEMP	No	No	No	Yes
WO_HEAD_TEMP	No	No	No	Yes
JOB_AUDIT_LOGS	No	Yes	No	No
JOB_AUDIT_PAYLOAD	No	Yes	No	No

Reject Purchase Orders

This section describes the Reject Purchase Orders service.

Business Overview

This service rejects a list of purchase order.

Service Type

Post

ReST URL

/PurchaseOrders/recent/rejectPurchaseOrders?orderNumbers={orderNumbers}

Input Parameters

Order Numbers - Required - comma separated list

Output

RpoStatusRDO

Parameter Name	Data Type
successOrdersCount	BigDecimal
successOrdersTable	List<BigDecimal>
failOrdersCount	BigDecimal
failOrdersTable	List<RpoFailRDO>

RpoFailRDO

Parameter Name	Data Type
orderNumber	BigDecimal
errorMessage	String

JSON Structure:

```
{
  "successOrdersCount": 0,
  "successOrdersTable": [],
  "failOrdersCount": 2,
  "failOrdersTable": [
    {
      "orderNumber": 123,
      "errorMessage": " Invalid Order Number.",
      "links": [],
      "hyperMediaContent": {
        "linkRDO": []
      }
    },
    {
      "orderNumber": 987,
```

```

        "errorMessage": "Invalid Order Number.",
        "links": [],
        "hyperMediaContent": {
            "linkRDO": []
        }
    },
    "links": [],
    "hyperMediaContent": {
        "linkRDO": []
    }
}

```

Table Impact

TABLE	SELECT	INSERT	UPDATE	DELETE
ALLOC_DETAIL	No	No	Yes	No
ALLOC_HEADER	Yes	No	Yes	No
CONTRACT_DETAIL	Yes	No	Yes	No
CONTRACT_HEADER	Yes	No	Yes	No
ITEM_MASTER	Yes	No	No	No
LC_ORDAPPLY	No	No	No	Yes
ORDHEAD	Yes	No	Yes	No
ORDLOC	Yes	No	No	No
OTB	No	No	Yes	No
SHIPMENT	Yes	No	No	No
SHIPSKU	Yes	No	No	No
SYSTEM_OPTIONS	Yes	No	No	No
JOB_AUDIT_LOGS	No	Yes	No	No
JOB_AUDIT_PAYLOAD	No	Yes	No	No

Reclass Detail Service

This section describes the Reclass Detail service.

Business Overview

This service is used to retrieve reclassification details for a given item.

Service Type

Get

ReST URL

Reclass/reclass?item={itemNumber}

Input Parameters

Parameter Name	Required	Description
Item	Yes	Item number

Output

RestReclassRecRDO

Parameter Name	Data Type
toClass	BigDecimal
reclassDate	Timestamp
reclassDesc	String
toSubclass	BigDecimal
reclassNo	BigDecimal
toDept	toDept

JSON Structure:

```
[
  {
    "toClass": null,
    "reclassDate": null,
    "reclassDesc": null,
    "toSubclass": null,
    "reclassNo": null,
    "toDept": null,
    "links": [],
    "hyperMediaContent": {
      "linkRDO": []
    }
  }
]
```

Table Impact

TABLE	SELECT	INSERT	UPDATE	DELETE
RECLASS_HEAD	Yes	No	No	No
RECLASS_ITEM	Yes	No	No	No
JOB_AUDIT_LOGS	No	Yes	No	No
JOB_AUDIT_PAYLOAD	No	Yes	No	No

Replenishment Schedule Services

This section describes the Replenishment Schedule service.

Functional Area

Inventory Movement

Business Overview

The primary role of these services is to create, modify, and delete scheduled replenishments and send them to Merchandising.

Create Replenishment Schedule

This section describes the Create Replenishment Schedule service.

Business Overview

This service creates scheduled replenishments by calling the SVCPROV_REPL package to load input data to the staging tables and then calling the core replenishment package to validate and insert data to the Merchandising tables.

Service Type

Post

ReST URL

inventory/replenishment/createReplSched

Input Parameters

ReplSchedCreModRDO

Parameter Name	Data Type
replAttrId	BigDecimal
schRplDesc	String
scheduledActiveDate	Long
replAction	String
item	String
diff1	String
diff2	String
diff3	String
diff4	String
dept	BigDecimal
class1	BigDecimal
subclass	BigDecimal
loc	BigDecimal
locType	String
autoRangeInd	String

Parameter Name	Data Type
activateDate	Long
deactivateDate	Long
presStock	BigDecimal
demoStock	BigDecimal
stockCat	String
replOrderCtrl	String
sourcingWh	BigDecimal
supplier	BigDecimal
originCountryId	String
pickupLeadTime	BigDecimal
whLeadTime	BigDecimal
replMethodInd	String
replMethod	String
minStock	BigDecimal
maxStock	BigDecimal
incrPct	BigDecimal
minSupplyDays	BigDecimal
maxSupplyDays	BigDecimal
timeSupplyHorizon	BigDecimal
addLeadTimeInd	String
invSellingDays	BigDecimal
serviceLevelType	String
serviceLevel	BigDecimal
serviceLevelFloatingStd	String
lostSalesFactor	BigDecimal
terminalStockQty	BigDecimal
seasonId	BigDecimal
phaseId	BigDecimal
rejectStoreOrdInd	String
multRunsPerDayInd	String
tsfZeroSohInd	String
nonScalingInd	String
maxScaleValue	BigDecimal
sizeProfileInd	String
reviewCycle	String
updateDaysInd	String
mondayInd	String
tuesdayInd	String
wednesdayInd	String

Parameter Name	Data Type
thursdayInd	String
fridayInd	String
saturdayInd	String
sundayInd	String
primaryPackNo	String
defaultPackInd	String
removePackInd	String
mraUpdate	String
mraRestore	String

JSON Structure:

```
[
  {
    "replAttrId": null,
    "schRplDesc": null,
    "scheduledActiveDate": null,
    "replAction": null,
    "item": null,
    "diff1": null,
    "diff2": null,
    "diff3": null,
    "diff4": null,
    "dept": null,
    "class1": null,
    "subclass": null,
    "loc": null,
    "locType": null,
    "autoRangeInd": null,
    "activateDate": null,
    "deactivateDate": null,
    "presStock": null,
    "demoStock": null,
    "stockCat": null,
    "replOrderCtrl": null,
    "sourcingWh": null,
    "supplier": null,
    "originCountryId": null,
    "pickupLeadTime": null,
    "whLeadTime": null,
    "replMethodInd": null,
    "replMethod": null,
    "minStock": null,
    "maxStock": null,
    "incrPct": null,
    "minSupplyDays": null,
    "maxSupplyDays": null,
    "timeSupplyHorizon": null,
    "addLeadTimeInd": null,
    "invSellingDays": null,
    "serviceLevelType": null,
    "serviceLevel": null,
    "serviceLevelFloatingStd": null,
    "lostSalesFactor": null,
    "terminalStockQty": null,
    "seasonId": null,
  }
]
```

```

    "phaseId": null,
    "rejectStoreOrdInd": null,
    "multRunsPerDayInd": null,
    "tsfZeroSohInd": null,
    "nonScalingInd": null,
    "maxScaleValue": null,
    "sizeProfileInd": null,
    "reviewCycle": null,
    "updateDaysInd": null,
    "mondayInd": null,
    "tuesdayInd": null,
    "wednesdayInd": null,
    "thursdayInd": null,
    "fridayInd": null,
    "saturdayInd": null,
    "sundayInd": null,
    "primaryPackNo": null,
    "defaultPackInd": null,
    "removePackInd": null,
    "mraUpdate": null,
    "mraRestore": null}
]

```

Output

ReplStatusRDO

Parameter Name	Data Type
statusMsg	String
failReplTable	List<ReplFailRDO>

ReplFailRDO

Parameter Name	Data Type
replAttrId	BigDecimal
item	String
dept	BigDecimal
class1	BigDecimal
subclass	BigDecimal
loc	BigDecimal
locType	String
effectiveDate	Long
errorMsg	String

The output will contain the status of the request including validation errors, if any.

JSON Structure:

```

{
  "statusMsg": null,
  "failReplTable": [
    {
      "replAttrId": null,
      "item": null,

```

```

    "dept": null,
    "class1": null,
    "subclass": null,
    "loc": null,
    "locType": null,
    "effectiveDate": null,
    "errorMsg": null,
    "links": [],
    "hyperMediaContent": {
      "linkRDO": []
    }
  },
  "links": [],
  "hyperMediaContent": {
    "linkRDO": []
  }
}

```

Table Impact

TABLE	SELECT	INSERT	UPDATE	DELETE
REPL_ATTR_UPDATE_HEAD	Yes	Yes	No	No
REPL_ATTR_UPDATE_ITEM	Yes	Yes	No	No
REPL_ATTR_UPDATE_LOC	Yes	Yes	No	No
SVC_PROCESS_TRACKER	Yes	Yes	Yes	No
CORESVC_REPL_ERR	No	Yes	No	No
SVC_REPL_ATTR_UPDATE	Yes	Yes	No	Yes
REPL_ITEM_LOC	Yes	No	No	No
ITEM_MASTER	Yes	No	No	No
ITEM_SUPP_COUNTRY	Yes	No	No	No
ITEM_LOC	Yes	No	No	No
SYSTEM_OPTIONS	Yes	No	No	No
V_STORE	Yes	No	No	No
V_WH	Yes	No	No	No
JOB_AUDIT_LOGS	No	Yes	No	No
JOB_AUDIT_PAYLOAD	No	Yes	No	No

Modify Replenishment Schedule

This section describes the Modify Replenishment service.

Business Overview

This service modifies scheduled replenishments by calling the SVCPROV_REPL package to load input to the staging tables and then calling the core replenishment package to validate and process data to the Merchandising tables.

Service Type

Post

ReST URL

inventory/replenishment/modifyReplSched

Input Parameters

ReplSchedCreModRDO

Parameter Name	Data Type
replAttrId	BigDecimal
schRplDesc	String
scheduledActiveDate	Long
replAction	String
item	String
diff1	String
diff2	String
diff3	String
diff4	String
dept	BigDecimal
class1	BigDecimal
subclass	BigDecimal
loc	BigDecimal
locType	String
autoRangeInd	String
activateDate	Long
deactivateDate	Long
presStock	BigDecimal
demoStock	BigDecimal
stockCat	String
replOrderCtrl	String
sourcingWh	BigDecimal
supplier	BigDecimal
originCountryId	String
pickupLeadTime	BigDecimal
whLeadTime	BigDecimal
replMethodInd	String
replMethod	String
minStock	BigDecimal
maxStock	BigDecimal

Parameter Name	Data Type
incrPct	BigDecimal
minSupplyDays	BigDecimal
maxSupplyDays	BigDecimal
timeSupplyHorizon	BigDecimal
addLeadTimeInd	String
invSellingDays	BigDecimal
serviceLevelType	String
serviceLevel	BigDecimal
serviceLevelFloatingStd	String
lostSalesFactor	BigDecimal
terminalStockQty	BigDecimal
seasonId	BigDecimal
phaseld	BigDecimal
rejectStoreOrdInd	String
multRunsPerDayInd	String
tsfZeroSohInd	String
nonScalingInd	String
maxScaleValue	BigDecimal
sizeProfileInd	String
reviewCycle	String
updateDaysInd	String
mondayInd	String
tuesdayInd	String
wednesdayInd	String
thursdayInd	String
fridayInd	String
saturdayInd	String
sundayInd	String
primaryPackNo	String
defaultPackInd	String
removePackInd	String
mraUpdate	String
mraRestore	String

JSON Structure:

```
[
  {
    "replAttrId": null,
    "schRplDesc": null,
    "scheduledActiveDate": null,
    "replAction": null,
    "item": null,
    "diff1": null,
```

```
"diff2": null,
"diff3": null,
"diff4": null,
"dept": null,
"class1": null,
"subclass": null,
"loc": null,
"locType": null,
"autoRangeInd": null,
"activateDate": null,
"deactivateDate": null,
"presStock": null,
"demoStock": null,
"stockCat": null,
"replOrderCtrl": null,
"sourcingWh": null,
"supplier": null,
"originCountryId": null,
"pickupLeadTime": null,
"whLeadTime": null,
"replMethodInd": null,
"replMethod": null,
"minStock": null,
"maxStock": null,
"incrPct": null,
"minSupplyDays": null,
"maxSupplyDays": null,
"timeSupplyHorizon": null,
"addLeadTimeInd": null,
"invSellingDays": null,
"serviceLevelType": null,
"serviceLevel": null,
"serviceLevelFloatingStd": null,
"lostSalesFactor": null,
"terminalStockQty": null,
"seasonId": null,
"phaseId": null,
"rejectStoreOrdInd": null,
"multRunsPerDayInd": null,
"tsfZeroSohInd": null,
"nonScalingInd": null,
"maxScaleValue": null,
"sizeProfileInd": null,
"reviewCycle": null,
"updateDaysInd": null,
"mondayInd": null,
"tuesdayInd": null,
"wednesdayInd": null,
"thursdayInd": null,
"fridayInd": null,
"saturdayInd": null,
"sundayInd": null,
"primaryPackNo": null,
"defaultPackInd": null,
"removePackInd": null,
"mraUpdate": null,
"mraRestore": null
}
]
```


Output

ReplStatusRDO

Parameter Name	Data Type
statusMsg	String
failReplTable	List<ReplFailRDO>

ReplFailRDO

Parameter Name	Data Type
replAttrId	BigDecimal
item	String
dept	BigDecimal
class1	BigDecimal
subclass	BigDecimal
loc	BigDecimal
locType	String
effectiveDate	Long
errorMsg	String

The output will contain the status of the request including validation errors, if any.

JSON Structure:

```
{
  "statusMsg": null,
  "failReplTable": [
    {
      "replAttrId": null,
      "item": null,
      "dept": null,
      "class1": null,
      "subclass": null,
      "loc": null,
      "locType": null,
      "effectiveDate": null,
      "errorMsg": null,
      "links": [],
      "hyperMediaContent": {
        "linkRDO": []
      }
    }
  ],
  "links": [],
  "hyperMediaContent": {
    "linkRDO": []
  }
}
```

Table Impact

TABLE	SELECT	INSERT	UPDATE	DELETE
REPL_ATTR_UPDATE_HEAD	Yes	Yes	No	No
REPL_ATTR_UPDATE_ITEM	Yes	Yes	No	No
REPL_ATTR_UPDATE_LOC	Yes	Yes	No	No
SVC_PROCESS_TRACKER	Yes	Yes	Yes	No
CORESVC_REPL_ERR	No	Yes	No	No
SVC_REPL_ATTR_UPDATE	Yes	Yes	No	Yes
REPL_ITEM_LOC	Yes	No	No	No
ITEM_MASTER	Yes	No	No	No
ITEM_SUPP_COUNTRY	Yes	No	No	No
ITEM_LOC	Yes	No	No	No
SYSTEM_OPTIONS	Yes	No	No	No
V_STORE	Yes	No	No	No
V_WH	Yes	No	No	No
JOB_AUDIT_LOGS	No	Yes	No	No
JOB_AUDIT_PAYLOAD	No	Yes	No	No

Delete Replenishment Schedule

This section describes the Delete Replenishment Schedule service.

Business Overview

This service deletes scheduled replenishments by calling the SVCPROV_REPL package to load input to the staging tables and then calling the core replenishment package to validate and delete data from the Merchandising tables.

Service Type

Post

ReST URL

inventory/replenishment/deleteReplSched

Input Parameters

ReplSchedDeIRDO

Parameter Name	Data Type
replAttrId	BigDecimal
item	String
dept	BigDecimal

Parameter Name	Data Type
class1	BigDecimal
subclass	BigDecimal
loc	BigDecimal
locType	String

JSON Structure:

```
[
  {
    "replAttrId": null,
    "item": null,
    "dept": null,
    "class1": null,
    "subclass": null,
    "loc": null,
    "locType": null
  }
]
```

Output

ReplStatusRDO

Parameter Name	Data Type
statusMsg	String
failReplTable	List<ReplFailRDO>

ReplFailRDO

The output will contain the status of the request including validation errors, if any.

Parameter Name	Data Type
replAttrId	BigDecimal
item	String
dept	BigDecimal
class1	BigDecimal
subclass	BigDecimal
loc	BigDecimal
locType	String
effectiveDate	Long
errorMsg	String

JSON Structure:

```
{
  "statusMsg": null,
  "failReplTable": [
    {
      "replAttrId": null,
      "item": null,

```

```

    "dept": null,
    "class1": null,
    "subclass": null,
    "loc": null,
    "locType": null,
    "effectiveDate": null,
    "errorMsg": null,
    "links": [],
    "hyperMediaContent": {
      "linkRDO": []
    }
  },
  "links": [],
  "hyperMediaContent": {
    "linkRDO": []
  }
}

```

Table Impact

TABLE	SELECT	INSERT	UPDATE	DELETE
REPL_ATTR_UPDATE_HEAD	Yes	Yes	No	Yes
REPL_ATTR_UPDATE_ITEM	Yes	Yes	No	Yes
REPL_ATTR_UPDATE_LOC	Yes	Yes	No	Yes
SVC_PROCESS_TRACKER	Yes	Yes	Yes	No
CORESVC_REPL_ERR	No	Yes	No	No
SVC_REPL_ATTR_UPDATE	Yes	Yes	No	Yes
JOB_AUDIT_LOGS	No	Yes	No	No
JOB_AUDIT_PAYLOAD	No	Yes	No	No

Shipment Detail Service

This section describes the Shipment Detail Service.

Business Overview

Shipment Detail service allows user to retrieve shipment and shipment item details for a given distro (transfer or allocation) or purchase order (PO).

Service Type

Get

ReST URL

Shipment/shipmentDetail?
 orderNumber={orderNumber}&distroNumber={distroNumber}&distroType={distroType}

Input Parameters

Parameter Name	Required	Description
orderNumber	No	Order Number. If none is specified, then Distro Number and Distro Type are required.
distroNumber	No	Distro Number. If none is specified, then Order Number is required.
distroType	No	Distro Type. If none is specified, then Order Number is required.

Output

RestShipmentRecRDO

Parameter Name	Data Type
shipment	BigDecimal
bolNo	String
asn	String
shipDate	Timestamp
receiveDate	Timestamp
estArrDate	Timestamp
shipOrigin	String
statusCode	String
toLoc	BigDecimal
toLocType	String
fromLoc	BigDecimal
fromLocType	String
parentShipment	BigDecimal
seqNo	BigDecimal
item	String
refItem	String
carton	String
invStatus	BigDecimal
shipskuStatusCode	String
qtyReceived	BigDecimal
unitCost	BigDecimal
unitRetail	BigDecimal
qtyExpected	BigDecimal
adjustType	String
actualReceivingStore	BigDecimal
reconcileUserId	String
reconcileDate	Timestamp

Parameter Name	Data Type
tamperedInd	String
dispositionedInd	String

JSON Structure:

```
{
  "shipment":null,
  "bolNo":null,
  "asn":null,
  "shipDate":null,
  "receiveDate":null,
  "estArrDate":null,
  "shipOrigin":null,
  "statusCode":null,
  "toLoc":null,
  "toLocType":null,
  "fromLoc":null,
  "fromLocType":null,
  "parentShipment":null,
  "seqNo":null,
  "item":null,
  "refItem":null,
  "carton":null,
  "invStatus":null,
  "shipskuStatusCode":null,
  "qtyReceived":null,
  "unitCost":null,
  "unitRetail":null,
  "qtyExpected":null,
  "adjustType":null,
  "actualReceivingStore":null,
  "reconcileUserId":null,
  "reconcileDate":null,
  "tamperedInd":null,
  "dispositionedInd":null,
  "links": [],
  "hyperMediaContent": {
    "linkRDO": []
  }
}
```

Table Impact

TABLE	SELECT	INSERT	UPDATE	DELETE
V_SHIPMENT	Yes	No	No	No
V_SHIPSKU	Yes	No	No	No
JOB_AUDIT_LOGS	No	Yes	No	No
JOB_AUDIT_PAYLOAD	No	Yes	No	No

Stock Count Detail Service

This section describes the Stock Count Detail service.

Business Overview

Stock Count Detail service allows user to retrieve open stock count details for a given item and/or store.

Service Type

Get

ReST URL

StockCount/stockCountDetail?
cycleCount={cycleCount}&locationType={locationType}&location={location}&item={item}&stocktakeDate={stocktakeDate}&pageSize={pageSize}&pageNumber={pageNumber}

Input Parameters

Parameter Name	Required	Description
Item	No	Item
Location	No	Location
Location Type	No	Location Type
Cycle Count	No	Cycle Count
Stocktake Date	No	Stocktake Date (always optional)

Output

RestStockCountRecRDO

Parameter Name	Data Type
cycleCount	BigDecimal
cycleCountDesc	String
stocktakeDate	Timestamp
stocktakeType	String
stakeSkuLoc	List<RestStakeSkuLocRecRDO>

RestStakeSkuLocRecRDO

Parameter Name	Data Type
item	String
location	BigDecimal
locType	String
snapshotOnHandQty	BigDecimal
snapshotInTransitQty	BigDecimal
snapshotUnitCost	BigDecimal
snapshotUnitRetail	BigDecimal

Parameter Name	Data Type
processed	String
physicalCountQty	BigDecimal
packCompQty	BigDecimal
inTransitAmt	BigDecimal
depositItemType	String
xformItemType	String
distributeQty	BigDecimal

JSON Structure:

```
{
  "cycleCount":null,
  "cycleCountDesc":null,
  "stocktakeDate":null,
  "stocktakeType":null,
  "stakeSkuLoc": [
    {
      "item":null,
      "location":null,
      "locType":null,
      "snapshotOnHandQty":null,
      "snapshotInTransitQty":null,
      "snapshotUnitCost":null,
      "snapshotUnitRetail":null,
      "processed":null,
      "physicalCountQty":null,
      "packCompQty":null,
      "inTransitAmt":null,
      "depositItemType":null,
      "xformItemType":null,
      "distributeQty":null,
      "links": [],
      "hyperMediaContent": {
        "linkRDO": []
      }
    }
  ],
  "links": [],
  "hyperMediaContent": {
    "linkRDO": []
  }
}
```

Table Impact

TABLE	SELECT	INSERT	UPDATE	DELETE
STAKE_HEAD	Yes	No	No	No
STAKE_SKU_LOC	Yes	No	No	No
JOB_AUDIT_LOGS	No	Yes	No	No
JOB_AUDIT_PAYLOAD	No	Yes	No	No

Store Day User Service

This section describes the Store Day User service.

Functional Area

Financials

Business Overview

The primary role of this service is to create or delete store day user and send them to Merchandising.

Create Store Day User

Business Overview

This service creates store day user by calling the SVCPROV_STOREDAYUSER package to load input data to the staging tables and then calling the core store day user package to validate and insert data to the Merchandising tables.

Service Type

Post

ReSTURL

financials/StoreDayUserREST/create StoreDayUser

Input Parameters

SvcprovSdudescRecRDO

Parameter Name	Data Type
store	BigDecimal
businessDate	String
userId	String

JSON Structure

```
[{"store": null,
  "businessDate": null,
  "userId": null}]
```

BusinessDate will take input in string with the format as 'DD-MON-YYYY' and later converted to Timestamp format and insert in table.

Output

SvcprovSduStatusRecRDO

Parameter Name	Data Type
statusMsg	String
sduErrTbl	List< SvcprovFailSduRecRDO >

SvcprovFailSduRecRDO

Parameter Name	Data Type
store	BigDecimal
businessDate	String
userId	String
errorMsg	String

The output will contain the status of the request including validation errors, if any.

JSON Structure:

```
{
  "statusMsg": null,
  " sduErrTbl ": [
    {
      " store": null,
      " businessDate": null,
      " userId": null,
      "errorMsg": null,
      "links": [],
      "hyperMediaContent": {
        "linkRDO": []
      }
    }
  ],
  "links": [],
  "hyperMediaContent": {
    "linkRDO": []
  }
}
```

Table Impact

TABLE	SELECT	INSERT	UPDATE	DELETE
STORE_DAY_USER	Yes	Yes	No	No
SVC_STORE_DAY_USER	Yes	Yes	Yes	Yes
JOB_AUDIT_LOGS	No	Yes	No	No
JOB_AUDIT_PAYLOAD	No	Yes	No	No

Delete Store Day User

Business Overview

This service deletes store day user by calling the SVCPROV_SDU package to load input data to the staging tables and then calling the core store day user package to validate and delete data to the Merchandising tables.

Service Type

Post

ReSTURL

financials/StoreDayUserREST/delete StoreDayUser

Input Parameters

SvcprovSdudescRecRDO

Parameter Name	Data Type
store	BigDecimal
businessDate	String
userId	String

JSON Structure

```
[{"store": null,
  " businessDate": null,
  " userId": null}]
```

BusinessDate will take input in string with the format as 'DD-MON-YYYY' and later converted to Timestamp format and delete from table.

Output

SvcprovSduStatusRecRDO

Parameter Name	Data Type
statusMsg	String
sduErrTbl	List< SvcprovFailSduRecRDO >

SvcprovFailSduRecRDO

Parameter Name	Data Type
store	BigDecimal
businessDate	String
userId	String
errorMsg	String

The output will contain the status of the request including validation errors, if any.

JSON Structure

```

{
  "statusMsg": null,
  " sduErrTbl ": [
    {
      " store": null,
      " businessDate": null,
      " userId": null,
      "errorMsg": null,
      "links": [],
      "hyperMediaContent": {
        "linkRDO": []
      }
    }
  ],
  "links": [],
  "hyperMediaContent": {
    "linkRDO": []
  }
}

```

Table Impact

TABLE	SELECT	INSERT	UPDATE	DELETE
STORE_DAY_USER	Yes	No	No	Yes
SVC_STORE_DAY_USER	Yes	Yes	Yes	Yes
JOB_AUDIT_LOGS	No	Yes	No	No
JOB_AUDIT_PAYLOAD	No	Yes	No	No

Store Detail Service

This section describes the Store Detail service.

Business Overview

Store Detail service allows user to retrieve Store information for a selected store or for all stores.

Service Type

Get

ReST URL

Store/storeDetail?store={storeNumber}

Input Parameters

Parameter Name	Required	Description
Store	No	Store ID. If none is specified, all stores will be retrieved.

Output

RestStoreRecRDO

Parameter Name	Data Type
store	BigDecimal
storeName	String
storeName10	String
storeName3	String
storeNameSecondary	String
storeClass	String
storeOpenDate	Timestamp
storeCloseDate	Timestamp
acquiredDate	Timestamp
remodelDate	Timestamp
vatRegion	BigDecimal
vatIncludeInd	String
stockholdingInd	String
channelId	BigDecimal
transferZone	BigDecimal
defaultWh	BigDecimal
stopOrderDays	BigDecimal
startOrderDays	BigDecimal
currencyCode	String
lang	BigDecimal
dunsNumber	String
dunsLoc	String
sisterStore	BigDecimal
tsfEntityId	BigDecimal
orgUnitId	BigDecimal
storeType	String
wfCustomerId	BigDecimal
timezoneName	String
customerOrderLocInd	String
company	BigDecimal

Parameter Name	Data Type
chain	BigDecimal
area	BigDecimal
region	BigDecimal
district	BigDecimal
add1	String
add2	String
add3	String
city	String
state	String
countryId	String
post	String
contactName	String
contactPhone	String
contactEmail	String

JSON Structure:

```
{
  "store": null,
  "storeName": null,
  "storeName10": null,
  "storeName3": null,
  "storeNameSecondary": null,
  "storeClass": null,
  "storeOpenDate": null,
  "storeCloseDate": null,
  "acquiredDate": null,
  "remodelDate": null,
  "vatRegion": null,
  "vatIncludeInd": null,
  "stockholdingInd": null,
  "channelId": null,
  "transferZone": null,
  "defaultWh": null,
  "stopOrderDays": null,
  "startOrderDays": null,
  "currencyCode": null,
  "lang": null,
  "dunsNumber": null,
  "dunsLoc": null,
  "sisterStore": null,
  "tsfEntityId": null,
  "orgUnitId": null,
  "storeType": null,
  "wfCustomerId": null,
  "timezoneName": null,
  "customerOrderLocInd": null,
  "company": null,
  "chain": null,
  "area": null,
  "region": null,
  "district": null,
  "add1": null,
```

```

    "add2": null,
    "add3": null,
    "city": null,
    "state": null,
    "countryId": null,
    "post": null,
    "contactName": null,
    "contactPhone": null,
    "contactEmail": null,
    "links": [],
    "hyperMediaContent": {
      "linkRDO": []
    }
  }
}

```

Table Impact

TABLE	SELECT	INSERT	UPDATE	DELETE
V_STORE	Yes	No	No	No
STORE_HIERARCHY	Yes	No	No	No
ADDR	Yes	No	No	No
JOB_AUDIT_LOGS	No	Yes	No	No
JOB_AUDIT_PAYLOAD	No	Yes	No	No

Supplier Detail Service

This section describes the Supplier Detail service.

Business Overview

Supplier Detail service allows user to retrieve Supplier information for a selected supplier.

Service Type

Get

ReST URL

Supplier/supplierDetail?supplierNumber={suppliernumber}

Input Parameters

Parameter Name	Required	Description
Supplier	Yes	Supplier number

Output

RestSupplierRecRDO

Parameter Name	Data Type
supplier	BigDecimal
sup_name	String
sup_name_secondary	String
supplier_parent	BigDecimal
sup_status	String
currency_code	String
terms	String
freight_terms	String
vat_region	BigDecimal
external_ref_id	String
Supplier_address	List<RestSupplierAddressRecRDO>

RestSupplierAddressRecRDO

Parameter Name	Data Type
add_1	String
add_2	String
add_3	String
city	String
state	String
country_id	String
post	String
contact_name	String
contact_phone	String
contact_email	String

JSON Structure:

```
[
  {
    "supplierAddress": [
      {
        "countryId": null,
        "add2": null,
        "add3": null,
        "city": null,
        "add1": null,
        "state": null,
        "contactEmail": null,
        "contactName": null,
        "contactPhone": null,
        "post": null,
        "links": [],
        "hyperMediaContent": {
          "linkRDO": []
        }
      }
    ],
  },
  {
```



```
    "countryId": null,
    "add2": null,
    "add3": null,
    "city": null,
    "add1": null,
    "state": null,
    "contactEmail": null,
    "contactName": null,
    "contactPhone": null,
    "post": null,
    "links": [],
    "hyperMediaContent": {
      "linkRDO": []
    }
  },
  {
    "countryId": null,
    "add2": null,
    "add3": null,
    "city": null,
    "add1": null,
    "state": null,
    "contactEmail": null,
    "contactName": null,
    "contactPhone": null,
    "post": null,
    "links": [],
    "hyperMediaContent": {
      "linkRDO": []
    }
  },
  {
    "countryId": null,
    "add2": null,
    "add3": null,
    "city": null,
    "add1": null,
    "state": null,
    "contactEmail": null,
    "contactName": null,
    "contactPhone": null,
    "post": null,
    "links": [],
    "hyperMediaContent": {
      "linkRDO": []
    }
  }
],
{
  "countryId": null,
  "add2": null,
  "add3": null,
  "city": null,
  "add1": null,
  "state": null,
  "contactEmail": null,
  "contactName": null,
  "contactPhone": null,
  "post": null,
  "links": [],
  "hyperMediaContent": {
    "linkRDO": []
  }
}
```

```

    }
  },
  "supNameSecondary": null,
  "supplierParent": null,
  "terms": null,
  "supStatus": null,
  "currencyCode": null,
  "supplier": null,
  "supName": null,
  "freightTerms": null,
  "vatRegion": null,
  "externalRefId": null,
  "links": [],
  "hyperMediaContent": {
    "linkRDO": []
  }
}
]

```

Table Impact

TABLE	SELECT	INSERT	UPDATE	DELETE
SUPS	Yes	No	No	No
ADDR	Yes	No	No	No
JOB_AUDIT_LOGS	No	Yes	No	No
JOB_AUDIT_PAYLOAD	No	Yes	No	No

Tax Rules Services

Functional Area

Foundation

Business Overview

The primary role of this service is to create, update or delete Tax Rules in Merchandising.

Create Tax Rule

Business Overview

Create a new Tax Rule

Service Type

Post

ReST URL

`/foundation/taxRules/create`

Input Parameters

Parameter Name	Required	Data Type	Description
ruleId	No	Number (15)	Tax rule unique ID.
ruleName	No	String (300)	Tax Rule Name.
taxRegionSource	No	Number (4)	Source Tax Region.
taxRegionDestination	No	Number (4)	Destination Tax Region.
activeDate	No	date	Effective date when this tax rule become active.
endDate	No	date	End date when this tax rule expired.
status	No	String (6)	Status of the Rule. Valid Values are AC for Active, AP for Approved, O for Obsolete, S for Submitted, U for Updated, W for Worksheet.
taxCode	No	String (6)	Tax code.
taxCalcType	No	String (6)	Define the tax calculation type. Possible values are P for Percentage or U for Unit.
taxRate	No	Number (20,10)	Tax rate in percentage when tax calculation type is in percentage.
taxBasisType	No	String (6)	Indicate which value is used as base calculation to apply tax. Valid values are: C for UNIT_COST, R for UNIT_RETAIL, B for BOTH, or N for NON_MERCH_COST.
taxBasisFactor	No	Number (20,10)	Contains a unique user specified code representing the non-merchandise code.
nonMerchCode	No	String (6)	Contains a unique user specified code representing the non-merchandise code.
uomType	No	String (4)	UOM type selected to be considered when tax calculation is in unit.
uomValue	No	Number (20,10)	UOM value selected to be considered when tax calculation is in unit.
currencyCode	No	String (3)	Contains the currency code for the tax rule. For unit tax application.
reverseTaxInd	No	String (1)	Reverse Tax Indicator: Valid values are Y for Enable or N for Disable.

Parameter Name	Required	Data Type	Description
reverseTaxThreshold	No	Number (20,10)	Threshold amount to be taken in consideration while returning the tax code. It will have value when REVERSE_TAX_IND is Y.
predecessorTaxRulesId	No	Number (15)	Holds the predecessor rule ID that the result of the tax calculated in the predecessor rule will be added to the tax basis of the current rule in order to address Tax over Tax calculation requirement.
regionExplodeType	No	String (6)	Tax over tax rate calculated based in predecessor tax rule when applied.
gtsRulesConfigTITbl	No	Collection of Object	Translation records associated with tax record.
gtsRulesConditionsTbl	No	Collection of Object	Conditions associated with tax record.
gtsRulesCfaTbl	No	Collection of Object	Rules restrictions associated with tax record.

GtsRulesConfigTITbl - This tables receive information regarding translations that may be necessary.

Parameter Name	Required	Data Type	Description
ruleName	No	String (300)	Translated Rule.
lang	No	Number (6)	Lang.

GtsRulesConditionsTbl - This table receives the conditions that will be used in the processing.

Parameter Name	Required	Data Type	Description
conditionType	No	String (6)	Type of condition used. Valid values are 1 for All Departments, 2 for Department, 3 for Class, 4 for Subclass, 5 for Item, 6 for Parent/Diff, 7 for Item List and 8 for Upload List.
dept	No	Number (4)	Item Department.
class	No	Number (4)	Item Class.
subclass	No	Number (4)	Item subclass.
itemId	No	String (25)	Item ID.
itemList	No	Number (8)	This field contains Item List number that was used to create this item record.
excludeInd	No	String (1)	Indicates if the condition in exclude or include. Valid values are Y or N.

Parameter Name	Required	Data Type	Description
diffId	No	String (10)	Diff ID used when condition type is Parent/DIFF.

GtsRulesCfaTbl - This table (CFA) receives and process the restrictions associated with the rules.

Parameter Name	Required	Data Type	Description
attributeId	No	Number (10)	This column holds the attribute id that this extended data is associated with.
attributeValue	No	String (250)	This column holds the CFA value selected for the rule.
restrictionLevel	No	String (6)	Restriction Level. Valid values are M for Merchandise Hierarchy, D for Destination Region and S for Source Region.

JSON Structure:

```
{
  "ruleId": 919875,
  "ruleName": "Tax Rules US-CA",
  "taxRegionSource": 1000,
  "taxRegionDestination": 200,
  "activeDate": "2022-09-01",
  "endDate": "2022-12-31",
  "status": "S",
  "taxCode": "S",
  "taxCalcType": "P",
  "taxRate": 35.00,
  "taxBasisType": "B",
  "taxBasisFactor": 11.00,
  "nonMerchCode": "",
  "uomType": "",
  "uomValue": 68.00,
  "currencyCode": "",
  "reverseTaxInd": "",
  "reverseTaxThreshold": 19.00,
  "predecessorTaxRulesId": 514301,
  "regionExplodeType": "A",
  "gtsRulesConfigTlTbl": [
    {
      "ruleName": "Tax Rules TL US-CA",
      "lang": 2
    }
  ],
  "gtsRulesConditionsTbl": [
    {
      "conditionType": "1"
    }
  ]
}
```

```

    ],
    "gtsRulesCfaTbl": [
      {
        "attributeId": 4,
        "attributeValue": "A",
        "restrictionLevel": "S"
      }
    ]
  }
}

```

Output

Element Name	Required	Data Type	Description
status	No	String (255)	Process Execution status. E – Execution finished with errors. S – Execution completed successfully.
errMsg	No	String (4000)	Error message, in case of error.
errTbl	No	Collection of Object	Error table in case of existing errors.

ErrTbl - Table used to detail errors associated with execution.

Element Name	Required	Data Type	Description
columnName	No	String (255)	Column name associated with error message.
errorMsg	No	String (4000)	Error message.

JSON Structure:

```

{
  "status": "E",
  "errMsg": "",
  "errTbl": [
    {
      "columnName": "STATUS",
      "errorMsg": "Invalid Status.", "
    }
  ]
}

```

Response Code: 400 (Error)

In case of error, the following standard error response will be returned. The element `validationErrors` is present when the input payload or input parameters do not match the schema definition for this service.

JSON Structure:

```
{
  "status": "ERROR",
  "message": "Error found in validation of input payload",
  "validationErrors": [
    {
      "error": "must be one of Y, N",
      "field": "createRecord.arg0.approveInd",
      "inputValue": "X"
    }
  ]
}
```

Table Impact

TABLE	SELECT	INSERT	UPDATE	DELETE
SVC_GTS_RULES_CONFIG	Yes	Yes	No	No
SVC_GTS_RULES_CONFIG_TL	Yes	Yes	No	No
SVC_GTS_RULES_CFA	Yes	Yes	No	No
SVC_GTS_RULES_CONDITIONS	Yes	Yes	No	No
GTS_RULES_CONFIG	Yes	No	No	No
GTS_RULES_CONFIG_TL	Yes	No	No	No
GTS_RULES_CFA	Yes	No	No	No
GTS_RULES_CONDITIONS	Yes	No	No	No

Update Tax Rule

Business Overview

Update an existing Tax Rule

Service Type

Put

ReST URL

/foundation/taxRules/update

Input Parameters

Element Name	Required	Data Type	Description
ruleId	No	Number (15)	Tax rule unique ID.
ruleName	No	String (300)	Tax Rule Name.

Element Name	Required	Data Type	Description
taxRegionSource	No	Number (4)	Source Tax Region.
taxRegionDestination	No	Number (4)	Destination Tax Region.
activeDate	No	date	Effective date when this tax rule become active.
endDate	No	date	End date when this tax rule expired.
status	No	String (6)	Status of the Rule. Valid Values are AC for Active, AP for Approved, O for Obsoleted, S for Submitted, U for Updated, W for Worksheet.
taxCode	No	String (6)	Tax code.
taxCalcType	No	String (6)	Define the tax calculation type. Possible values are P for Percentage or U for Unit.
taxRate	No	Number (20,10)	Tax rate in percentage when tax calculation type is in percentage.
taxBasisType	No	String (6)	Indicate which value is used as base calculation to apply tax. Valid values are: C for UNIT_COST, R for UNIT_RETAIL, B for BOTH, or N for NON_MERCH_COST.
taxBasisFactor	No	Number (20,10)	Contains a unique user specified code representing the non-merchandise code.
nonMerchCode	No	String (6)	Contains a unique user specified code representing the non-merchandise code.
uomType	No	String (4)	UOM type selected to be considered when tax calculation is in unit.
uomValue	No	Number (20,10)	UOM value selected to be considered when tax calculation is in unit.
currencyCode	No	String (3)	Contains the currency code for the tax rule. For unit tax application.
reverseTaxInd	No	String (1)	Reverse Tax Indicator: Valid values are Y for Enable or N for Disable.
reverseTaxThreshold	No	Number (20,10)	Threshold amount to be taken in consideration while returning the tax code. It will have value when REVERSE_TAX_IND is Y.
predecessorTaxRulesId	No	Number (15)	Holds the predecessor rule ID that the result of the tax calculated in the predecessor rule will be added to the tax basis of the current rule in order to address Tax over Tax calculation requirement.
regionExplodeType	No	String (6)	Tax over tax rate calculated based in predecessor tax rule when applied.

Element Name	Required	Data Type	Description
gtsRulesConfigTITbl	No	Collection of Object	Translation records associated with tax record.
gtsRulesConditionsTbl	No	Collection of Object	Conditions associated with tax record.
gtsRulesCfaTbl	No	Collection of Object	Rules restrictions associated with tax record.

GtsRulesConfigTITbl - This table receive information regarding translations that may be necessary.

Element Name	Required	Data Type	Description
ruleName	No	String (300)	Translated Rule.
lang	No	Number (6)	Lang.

GtsRulesConditionsTbl - This table receives the conditions that will be used in the processing.

Element Name	Required	Data Type	Description
conditionType	No	String (6)	Type of condition used. Valid values are 1 for All Departments, 2 for Department, 3 for Class, 4 for Subclass, 5 for Item, 6 for Parent/Diff, 7 for Item List and 8 for Upload List.
dept	No	Number (4)	Item Department.
class	No	Number (4)	Item Class.
subclass	No	Number (4)	Item subclass.
itemId	No	String (25)	Item ID.
itemList	No	Number (8)	This field contains Item List number that was used to create this item record.
excludeInd	No	String (1)	Indicates if the condition in exclude or include. Valid values are Y or N.
diffId	No	String (10)	Diff ID used when condition type is Parent/DIFF.

GtsRulesCfaTbl - This table (CFA) receives and process the restrictions associated with the rules.

Element Name	Required	Data Type	Description
attributeId	No	Number (10)	This column holds the attribute id that this extended data is associated with.
attributeValue	No	String (250)	This column holds the CFA value selected for the rule.

Element Name	Required	Data Type	Description
restrictionLevel	No	String (6)	Restriction Level. Valid values are M for Merchandise Hierarchy, D for Destination Region and S for Source Region.

JSON Structure:

```
{
  "ruleId": 919875,
  "ruleName": "Tax Rules US-CA",
  "taxRegionSource": 1000,
  "taxRegionDestination": 200,
  "activeDate": "2022-09-01",
  "endDate": "2022-12-31",
  "status": "W",
  "taxCode": "S",
  "taxCalcType": "P",
  "taxRate": 35.00,
  "taxBasisType": "B",
  "taxBasisFactor": 11.00,
  "nonMerchCode": "",
  "uomType": "",
  "uomValue": 68.00,
  "currencyCode": "",
  "reverseTaxInd": "",
  "reverseTaxThreshold": 19.00,
  "predecessorTaxRulesId": 514301,
  "regionExplodeType": "A",
  "gtsRulesConfigTlTbl": [
    {
      "ruleName": "Tax Rules TL US-CA",
      "lang": 2
    }
  ],
  "gtsRulesConditionsTbl": [
    {
      "conditionType": "1"
    }
  ],
  "gtsRulesCfaTbl": [
    {
      "attributeId": 4,
      "attributeValue": "A",
      "restrictionLevel": "S"
    }
  ]
}
```

Output

Element Name	Required	Data Type	Description
status	No	String (255)	Process Execution status. E – Execution finished with errors. S – Execution completed successfully.
errMsg	No	String (4000)	Error message, in case of error.
errTbl	No	Collection of Object	Error table in case of existing errors.

ErrTbl - Table used to detail errors associated with execution.

Element Name	Required	Data Type	Description
columnName	No	String (255)	Column name associated with error message.
errorMsg	No	String (4000)	Error message.

JSON Structure:

```
{
  "status": "E",
  "errMsg": " CANNOT_UPD_RULE ",
  "errTbl": [
    {
      "columnName": "",
      "errorMsg": " The rule was not submitted successfully."
    }
  ]
}
```

Response Code: 400 (Error)

In case of error, the following standard error response is returned. The element `validationErrors` is present when the input payload or input parameters do not match the schema definition for this service.

JSON Structure:

```
{
  "status": "ERROR",
  "message": "Error found in validation of input payload",
  "validationErrors": [
    {
      "error": "must be one of Y, N",
      "field": "createRecord.arg0.approveInd",
      "inputValue": "X"
    }
  ]
}
```

Table Impact

TABLE	SELECT	INSERT	UPDATE	DELETE
SVC_GTS_RULES_CONFIG	Yes	Yes	No	No
SVC_GTS_RULES_CONFIG_TL	Yes	Yes	No	No
SVC_GTS_RULES_CFA	Yes	Yes	No	No
SVC_GTS_RULES_CONDITIONS	Yes	Yes	No	No
GTS_RULES_CONFIG	Yes	No	Yes	No
GTS_RULES_CONFIG_TL	Yes	No	Yes	No
GTS_RULES_CFA	Yes	No	Yes	No
GTS_RULES_CONDITIONS	Yes	No	Yes	No

Delete Tax Rule

Business Overview

Delete an existing Tax Rule

Service Type

Delete

ReST URL

/foundation/taxRules/delete

Input Parameters

Element Name	Required	Data Type	Description
ruleId	No	Number (15)	Tax rule unique ID.
ruleName	No	String (300)	Tax Rule Name.
taxRegionSource	No	Number (4)	Source Tax Region.
taxRegionDestination	No	Number (4)	Destination Tax Region.
activeDate	No	date	Effective date when this tax rule become active.
endDate	No	date	End date when this tax rule expired.
status	No	String (6)	Status of the Rule. Valid Values are AC for Active, AP for Approved, O for Obsolete, S for Submitted, U for Updated, W for Worksheet.
taxCode	No	String (6)	Tax code.

Element Name	Required	Data Type	Description
taxCalcType	No	String (6)	Define the tax calculation type. Possible values are P for Percentage or U for Unit.
taxRate	No	Number (20,10)	Tax rate in percentage when tax calculation type is in percentage.
taxBasisType	No	String (6)	Indicate which value is used as base calculation to apply tax. Valid values are: C for UNIT_COST, R for UNIT_RETAIL, B for BOTH, or N for NON_MERCH_COST.
taxBasisFactor	No	Number (20,10)	Contains a unique user specified code representing the non-merchandise code.
nonMerchCode	No	String (6)	Contains a unique user specified code representing the non-merchandise code.
uomType	No	String (4)	UOM type selected to be considered when tax calculation is in unit.
uomValue	No	Number (20,10)	UOM value selected to be considered when tax calculation is in unit.
currencyCode	No	String (3)	Contains the currency code for the tax rule. For unit tax application.
reverseTaxInd	No	String (1)	Reverse Tax Indicator: Valid values are Y for Enable or N for Disable.
reverseTaxThreshold	No	Number (20,10)	Threshold amount to be taken in consideration while returning the tax code. It will have value when REVERSE_TAX_IND is Y.
predecessorTaxRulesId	No	Number (15)	Holds the predecessor rule ID that the result of the tax calculated in the predecessor rule will be added to the tax basis of the current rule to address Tax over Tax calculation requirement.
regionExplodeType	No	String (6)	Tax over tax rate calculated based in predecessor tax rule when applied.
gtsRulesConfigTITbl	No	Collection of Object	Translation records associated with tax record.
gtsRulesConditionsTbl	No	Collection of Object	Conditions associated with tax record.
gtsRulesCfaTbl	No	Collection of Object	Rules restrictions associated with tax record.

GtsRulesConfigTITbl - This table receive information regarding translations that may be necessary.

Element Name	Required	Data Type	Description
ruleName	No	String (300)	Translated Rule.
lang	No	Number (6)	Lang.

GtsRulesConditionsTbl - This table receives the conditions that will be used in the processing.

Element Name	Required	Data Type	Description
conditionType	No	String (6)	Type of condition used. Valid values are 1 for All Departments, 2 for Department, 3 for Class, 4 for Subclass, 5 for Item, 6 for Parent/Diff, 7 for Item List and 8 for Upload List.
dept	No	Number (4)	Item Department.
class	No	Number (4)	Item Class.
subclass	No	Number (4)	Item subclass.
itemId	No	String (25)	Item ID.
itemList	No	Number (8)	This field contains Item List number that was used to create this item record.
excludeInd	No	String (1)	Indicates if the condition in exclude or include. Valid values are Y or N.
diffId	No	String (10)	Diff ID used when condition type is Parent/DIFF.

GtsRulesCfaTbl - This table (CFA) receives and process the restrictions associated with the rules.

Element Name	Required	Data Type	Description
attributeId	No	Number (10)	This column holds the attribute id that this extended data is associated with.
attributeValue	No	String (250)	This column holds the CFA value selected for the rule.
restrictionLevel	No	String (6)	Restriction Level. Valid values are M for Merchandise Hierarchy, D for Destination Region and S for Source Region.

JSON Structure:

```
{
  "ruleId": 919875,
  "ruleName": "Tax Rules US-CA",
  "taxRegionSource": 1000,
  "taxRegionDestination": 200,
  "activeDate": "2022-09-01",
```

```

"endDate": "2022-12-31",
"status": "W",
"taxCode": "S",
"taxCalcType": "P",
"taxRate": 35.00,
"taxBasisType": "B",
"taxBasisFactor": 11.00,
"nonMerchCode": "",
"uomType": "",
"uomValue": 68.00,
"currencyCode": "",
"reverseTaxInd": "",
"reverseTaxThreshold": 19.00,
"predecessorTaxRulesId": 514301,
"regionExplodeType": "A",
"gtsRulesConfigTlTbl": [
  {
    "ruleName": "Tax Rules TL US-CA",
    "lang": 2
  }
],
"gtsRulesConditionsTbl": [
  {
    "conditionType": "1"
  }
],
"gtsRulesCfaTbl": [
  {
    "attributeId": 4,
    "attributeValue": "A",
    "restrictionLevel": "S"
  }
]
}

```

Output

Element Name	Required	Data Type	Description
status	No	String (255)	Process Execution status. E – Execution finished with errors. S – Execution completed successfully.
errMsg	No	String (4000)	Error message, in case of error.
errTbl	No	Collection of Object	Error table in case of existing errors.

ErrTbl - Table used to detail errors associated with execution.

Element Name	Required	Data Type	Description
columnName	No	String (255)	Column name associated with error message.

Element Name	Required	Data Type	Description
errorMsg	No	String (4000)	Error msg.

JSON Structure:

```
{
  "status": "E",
  "errMsg": " CANNOT_UPD_RULE ",
  "errTbl": [
    {
      "columnName": "",
      "errorMsg": " The rule was not submitted successfully."
    }
  ]
}
```

Response Code: 400 (Error)

In case of error, the following standard error response is returned. The element `validationErrors` is present when the input payload or input parameters do not follow the schema definition for this service.

JSON Structure:

```
{
  "status": "ERROR",
  "message": "Error found in validation of input payload",
  "validationErrors": [
    {
      "error": "must be one of Y, N",
      "field": "createRecord.arg0.approveInd",
      "inputValue": "X"
    }
  ]
}
```

Table Impact

TABLE	SELECT	INSERT	UPDATE	DELETE
SVC_GTS_RULES_CONFIG	Yes	Yes	No	No
SVC_GTS_RULES_CONFIG_TL	Yes	Yes	No	No
SVC_GTS_RULES_CFA	Yes	Yes	No	No
SVC_GTS_RULES_CONDITIONS	Yes	Yes	No	No
GTS_RULES_CONFIG	Yes	No	No	Yes
GTS_RULES_CONFIG_TL	Yes	No	No	Yes
GTS_RULES_CFA	Yes	No	No	Yes
GTS_RULES_CONDITIONS	Yes	No	No	Yes

Tran Data Upload Service

Functional Area

Financials

Business Overview

The primary role of this service is to create tran data records in the stock ledger, such as to adjust previously posted transactions. If the transaction is marked for WAC recalculation, it will recalculate the WAC for the item/location at the same time. The transactions written by this service will not impact unit inventory in Merchandising.

Service Type

Post

ReST URL

/Financials/uploadTranData

Input Parameters

Parameter Name	Required?	Data Type	Description
Item	No	String	Item Identification
Dept	Yes	Number	Department Id
Class	Yes	Number	Class Id
Subclass	Yes	Number	Subclass Id
LocType	Yes	String	Location Type, Valid values are 'S'tore,'W'arehouse and 'E'xternal Finisher.
Location	Yes	Number	Location Id
TranDate	No	Date	Date of Transaction. (Format: 'DD-MON-YYYY').If null vdate will be populated.
TranCode	Yes	Number	Transaction Type Identification.
AdjCode	No	String	Type of Adjustment. Valid values are 'A'-Actual Landed Cost Adjustment, 'C'-Cost Adjustment, 'U'-Unit Adjustment.
Units	Yes	BigDecima I	Number of units involved in transaction.
TotalCost	No	BigDecima I	Total Cost associated with transaction.
TotalRetail	No	BigDecima I	Total Retail value associated with transaction.

Parameter Name	Required?	Data Type	Description
RefNo1	No	String	Reference number associated with transaction.
RefNo2	No	String	Reference number associated with transaction.
GLRefNo	No	String	Reference number associated with transactions, and is used for defining the General Ledger account relationship, along with dept, class, subclass, location and tran_code.
OldUnitRetail	No	BigDecimal	Old Unit Retail
NewUnitRetail	No	BigDecimal	New Unit Retail
SalesType	No	String	Type of Sale. Valid Values are 'Clearance', 'Promotion' and 'Regular'.
VatRate	No	BigDecimal	VAT Rate
Avg Cost	No	BigDecimal	Average cost for the SKU
RefPackNo	No	String	Pack Number for the transaction item.
TotalCostExclusiveELC	No	BigDecimal	Cost Exclusive for ELC
WACRecalcIndicator	No	String	WAC should be recalculated. Valid values are 'Yes' or 'No'.

Example JSON input:

```
[
  {
    "item": "string",
    "dept": 0,
    "classId": 0,
    "subClass": 0,
    "locType": "string",
    "location": 0,
    "tranDate": "string",
    "tranCode": 0,
    "adjCode": "string",
    "units": 0,
    "totalCost": 0,
    "totalRetail": 0,
    "refNo1": "string",
    "refNo2": "string",
    "glRefNo": "string",
    "oldUnitRetail": 0,
    "newUnitRetail": 0,
    "salesType": "string",
    "vatRate": 0,
    "avCost": 0,
    "refPackNo": "string",
```

```

    "totalCostExclElc": 0,
    "wacRecalcInd": "string"
    "hyperMediaContent": {}
  }
]

```

Output

Table 5-22 RestTranDataStatusRec

Parameter Name	Data Type
Status Message	String
TranDataErrorTBL	List< RestTranDataFailTbl>

Table 5-23 TranDataErrorTBL

Parameter Name	Data Type
Item	String
Dept	Number
Class	Number
Subclass	Number
Location Type	String
Location	Number
Tran Code	Number
Units	BigDecimal
Total Cost	BigDecimal
Total Retail	BigDecimal
RefNo1	String
RefNo2	String
GLRefNo	String
WACRecalcIndicator	String
ErrorMessage	String

Table Impact

TABLE	SELECT	INSERT	UPDATE	DELETE
TRAN_DATA	No	Yes	No	No
ITEM_MASTER	Yes	No	No	No
DEPS	Yes	No	No	No
SUBCLASS	Yes	No	No	No
STORE	Yes	No	No	No
WH	Yes	No	No	No

TABLE	SELECT	INSERT	UPDATE	DELETE
PARTNER	Yes	No	No	No
TRAN_DATA_CODES	Yes	No	No	No
ITEM_LOC	Yes	No	No	No
ITEM_LOC_SOH	Yes	No	Yes	No
VAT_RATE	Yes	No	No	No
VAT_REGION	Yes	No	No	No
PACKITEM	Yes	No	No	No
SYSTEM_OPTIONS	Yes	No	No	No
PERIOD	Yes	No	No	No

Transfer Detail Service

This section describes the Transfer Detail service.

Business Overview

Transfer Detail service allows user to retrieve details for a given transfer.

Service Type

Get

ReST URL

Transfer/transferDetail?transferNumber={transferNumber}

Input Parameters

Parameter Name	Required	Description
Transfer Number	Yes	Transfer number

Output

RestTsfheadRecRDO

Parameter Name	Data Type
tsfNo	BigDecimal
tsfParentNo	BigDecimal
fromLocType	String
fromLoc	BigDecimal
toLocType	String
toLoc	BigDecimal
expDcDate	Timestamp

Parameter Name	Data Type
dept	BigDecimal
inventoryType	String
tsfType	String
status	String
deliveryDate	Timestamp
closeDate	Timestamp
notAfterDate	Timestamp
contextType	String
contextValue	String
wfOrderNo	BigDecimal
tsfdetail	List<RestTsfdetailRecRDO>

RestTsfdetailRecRDO

Parameter Name	Data Type
tsfSeqNo	BigDecimal
item	String
invStatus	BigDecimal
tsfPrice	BigDecimal
tsfQty	BigDecimal
fillQty	BigDecimal
shipQty	BigDecimal
receivedQty	BigDecimal
reconciledQty	BigDecimal
distroQty	BigDecimal
selectedQty	BigDecimal
cancelledQty	BigDecimal
suppPackSize	BigDecimal
tsfCost	BigDecimal
publishInd	String

JSON Structure:

```
{
  "tsfNo": null,
  "tsfParentNo": null,
  "fromLocType": null,
  "fromLoc": null,
  "toLocType": null,
  "toLoc": null,
  "expDcDate": null,
  "dept": null,
  "inventoryType": null,
  "tsfType": null,
  "status": null,
  "deliveryDate": null,
```

```

"closeDate": null,
"notAfterDate": null,
"contextType": null,
"contextValue": null,
"wfOrderNo": null,
"tsfdetail": [
  {
    "tsfSeqNo": null,
    "item": null,
    "invStatus": null,
    "tsfPrice": null,
    "tsfQty": null,
    "fillQty": null,
    "shipQty": null,
    "receivedQty": null,
    "reconciledQty": null,
    "distroQty": null,
    "selectedQty": null,
    "cancelledQty": null,
    "suppPackSize": null,
    "tsfCost": null,
    "publishInd": null,
    "links": [],
    "hyperMediaContent": {
      "linkRDO": []
    }
  }
],
"links": [],
"hyperMediaContent": {
  "linkRDO": []
}
}

```

Table Impact

TABLE	SELECT	INSERT	UPDATE	DELETE
TSFHEAD	Yes	No	No	No
TSFDETAIL	Yes	No	No	No
JOB_AUDIT_LOGS	No	Yes	No	No
JOB_AUDIT_PAYLOAD	No	Yes	No	No

VAT Detail Service

This section describes the VAT Detail service.

Business Overview

VAT Detail service allows user to retrieve VAT information for a selected department.

Service Type

Get

ReST URL

Vat/vatDetail?department={departmentNumber}

Input Parameters

Parameter Name	Required	Description
Department	Yes	Department ID

Output

RestVatRecRDO

Parameter Name	Data Type
vatRegion	BigDecimal
vatRegionName	String
vatRegionType	String
vatType	String
vatCode	String
vatCodeDesc	String
vatRate	BigDecimal

JSON Structure:

```
{
  "vatRegion": null,
  "vatRegionName": null,
  "vatRegionType": null,
  "vatType": null,
  "vatCode": null,
  "vatCodeDesc": null,
  "vatRate": null,
  "links": [],
  "hyperMediaContent": {
    "linkRDO": []
  }
}
```

Table Impact

TABLE	SELECT	INSERT	UPDATE	DELETE
V_DEPS	Yes	No	No	No
VAT_DEPS	Yes	No	No	No
VAT_REGION	Yes	No	No	No
V_VAT_REGION_TL	Yes	No	No	No
V_VAT_CODES_TL	Yes	No	No	No
VAT_CODE_RATES	Yes	No	No	No
JOB_AUDIT_LOGS	No	Yes	No	No

TABLE	SELECT	INSERT	UPDATE	DELETE
JOB_AUDIT_PAYLOAD	No	Yes	No	No

Warehouse Detail Service

This section describes the Warehouse Detail service.

Business Overview

Warehouse Detail service allows user to retrieve Warehouse information for a selected warehouse or for all warehouses.

Service Type

Get

ReST URL

Wh/whDetail?warehouse={whNumber}

Input Parameters

Parameter Name	Required	Description
Warehouse	No	Warehouse Number. If none is specified, all warehouses will be retrieved.

Output

RestWhRecRDO

Parameter Name	Data Type
warehouse	BigDecimal
warehouseName	String
warehouseSecondaryName	String
vatRegion	BigDecimal
organizationHierarchyType	BigDecimal
organizationHierarchyValue	BigDecimal
currencyCode	String
physicalWarehouse	BigDecimal
primaryVirtualWarehouse	BigDecimal
channelId	BigDecimal
stockholdingIndicator	String
breakPackIndicator	String
redistributeWarehouseIndicator	String

Parameter Name	Data Type
restrictedIndicator	String
protectedIndicator	String
transferEntityId	BigDecimal
finisherInd	String
inboundHandlingDays	BigDecimal
organizationalUnitId	BigDecimal
virtualWarehouseType	String
customerOrderLocationIndicator	String
address1	String
address2	String
address3	String
city	String
state	String
countryId	String
post	String
contactName	String
contactPhone	String
contactEmail	String

JSON Structure:

```
{
  "warehouse": null,
  "warehouseName": null,
  "warehouseSecondaryName": null,
  "vatRegion": "null,
  "organizationHierarchyType": null,
  "organizationHierarchyValue": null,
  "currencyCode": null,
  "physicalWarehouse": null,
  "primaryVirtualWarehouse": null,
  "channelId": null,
  "stockholdingIndicator": null,
  "breakPackIndicator": null,
  "redistributeWarehouseIndicator": null,
  "restrictedIndicator": null,
  "protectedIndicator": null,
  "transferEntityId": null,
  "finisherIndicator": null,
  "inboundHandlingDays": null,
  "organizationalUnitId" :null,
  "virtualWarehouseType" :null,
  "customerOrderLocationIndicator" :null,
  "address1": null,
  "address2": null,
  "address3": null,
  "city": null,
  "state": null,
  "countryId": null,
  "post": null,
  "contactName": null,
```

```
"contactPhone": null,  
"contactEmail": null,  
"links": [],  
"hyperMediaContent": {  
  "linkRDO": []  
}  
}
```

Table Impact

TABLE	SELECT	INSERT	UPDATE	DELETE
V_WH	Yes	No	No	No
ADDR	Yes	No	No	No
JOB_AUDIT_LOGS	No	Yes	No	No
JOB_AUDIT_PAYLOAD	No	Yes	No	No

Fiscal Document Generation Services

This section describes the Fiscal Document Generation Services

Fiscal Pre Document Request Service

Functional Area

Finance

Business Overview

This web service will support fiscal document requests from inventory systems. It will have a simplified payload for the fiscal document request. Once the shipment system calls for this service, Fiscal Document Generation will return an internal ID for the request. This ID can later be used by the shipment systems to request for the document(s) created for the specific shipment.

Service Type

Post

ReST URL

```
{{baseUrl}}/services/private/fdg/processPreDoc
```

Input

Parameter Name	Required	Data Type	Description	Valid values
requestSystem	Yes	String	This field contains the identification name of system that is requesting a fiscal document. This name is provided by system that is requesting.	
docType	Yes	String	This field contains the type of the document. Valid values are codes from CODE_DETAIL with a CODE_TYPE of FDTY.	INV, WBL.
extReqId	No	String	This field contains the external request ID. This is the internal number from system that is requesting a fiscal document. Example: asn_nbr ID from shipments.	
issueDate	Yes	OffsetDateTime	This field contains the issue date of document in the format YYYY-MM-DDThh:mm:ssTZD (eg 1997-07-16T19:20:30+01:00) equivalent to ISO 8601.	
transactionType	Yes	String	This field contains the code type of the transaction. Valid values are codes from CODE_DETAIL with a CODE_TYPE of GTRT.	ITMTAX, PO, TSF, DEAL, RTV, POSSAL, SALES, TAXINF, FO, FR.
currencyCode	Yes	String	This field contains the currency code for the fiscal document. It is based on currency code defined in shipment location that is going to ship goods.	
entities	Yes	List<RestFdgEtt>	Collection of entities.	
details	Yes	List<RestFdgDtl>	Collection of item details.	
comments	No	List<RestFdgText>	Collection of comments.	
references	No	List<RestFdgRef>	Collection of references.	

RestFdgEtt

The *fiscalDocument/entities* node has the following parameters:

Parameter Name	Required	Data Type	Description	Valid values
entityType	Yes	String	This field contains the entity type. Valid values are codes from CODE_DETAIL with a CODE_TYPE of FDED.	SHIPP, DEST, VENDOR, CR, LOC.
entityInternalType	No	String	This field contains the RMS entity type. Valid values are codes from CODE_DETAIL with a CODE_TYPE of FDET.	WH, ST, SUPP, PTNR, CUST.
entityInternalPtnrType	No	String	This field contains the RMS partner type. Valid values are codes from CODE_DETAIL with a CODE_TYPE of PTNR.	AG, AP, BK, BR, CN, CO, E, EC, ES, EV, FA, FF, IA, IM, S1, S2, S3.
entityCode	No	String	This field contains the entity's RMS internal code. Ex. ID of store or warehouse.	
name	No	String	This field contains the name responsible for transportation. This field is used only when there is no entity code defined for transportation. It's used by SIM shipments because there is no transportation entity.	
carrierVehicleRn	No	String	This field contains the vehicle registration number responsible for transportation.	
carrierVehicleState	No	String	This field contains the state of the country where vehicle, responsible for transportation, is registered.	
carrierDriverName	No	String	This field contains the driver name responsible for transportation.	
carrierDriverLn	No	String	This field contains the information about driver license ID responsible for transportation.	

RestFdgDtl

The *fiscalDocument/details* node has the following parameters:

Parameter Name	Required	Data Type	Description	Valid values
item	Yes	String	This field contains the unique code ID that identifies the item.	
uom	Yes	String	This field contains the unit of measure used in item in the fiscal document.	
qty	Yes	BigDecimal	This field contains the quantity of item in this fiscal document.	
references	No	RestFdgDtl	Collection of item references	

RestFdgRef

The *fiscalDocument/details/references* node has the following parameters:

Parameter Name	Required	Data Type	Description	Valid values
refType	Yes	String	This field contains the type of the reference at item level. It is used when there is any reference that it is necessary to add in this fiscal document, as shipment number, contract number, order, id, etc. Valid values: CODE_DETAIL – FDRT.	FDOC,ORDER , SHIPN,FULLF, ALLOC,DISTR O, BOLN,TSFNO, RTVNO,CON O RTVAUT,CTR C
refCode	Yes	String	This field contains the reference code/value of reference type at item level.	

RestFdgText

The *fiscalDocument/comments* node has the following parameters:

Parameter Name	Required	Data Type	Description	Valid values
text	Yes	String	This field contains additional comments at item level. It is used to provide any additional information or legal message important to fiscal document.	

Parameter Name	Required	Data Type	Description	Valid values
textGroupId	No	String	Text group ID. It is used to group related comments when there is more than one or when the size of a message is bigger than the size of the text field and it is necessary to generate more than one line.	
textGroupSeq	No	String	Sequence of the text for a given group ID. It is used in case comments information has more than one line.	

JSON Input Structure

```

"{
  "fiscalDocument": {
    "requestSystem": "SIOCS",
    "docType": "WBL",
    "extReqId": "10001",
    "transactionType": "TSF",
    "issueDate": "2020-12-16T08:19:42-05:00",
    "currencyCode": "USD",
    "entities": [
      {
        "entityType": "SHIPP",
        "entityInternalType": "WH",
        "entityCode": "252525"
      },
      {
        "entityType": "DEST",
        "entityInternalType": "ST",
        "entityCode": "9625518"
      },
      {
        "entityType": "CR",
        "name": "UPS",
        "carrierVehicleRn": "IF5597",
        "carrierVehicleState": "SP",
        "carrierDriverName": "Driver Name",
        "carrierDriverLn": "asasdd984955456"
      }
    ],
    "details": [
      {
        "item": "100450025",
        "uom": "EA",
        "qty": 5,
        "references": [
          {
            "refType": "TSFNO",
            "refCode": "897889"
          }
        ]
      }
    ]
  }
}

```

```

    }
  ]
}],
"comments": [{
  "text": "Additional comments to the document. If it's too l'",
  "textGroupId": "1",
  "textGroupSeq": "1"
},
{
  "text": "ong, it can end in other line",
  "textGroupId": "1",
  "textGroupSeq": "2"
}
],
"references": [
  {
    "refType": "DOC",
    "refCode": "9848945651"
  }
]
}
}"

```

Output

Parameter Name	Data Type	Description
requestId	String	This field contains the unique identifier of the document request.
requestSystem	String	This column contains the unique identifier of the requester.
requestAction	String	This column contains the current action for the request.
docId	BigDecimal	This column contains the unique document ID.
docNo	String	This column contains the unique document number.
docType	String	This column contains the type of the document.
status	String	This column contains the status of the document.
extReqId	String	This field contains the external request ID. This is the internal number from system that is requesting a fiscal document. Example: asn_nbr ID from shipments.
docPrintUrl	String	This column contains the url to access the fiscal document issued by government. It can be a url from third-party repository responsible to communicate with government or other url.
additionalproperties	List<RestFdgExt>	Collection of additional properties.

Parameter Name	Data Type	Description
errors	List<RestFdgError>	Collection of errors.

RestFdgExt

Parameter Name	Data Type	Description
key	String	This column contains the key value of the entity additional properties at entity address level.
value	String	This column contains the value of the entity additional properties at entity address level.
type	String	This column contains the extension type at entity address level.

RestFdgError

Parameter Name	Data Type	Description
errorCode	String	This field contains the error code.
errorDesc	String	This field contains the error description.

JSON Structure

```
"{
  "requestId": 62,
  "requestSystem": "SIOCS",
  "requestAction": "A",
  "docId": 62,
  "docNo": null,
  "docType": "WBL",
  "status": "N",
  "extReqId": "10003",
  "docPrintUrl": null,
  "additionalproperties": [],
  "errors": []
}"
```

Table Impact

TABLE	SELECT	INSERT	UPDATE	DELETE
FDG_ATTRIB_MAP	Yes	No	No	No
FDG_ATTRIB_MAP_NM	Yes	No	No	No
SVC_FDG_DTL	Yes	Yes	Yes	No
SVC_FDG_DTL_PACK	Yes	Yes	Yes	No
SVC_FDG_ERROR	Yes	Yes	No	No

TABLE	SELECT	INSERT	UPDATE	DELETE
SVC_FDG_ETT	Yes	Yes	Yes	No
SVC_FDG_EXT	Yes	Yes	Yes	No
SVC_FDG_HDR	Yes	Yes	Yes	No
SVC_FDG_NON_MERCH	Yes	Yes	Yes	No
SVC_FDG_REF	Yes	Yes	No	No
SVC_FDG_TAX	Yes	Yes	No	Yes
SVC_FDG_TEXT	Yes	Yes	No	No
CODE_DETAIL	Yes	No	No	No
V CODE_DETAIL	Yes	No	No	No
STORE	Yes	No	No	No
PARTNER	Yes	No	No	No
WH	Yes	No	No	No
LANG	Yes	No	No	No
FDG_SYSTEM_OPTIONS	Yes	No	No	No
CFA_ATTRIB	Yes	No	No	No
CFA_ATTRIB_MAP	Yes	No	No	No
CFA_ATTRIB_MAP_LABES	Yes	No	No	No
RTV_DETAIL	Yes	No	No	No
ORDCUST	Yes	No	No	No
ORDCUST_DETAIL	Yes	No	No	No
TSFHEAD	Yes	No	No	No
TSFDETAIL	Yes	No	No	No
SHIPSKU	Yes	No	No	No
TSFDETAIL_CHRG	Yes	No	No	No
ALLOC_CHRG	Yes	No	No	No
NON_MERCH_CODE_COMP	Yes	No	No	No
V_NON_MERCH_CODE_HEAD_TL	Yes	No	No	No
ADDR	Yes	No	No	No
PACKITEM_BREAKOUT	Yes	No	No	No
ITEM_MASTER	Yes	No	No	No
V_ITEM_MASTER	Yes	No	No	No
ITEM_SUPPLIER	Yes	No	No	No
GTS_CALC_INPUT_STG	Yes	Yes	No	No
GTS_CALC_OUTPUT_STG	Yes	No	No	No

Fiscal Document Request Service

Functional Area

Finance

Business Overview

This web service will be used to support fiscal document requests from any system based on a comprehensive payload with detailed level of input parameters. This web service is meant to support inbound transactions such as PO receiving. Once the requester system calls for this service, Fiscal Document Generation will return an internal ID for the request. This ID can later be used by these systems to request for the document(s) created.

Service Type

Post

ReST URL

```
{{baseUrl}}/services/private/fdg/processDoc
```

Input

Parameter Name	Required	Data Type	Description	Valid values
requestSystem	Yes	String	This field contains the identification name of system that is requesting a fiscal document. This name is provided by system that is requesting.	
docType	Yes	String	This field contains the type of the document. Valid values are codes from CODE_DETAIL with a CODE_TYPE of FDTY.	INV, WBL.
extReqId	No	String	This field contains the external request ID. This is the internal number from system that is requesting a fiscal document. Example: asn_nbr ID from shipments	
issueDate	Yes	OffsetDateTime	This field contains the issue date of document in the format YYYY-MM-DDThh:mm:ssTZD (eg 1997-07-16T19:20:30+01:00) equivalent to ISO 8601.	

Parameter Name	Required	Data Type	Description	Valid values
transactionType	Yes	String	This field contains the code type of the transaction. Valid values are codes from CODE_DETAIL with a CODE_TYPE of GTRT.	ITMTAX, PO, TSF, DEAL, RTV, POSSAL, SALES, TAXINF, FO, FR.
currencyCode	Yes	String	This field contains the currency code for the fiscal document. It is based on currency code defined in shipment location that is going to ship goods.	
deduceTax	Yes	String	This field contains the indicator to control in this request, if taxes is sent from request system and any calculation is needed or request system not send taxes and it is necessary to call tax calculation to fill taxes information.	Y or N.
entities	Yes	List<RestFdgEtt>	Collection of entities.	
details	Yes	List<RestFdgDtl>	Collection of item details.	
comments	No	List<RestFdgText>	Collection of comments.	
references	No	List<RestFdgRef>	Collection of references.	

RestFdgEtt

The *fiscalDocument/entities* node has the following parameters:

Parameter Name	Required	Data Type	Description	Valid values
entityType	Yes	String	This field contains the entity type. Valid values are codes from CODE_DETAIL with a CODE_TYPE of FDED.	SHIPP, DEST, VENDOR, CR, LOC.
entityInternalType	No	String	This field contains the RMS entity type. Valid values are codes from CODE_DETAIL with a CODE_TYPE of FDET.	WH, ST, SUPP, PTNR, CUST.
entityInternalPartner	No	String	This field contains the RMS partner type. Valid values are codes from CODE_DETAIL with a CODE_TYPE of PTNR.	AG, AP, BK, BR, CN, CO, E, EC, ES, EV, FA, FF, IA, IM, S1, S2, S3.

Parameter Name	Required	Data Type	Description	Valid values
entityCode	No	String	This field contains the entity's RMS internal code. Ex. ID of store or warehouse	
name	No	String	This field contains the name responsible for transportation. This field is used only when there is no entity code defined for transportation. It's used by SIM shipments because there is no transportation entity.	
carrierVehicleRn	No	String	This field contains the vehicle registration number responsible for transportation.	
carrierVehicleState	No	String	This field contains the state of the country where vehicle, responsible for transportation, is registered.	
carrierDriverName	No	String	This field contains the driver name responsible for transportation.	
carrierDriverLn	No	String	This field contains the information about driver license ID responsible for transportation.	

RestFdgDtl

The *fiscalDocument/details* node has the following parameters:

Parameter Name	Required	Data Type	Description	Valid values
Item	Yes	String	This field contains the unique code ID that identifies the item.	
totalAmt	Yes	BigDecimal	This field contains the total value of this item in the fiscal document.	
unitAmt	Yes	BigDecimal	This field contains the item unit cost of this item in the document.	
uom	Yes	String	This field contains the unit of measure used in item in the fiscal document.	
qty	Yes	BigDecimal	This field contains the quantity of item in this fiscal document.	

Parameter Name	Required	Data Type	Description	Valid values
totalTaxAmt	No	BigDecimal	This field contains the total value of item in this fiscal document.	
totalAmtInclTax	No	BigDecimal	This field contains the total value including all taxes value of item in this fiscal document.	
taxes	No	List<RestFdgTax>	Collection of item taxes.	
references	No	List<RestFdgRef>	Collection of item references.	
allowanceCharges	No	List<RestFdgNonMerch >	Collection of item allowance/changes.	

RestFdgTax

The *fiscalDocument/details/taxes* node has the following parameters:

Parameter Name	Required	Data Type	Description	Valid values
taxType	Yes	String	This field contains the tax type at item level.	U (for tax per value) or P (for tax per percentage)
taxCode	Yes	String	This field contains the tax identification code that should be applied at item level.	.
taxAmt	Yes	BigDecimal	This field contains the tax value at item level, when tax type is set as per value.	
taxCodeDesc	Yes	String	This field contains the tax code description at item level. This field can be used to indicate extra code or detailed information about tax code.	

RestFdgRef

The *fiscalDocument/details/references* node has the following parameters:

Parameter Name	Required	Data Type	Description	Valid values
refType	Yes	String	This field contains the type of the reference at item level. It is used when there is any reference that it is necessary to add in this fiscal document, as shipment number, contract number, order, id, etc. Valid values: CODE_DETAIL – FDRT.	FDOC,ORDER, SHIPN,FULL F, ALLOC,DIST RO, BOLN,TSFN O, RTVNO,CON O RTVAUT,CTR C
refCode	Yes	String	This field contains the reference code/value of reference type at item level.	

RestFdgNonMerch

The *fiscalDocument/details/allowanceCharges* node has the following parameters:

Parameter Name	Required	Data Type	Description	Valid values
chargeInd	Yes	String	This field contains the charge indicator at item allowance charges level.	Y (represent s a charge). N (represent s an allowance)
nonMerchCode	Yes	String	This field contains the non-merchandise code that identify non-merchandise cost at item allowance changes level. This value should be existing in table NON_MERCH_CODE_HEAD.	.
nonMerchAmt	Yes	BigDecimal	This field contains the non-merchandise value at item allowance changes level.	
taxes	No	List<RestFdgTax>	Collection of non-merchandise taxes.	

RestFdgTax

The *fiscalDocument/details/allowanceCharges/taxes* node has the following parameters:

Parameter Name	Required	Data Type	Description	Valid values
taxType	Yes	String	This field contains the tax type at item allowance charge taxes level.	U (for tax per value) or P (for tax per percentage)
taxCode	Yes	String	This field contains the tax identification code that should be applied at item allowance charges taxes level.	.
taxAmt	Yes	BigDecimal	This field contains the tax value at item allowance charges taxes level, when tax is set up as per value.	
taxCodeDesc	Yes	String	This field contains the tax code description at item allowance charges taxes level. This field can be used to indicate extra code or detailed information about tax code.	
comments	No	List<RestFdgText>	Collection of non-merchandise taxes comments.	

RestFdgText

The *fiscalDocument/details/allowanceCharges/taxes/comments* and *fiscalDocument/comments* node have the following parameters:

Parameter Name	Required	Data Type	Description	Valid values
text	Yes	String	This field contains additional comments at item level. It is used to provide any additional information or legal message important to fiscal document.	
textGroupId	No	String	Text group ID. It is used to group related comments when there is more than one or when the size of a message is bigger than the size of the text field and it is necessary to generate more than one line.	
textGroupSeq	No	String	Sequence of the text for a given group ID. It is used in case comments information has more than one line.	

RestFdgRef

The *fiscalDocument/references* node has the following parameters:

Parameter Name	Required	Data Type	Description	Valid values
refType	Yes	String	This field contains the type of the reference at item level. It is used when there is any reference that it is necessary to add in this fiscal document, as shipment number, contract number, order, id, etc. Valid values: CODE_DETAIL – FDRT.	FDOC,ORDE R, SHIPN,FULL F, ALLOC,DIST RO, BOLN,TSFN O, RTVNO,CON O RTVAUT,CTR C
refCode	Yes	String	This field contains the reference code/value of reference type at item level.	

JSON Input Structure

```
{
  "fiscalDocument": {
    "requestSystem": "SUPPLIER_PORTAL",
    "docType": "WBL",
    "extReqId": "1234",
    "transactionType": "TSF",
    "issueDate": "2020-12-16T08:19:42-05:00",
    "currencyCode": "USD",
    "deduceTax": "Y",
    "entities": [
      {
        "entityType": "SHIPP",
        "entityInternalType": "WH",
        "entityCode": "7001"
      },
      {
        "entityType": "DEST",
        "entityInternalType": "ST",
        "entityCode": "777701"
      },
      {
        "entityType": "CR",
        "entityInternalType": "PTNR",
        "entityInternalPtnrType": "FF",
        "entityCode": "11",
        "carrierVehicleRn": "IF5597",
        "carrierVehicleState": "SP",
        "carrierDriverName": "Driver Name",

```



```
        "carrierDriverLn": "asasdd984955456"
    }],
    "details": [
        {
            "item": "54987897",
            "totalAmt": 10,
            "unitAmt": null,
            "uom": "EA",
            "qty": 5,
            "totalTaxAmt": 7,
            "totalAmtInclTax": 57,
            "taxes": [
                {
                    "taxType": "P",
                    "taxCode": "GST",
                    "taxAmt": 5,
                    "taxCodeDesc": null
                },
                {
                    "taxType": "P",
                    "taxCode": "PST",
                    "taxAmt": 2,
                    "unitTaxAmt": null,
                    "taxCodeDesc": null
                }
            ],
            "references": [
                {
                    "refType": "ORDER",
                    "refCode": "897889"
                }
            ],
            "allowanceCharges": [
                {
                    "chargeInd": "Y",
                    "nonMerchCode": "FRHT",
                    "nonMerchAmt": 10,
                    "taxes": [
                        {
                            "taxType": "P",
                            "taxCode": "GST",
                            "taxAmt": 1.8,
                            "taxCodeDesc": null,
                            "comments": []
                        },
                        {
                            "taxType": "P",
                            "taxCode": "PST",
                            "taxAmt": 1,
                            "taxCodeDesc": null,
                            "comments": []
                        }
                    ]
                }
            ],
            "comments": [{"
```

```

        "text": "Open box Item"
      }
    ]
  ],
  "comments": [{
    "text": "Additional comments to the document. If it's too l'",
    "textGroupId": "1",
    "textGroupSeq": "1"
  },
  {
    "text": "ong, it can end in other line",
    "textGroupId": "1",
    "textGroupSeq": "2"
  }
  ],
  "references": [{
    "refType": "DOC",
    "refCode": "9848945651"
  }
  ]
}
}"

```

Output

Parameter Name	Data Type	Description
requestId	String	This field contains the unique identifier of the document request.
requestSystem	String	This column contains the unique identifier of the requester.
requestAction	String	This column contains the current action for the request.
docId	BigDecimal	This column contains the unique document ID.
docNo	String	This column contains the unique document number.
docType	String	This column contains the type of the document.
status	String	This column contains the status of the document.
extReqId	String	This field contains the external request ID. This is the internal number from system that is requesting a fiscal document. Example: asn_nbr ID from shipments.
docPrintUrl	String	This column contains the url to access the fiscal document issued by government. It can be a url from third-party repository responsible to communicate with government or other url.
additionalproperties	List<RestFdgExt>	Collection of additional properties.

Parameter Name	Data Type	Description
errors	List<RestFdgError >	Collection of errors.

JSON Structure

```
"{
  "requestId": 62,
  "requestSystem": "SIOCS",
  "requestAction": "A",
  "docId": 62,
  "docNo": null,
  "docType": "WBL",
  "status": "N",
  "extReqId": "10003",
  "docPrintUrl": null,
  "additionalproperties": [],
  "errors": []
}"
```

Table Impact

TABLE	SELECT	INSERT	UPDATE	DELETE
FDG_ATTRIB_MAP	Yes	No	No	No
FDG_ATTRIB_MAP_NM	Yes	No	No	No
SVC_FDG_DTL	Yes	Yes	Yes	No
SVC_FDG_DTL_PACK	Yes	Yes	Yes	No
SVC_FDG_ERROR	Yes	Yes	No	No
SVC_FDG_ETT	Yes	Yes	Yes	No
SVC_FDG_EXT	Yes	Yes	Yes	No
SVC_FDG_HDR	Yes	Yes	Yes	No
SVC_FDG_NON_MERCH	Yes	Yes	No	No
SVC_FDG_REF	Yes	Yes	No	No
SVC_FDG_TAX	Yes	Yes	No	No
SVC_FDG_TEXT	Yes	Yes	No	No
CODE_DETAIL	Yes	No	No	No
V_CODE_DETAIL_TL	Yes	No	No	No
STORE	Yes	No	No	No
PARTNER	Yes	No	No	No
WH	Yes	No	No	No
LANG	Yes	No	No	No
FDG_SYSTEM_OPTIONS	Yes	No	No	No
CFA_ATTRIB	Yes	No	No	No
CFA_ATTRIB_MAP	Yes	No	No	No

TABLE	SELECT	INSERT	UPDATE	DELETE
CFA_ATTRIB_MAP_LABES	Yes	No	No	No
RTV_DETAIL	Yes	No	No	No
ORDCUST	Yes	No	No	No
ORDCUST_DETAIL	Yes	No	No	No
TSFHEAD	Yes	No	No	No
TSFDETAIL	Yes	No	No	No
SHIPSKU	Yes	No	No	No
ADDR	Yes	No	No	No
PACKITEM_BREAKOUT	Yes	No	No	No
ITEM_MASTER	Yes	No	No	No
V_ITEM_MASTER	Yes	No	No	No
ITEM_SUPPLIER	Yes	No	No	No

Fiscal Document Status Detail Request Service

Functional Area

Finance

Business Overview

In both scenarios for shipment and Inbound, it will be the requestor's responsibility to pull the approved documents from Fiscal Document Generation in order to conclude their transactions. This web service will be used to provide the status of fiscal documents requested via the Pre Document and Document Request Services. In case the document is approved, approval related data will also be provided such as the formal document number and print version URL path.

Service Type

Get

ReST URL

```
{{baseUrl}}/services/private/fdg/documentStatusDetail?
requestId=<string>&requestSystem=<string>&entityType=<string>&entityInt
ernalType=<string>&entityCode=<string>&country=<string>
```

Input Parameters

Parameter Name	Required	Data Type	Description	Valid values
requestId	Yes	String	This column contains the unique identifier of the document request.	
requestSystem	No	String	This column contains the unique identifier of the requester.	
entityType	No	String	This field contains the entity type. Valid values are codes from CODE_DETAIL with a CODE_TYPE of FDED.	SHIPP, DEST, VENDOR, CR, LOC.
entityInternalType	No	String	This field contains the RMS entity type. Valid values are codes from CODE_DETAIL with a CODE_TYPE of FDET.	WH, ST, SUPP, PTNR, CUST.
entityCode	No	String	This column contains the entity's RMS internal code.	
country	No	String	This column contains the country where is the location address.	

Output

Parameter Name	Data Type	Description
requestId	String	Internal fiscal document REQUEST ID generated by FDG. This ID can be used to identify the fiscal document(s) being processed.
requestSystem	String	Name of the system that performed the request for a fiscal document.
requestAction	String	Action code to be passed along with the request for a fiscal document process.
docNo	String	Official fiscal document number. This number is given to approved documents.
docType	String	Valid values: INV (Invoice), WBL (eWaybill), NFe (Nota Fiscal Eletronica).
status	String	Internal FDG status of the fiscal document.
extReqId	String	External system's request ID used to link FDG internal ID with the requestor system's transaction ID.
docPrintUrl	String	Url of the fiscal document report.

Parameter Name	Data Type	Description
additionalproperties	List<RestFdgExt>	Collection of additional properties.
errors	List<RestFdgError>	Collection of errors.

Table 5-24 RestFdgExt

Parameter Name	Data Type	Description
key	String	This column contains the key value of the entity additional properties at entity address level.
value	String	This column contains the value of the entity additional properties at entity address level.
type	String	This column contains the extension type at entity address level.

Table 5-25 RestFdgError

Parameter Name	Data Type	Description
errorCode	String	This field contains the error code.
errorDesc	String	This field contains the error description.

JSON Structure

```
"{
  "requestId": 50039,
  "requestSystem": "SUPPLIER_PORTAL",
  "requestAction": "A",
  "docId": 50038,
  "docNo": null,
  "docType": "WBL",
  "status": "S",
  "extReqId": "10000600",
  "docPrintUrl": null,
  "additionalproperties": [],
  "errors": []
}"
```

Table Impact

TABLE	SELECT	INSERT	UPDATE	DELETE
SVC_FDG_ERROR	Yes	No	No	No
SVC_FDG_EXT	Yes	No	No	No
SVC_FDG_HDR	Yes	No	No	No
FDG_ERROR	Yes	No	No	No

TABLE	SELECT	INSERT	UPDATE	DELETE
FDG_EXT	Yes	No	No	No
FDG_HDR	Yes	No	No	No

Next Fiscal Document Request Service

Functional Area

Finance

Business Overview

This is the service to be called by the 3rd party system in charge of communicating fiscal document requests to local Governments to fetch documents ready to be approved.

Service Type

Get

ReST URL

```
{{baseUrl}}/services/private/fdg/nextDocumentDetail?
docId=<string>&docType=<string>&entityType=<string>&internalType=<string>&int
ernalCode=<string>&integrationSystemName=<string>&integrationSystemId=<string
>
```

Input Parameters

Parameter Name	Required	Data Type	Description	Valid values
docId	No	String	Internal fiscal document ID generated by FDG. This ID is unique to each fiscal document created.	
docType	No	String	This field contains the type of the document. Valid values are codes from CODE_DETAIL with a CODE_TYPE of FDTY.	INV, WBL
entityType	No	String	This field contains the entity type. Valid values are codes from CODE_DETAIL with a CODE_TYPE of FDED.	SHIPP, DEST, VENDOR, CR, LOC.
internalType	No	String	Filter criteria to fetch documents based on specific entity types in RMS irrespective of their role in the fiscal documents.	WH, ST, SUPP, PTRN, CUST.

Parameter Name	Required	Data Type	Description	Valid values
internalCode	No	String	Filter criteria to fetch documents based on a specific entity code. This field contains the entity's RMS internal code and must be sent in conjunction with entityInternatType tag. Ex. ID of store or warehouse	
integrationSystemName	Yes	String	Name of the integration system that will process this document.	
integrationSystemId	Yes	String	Unique identifier to be used by the integration system.	

Output

Table 5-26 Output

Parameter Name	Data Type	Description
requestId	String	Internal fiscal document REQUEST ID generated by FDG. This ID can be used to identify the fiscal document(s) being processed.
requestSystem	String	Name of the system that performed the request for a fiscal document.
requestAction	String	Action code to be passed along with the request for a fiscal document process.
docId	BigDecimal	Internal fiscal document ID generated by FDG. This ID is unique to each fiscal document created.
docNo	String	Official fiscal document number. This number is given to approved documents.
status	String	Internal FDG status of the fiscal document.
docType	String	Valid values: INV (Invoice), WBL (eWaybill), NFe (Nota Fiscal Eletronica).
docTypeCode	String	Extended description of a document type.
extReqId	String	External system's request ID used to link FDG internal ID with the requestor system's transaction ID.
transactionType	String	Identification of the transaction associated with the fiscal document.

Table 5-26 (Cont.) Output

Parameter Name	Data Type	Description
issueDate	OffsetDateTime	Issue date of the fiscal document. Date defined at the time the document is submitted for approval.
dueDate	date	Payment date of fiscal document or invoice.
docLang	String	Language of fiscal document. It is based on language defined in source location.
printerId	String	Printer ID that is designated to print fiscal document.
docPrintUrl	String	Url of the fiscal document report.
currencyCode	String	Currency code for the fiscal document. It is based on currency code defined in source location.
totalTaxAmt	BigDecimal	Total TAX amount of the fiscal document. This field is the sum of totalTaxAmt of all itens of the document. Rounding applied based on system option.
totalNonMerchTaxAmt	BigDecimal	Total TAX amount of allowances/charges of the fiscal document. Rounding applied based on system option.
totalAmtInclTax	BigDecimal	Total value of fiscal document including taxes. This field is the sum of totalAmtInclTax of all itens of the document. Rounding applied based on system option.
totalAmtExclTax	BigDecimal	Total value of fiscal document excluding taxes. This field has the totalAmtInclTax minus totalTaxAmt. Rounding applied based on system option.
totalAllowance	BigDecimal	Total value of all allowances in fiscal document. This field is the sum of totalAllowance of all itens of the document. Rounding applied based on system option.
totalCharge	BigDecimal	Total value of all charges in fiscal document. This field is the sum of totalCharge of all itens of the document. Rounding applied based on system option.
totalRounding	BigDecimal	The rounding amount (positive or negative) added to produce the line extension totalAmt. This field is the sum of totalRounding of all itens of the document.

Table 5-26 (Cont.) Output

Parameter Name	Data Type	Description
terms	String	Payment terms used in this transaction
deduceTax	String	Flag that indicates if taxes were calculated by the system or informed at the fiscal document request.
createBy	String	User ID that created the record.
createDatetime	date	Record creation date.
lastUpdateBy	String	User ID that did last update of fiscal document.
lastUpdateDatetime	date	Date of last update of fiscal document.
entities	List<RestFdgEtt>	Collection of entities.
details	List<RestFdgDtl>	Collection of item details

Table 5-27 RestFdgEtt

Parameter Name	Data Type	Description
entityType	String	Entity type.
entityInternalPtnrType	String	RMS partner type.
entityInternalType	String	RMS entity type.
entityCode	String	Internal RMS code of the entity. Ex. ID of store or warehouse
name	String	Entity name.
legalName	String	Entity's legal name.
carrierVehicleRn	String	Carrier vehicle's registration number.
carrierVehicleState	String	Carrier vehicle's state.
carrierDriverName	String	Carrier driver's name.
carrierDriverLn	String	Carrier driver's license number.
address	RestFdgEttAddr	Collection of address details.
identification	List<RestFdgExt>	Collection of Entity Identification Additional Properties. Fiscal Attributes will be send in this group.

Table 5-28 RestFdgEttAddr

Parameter Name	Data Type	Description
street	String	Street name of the entity's address.
additionalStreetName	String	Extension of street name of the entity's address.

Table 5-28 (Cont.) RestFdgEttAddr

Parameter Name	Data Type	Description
complement	String	Additional information of entity's address, such as neighborhood, county, etc.
city	String	City name of entity's address.
jurisdictionCode	String	Jurisdiction code of entity's address. This can be legal code associated to city, county or any other level.
postalCode	String	Zip or postal code of entity's address.
country	String	Country name of entity's address.
telephone	String	Telephone number of the person who is the contact point.
email	String	Email address of the person who is the contact point.
state	String	State name of entity's address.
contactName	String	Contact name.
additionalproperties	List<RestFdgExt>	Collection of address additional properties.

Table 5-29 RestFdgExt

Parameter Name	Data Type	Description
key	String	This column contains the key value of the entity additional properties at entity address level.
value	String	This column contains the value of the entity additional properties at entity address level.
type	String	This column contains the extension type at entity address level.
description	String	Description of additional property data.

Table 5-30 RestFdgExt (identification)

Parameter Name	Data Type	Description
key	String	This column contains the key value of the entity additional properties at entity address level.
value	String	This column contains the value of the entity additional properties at entity address level.
type	String	This column contains the extension type at entity address level.
description	String	Description of additional property data.

Table 5-30 (Cont.) RestFdgExt (identification)

Parameter Name	Data Type	Description
references	List<RestFdgRef>	Collection of references.
totaltaxes	List<RestFdgTax>	Collection of taxes.
errors	List<RestFdgError>	Collection of errors.
allowanceCharges	List<RestFdgNonMerc h>	Collection of allowance charges.
additionalproperties	List<RestFdgExt>	Collection of additional properties.
comments	List<RestFdgText>	Collection of comments.

Table 5-31 RestFdgDtl

Parameter Name	Data Type	Description
lineNo	long	Sequential number of the item in fiscal document. This number is a sequence starts with number one. Any gap is not allowed.
Item	String	Unique code ID that identifies the item.
itemDesc	String	Long description of the item.
itemShortDesc	String	Short description of the item.
totalAmt	BigDecimal	Total amount for the item inclusive of allowances and charges. Calculation formula is ((unitAmt*qty) - totalAllowance + totalCharge). Rounding applied based on system option.
unitAmt	BigDecimal	Item unit amount. No rounding applied.
uom	String	Unit of measure used in item in the fiscal document.
qty	BigDecimal	Quantity of item in the fiscal document.
totalTaxAmt	BigDecimal	Total amount of taxes applied to the item. This field is sum of all taxAmt at tax level for this item. Rounding applied based on system option.
totalAmtInclTax	BigDecimal	Total amount for item inclusive of taxes. Calculation formula is totalAmt + totalTaxAmt. Rounding applied based on system option.
totalRounding	BigDecimal	The rounding amount (positive or negative) added to produce the line extension totalAmt. Calculation formula ((unitAmt*qty) - totalAllowance + totalCharge)- totalAmt.

Table 5-31 (Cont.) RestFdgDtl

Parameter Name	Data Type	Description
totalAllowance	BigDecimal	Total amount of all allowances at item level. This field is sum of all allowances at allowanceCharges level for this item. Rounding applied based on system option.
totalCharge	BigDecimal	Total amount of all charges at item level. This field is sum of all charges at allowanceCharges level for this item. Rounding applied based on system option.
packInd	String	Pack indicator. Valid values: Y (this item is a pack) or N (this item is not a pack).
gtin	String	GTIN number (Global Trade Item Number) of item.
gtinType	String	GTIN Type. Valid values: GTIN-8, GTIN-12, GTIN-13, GTIN-14
vpn	String	Vendor Product Number (VPN) associated with the item.
taxes	List<RestFdgTax>	Collection of item taxes.
references	List<RestFdgRef>	Collection of item references
allowanceCharges	List<RestFdgNonMerch>	Collection of item allowance charges.
packdetails	List<RestFdgDtlPack>	Collection of item pack details.
additionalproperties	List<RestFdgExt>	Collection of item additional properties.
comments	List<RestFdgText>	Collection of item comments.

Table 5-32 RestFdgTax

Parameter Name	Data Type	Description
taxType	String	Tax type.
taxCode	String	Tax identification code.
taxRate	BigDecimal	Tax rate when tax type is set as percentage.
taxAmt	BigDecimal	Total tax amount for the tax code. Rounding applied based on system option.
unitTaxAmt	BigDecimal	Tax amount per unit for the tax code.
taxBasis	BigDecimal	Tax basis amount used to calculate tax value based on taxRate field. Rounding applied based on system option.

Table 5-32 (Cont.) RestFdgTax

Parameter Name	Data Type	Description
taxCodeDesc	String	Tax code description. This field can be used to indicate extra code or detailed information about tax code.
additionalproperties	List<RestFdgExt>	Collection of taxes addition properties.
comments	List<RestFdgText>	Collection of taxes comments.

Table 5-33 RestFdgText

Parameter Name	Data Type	Description
text	String	Additional information and/or comments. It is used to provide any additional information or legal message important to the fiscal document.
textGroupId	String	Text group ID. It is used to group related comments when there is more than one or when the size of a message is bigger than the size of the text field and it is necessary to generate more than one line.
textGroupSeq	String	Sequence of the text for a given group ID. It is used in case comments information has more than one line.

Table 5-34 RestFdgRef

Parameter Name	Data Type	Description
refType	String	Type of the reference information. It is used when there is any reference that it is necessary to add in the fiscal document, as shipment number, contract number, order, id, etc. Valid values are pre-defined in RMS System.
refCode	String	Reference code/value of reference type.

Table 5-35 RestFdgNonMerch

Parameter Name	Data Type	Description
chargeInd	String	Charge indicator.
nonMerchCode	String	Non-merchandise code that identifies a non-merchandise cost used as a charge or allowance.

Table 5-35 (Cont.) RestFdgNonMerch

Parameter Name	Data Type	Description
nonMerchCodeDesc	String	Non-merchandise code description. It is used to provide more detailed about non-merchandise code, for example: allowance details, types of non-merchandise code such as Freight, Insurance, etc.
nonMerchAmt	BigDecimal	Non-merchandise amount.
nonMerchType	String	Non-merchandise type exclusive for Charges.
taxes	List<RestFdgTax>	Collection of taxes for non-merchandise.
additionalproperties	List<RestFdgExt>	Collection of additional properties for non-merchandise.
comments	List<RestFdgText>	Collection of comments for non-merchandises.

Table 5-36 packdetails

Parameter Name	Data Type	Description
item	String	Unique alphanumeric ID of a pack component that is part of item pack.
itemQty	BigDecimal	Quantity of the component item that is part of the pack.
itemDesc	String	Description of the component item component.
uom	String	Unit of measure of the component item.

Table 5-37 totaltaxes

Parameter Name	Data Type	Description
taxType	String	Not applicable for totalTaxes.
taxCode	String	Tax identification code.
taxRate	BigDecimal	Not applicable for totalTaxes.
taxAmt	BigDecimal	Sum of taxAmt for all items for the tax code. Rounding applied based on system option
unitTaxAmt	BigDecimal	Not applicable for totalTaxes.
taxBasis	BigDecimal	Sum of taxBasis for all items for the tax code. Rounding applied based on system option.
taxCodeDesc	String	Tax code description. This field can be used to indicate extra code or detailed information about tax code.

JSON Structure

```
{
  "fiscalDocument": {
    "docId": 17200,
    "requestId": 17207,
    "requestSystem": "OMNI",
    "requestAction": "A",
    "status": "R",
    "docNo": "17200",
    "docType": "WBL",
    "docTypeCode": null,
    "extReqId": "10065007",
    "transactionType": "SALES",
    "issueDate": 1617196782000,
    "dueDate": null,
    "printerId": null,
    "docPrintUrl": null,
    "currencyCode": "USD",
    "totalTaxAmt": null,
    "totalNonMerchTaxAmt": null,
    "totalAmtInclTax": null,
    "totalAmtExclTax": null,
    "totalAllowance": null,
    "totalCharge": null,
    "totalRounding": null,
    "terms": null,
    "docLang": "EN",
    "deduceTax": null,
    "createBy": "RMS_ADMIN",
    "createDatetime": "2021-06-16",
    "lastUpdateBy": "RFM01",
    "lastUpdateDatetime": "2021-06-16",
    "entities": [
      {
        "entityType": "SHIPP",
        "entityInternalType": "ST",
        "entityInternalPtnrType": null,
        "entityCode": "990011",
        "name": "QABR-Store Test Washington",
        "legalName": "QABR-Store Test Washington",
        "carrierVehicleRn": null,
        "carrierVehicleState": null,
        "carrierDriverName": null,
        "carrierDriverLn": null,
        "address": {
          "street": "R da Juta",
          "additionalStreetName": "1512",
          "complement": null,
          "city": "Washijngton",
          "jurisdictionCode": null,
          "postalCode": "03308070",
          "country": "US",
```



```
        "telephone": "2360873",
        "email": "washington.dias@oracle.com",
        "state": "DC",
        "contactName": "Washington Dias",
        "additionalproperties": []
    },
    "identification": []
},
{
    "entityType": "DEST",
    "entityInternalType": "CUST",
    "entityInternalPtnrType": null,
    "entityCode": "75",
    "name": "wdias losnisk",
    "legalName": "JADLOG",
    "carrierVehicleRn": null,
    "carrierVehicleState": null,
    "carrierDriverName": null,
    "carrierDriverLn": null,
    "address": {
        "street": "end10",
        "additionalStreetName": "end20",
        "complement": "end30",
        "city": "Washington",
        "jurisdictionCode": "3100104",
        "postalCode": "08573160",
        "country": "US",
        "telephone": null,
        "email": null,
        "state": "DC",
        "contactName": "wdias losnisk",
        "additionalproperties": []
    },
    "identification": []
},
{
    "entityType": "CR",
    "entityInternalType": "PTNR",
    "entityInternalPtnrType": "FF",
    "entityCode": "11",
    "name": "Carrier",
    "legalName": "Carrier",
    "carrierVehicleRn": "CJU0939",
    "carrierVehicleState": "São Paulo",
    "carrierDriverName": "Djalma da Carreta",
    "carrierDriverLn": "77799573959957",
    "address": {
        "street": "15333 JFK Blvd",
        "additionalStreetName": null,
        "complement": null,
        "city": "Houston",
        "jurisdictionCode": null,
        "postalCode": "77032",
        "country": "US",
        "telephone": null,
    }
}
```

```
"email": "juliano.costa@oracle.com",
"state": "TX",
"contactName": "Juliano Costa",
"additionalproperties": [
  {
    "key": "NIP",
    "value": "NIP09",
    "type": "VARCHAR2",
    "description": "NIP"
  },
  {
    "key": "POSTALCODE",
    "value": "POST09",
    "type": "VARCHAR2",
    "description": "Postal Code"
  }
]
},
"identification": [
  {
    "key": "TAX_ID",
    "value": "332424324",
    "type": "VARCHAR2",
    "description": "TAX ID"
  }
]
}
],
"details": [
  {
    "lineNo": 1,
    "item": "100900006",
    "itemDesc": "item regular scenario 3",
    "itemShortDesc": "regular item regular scenario 3",
    "totalAmt": null,
    "unitAmt": null,
    "uom": "EA",
    "qty": 5,
    "totalTaxAmt": null,
    "totalAmtInclTax": null,
    "packInd": "N",
    "gtin": null,
    "gtinType": "ITEM",
    "vpn": null,
    "totalRounding": null,
    "totalAllowance": null,
    "totalCharge": null,
    "taxes": [],
    "references": [],
    "allowanceCharges": [],
    "packdetails": [],
    "additionalproperties": [],
    "comments": []
  },
  {
```



```

    }
  }"

```

Table Impact

TABLE	SELECT	INSERT	UPDATE	DELETE
SVC_FDG_DTL	Yes	No	No	No
SVC_FDG_DTL_PACK	Yes	No	No	No
SVC_FDG_ERROR	Yes	No	No	No
SVC_FDG_ETT	Yes	No	No	No
SVC_FDG_EXT	Yes	No	No	No
SVC_FDG_HDR	Yes	No	Yes	No
SVC_FDG_NON_MERCH	Yes	No	No	No
SVC_FDG_REF	Yes	No	No	No
SVC_FDG_TAX	Yes	No	No	No
SVC_FDG_TEXT	Yes	No	No	No
FDG_DTL	Yes	No	No	No
FDG_DTL_PACK	Yes	No	No	No
FDG_ERROR	Yes	No	No	No
FDG_ETT	Yes	No	No	No
FDG_EXT	Yes	No	No	No
FDG_HDR	Yes	No	No	No
FDG_NON_MERCH	Yes	No	No	No
FDG_REF	Yes	No	No	No
FDG_TAX	Yes	No	No	No
FDG_TEXT	Yes	No	No	No

Fiscal Document Process Status Request Service

Functional Area

Finance

Business Overview

This web service will be used by 3rd party system in charge of communicating fiscal document request to local Governments to provide to Fiscal Document Generation the approval status and associated data, such as the formal document number and print version URL path.

Service Type

Post

ReST URL

```
{{baseUrl}}/services/private/fdg/processDocStatus
```

Input Parameters

Parameter Name	Required	Data Type	Description	Valid values
integrationSystemName	Yes	String	Name of the integration system that will process this document.	
integrationSystemId	Yes	String	Unique identifier to be used by the integration system.	
requestAction	No	String	Action code returned for a given document in the nextDocumentDetail response.	A (approve) or C (cancel).
docId	Yes	String	Document ID returned for a given document in nextDocumentDetail response.	
docNo	Yes	String	Document number returned for a given document in nextDocumentDetail response.	
operationStatus	Yes	String	Status of the processing executed by third party for the given document. Success means that request was executed properly, and fail means that request was not completed. In this case it is expected the error or errors to be informed.	S (Success) or F (Fail).
extDocNo	No	String	External document number when applicable. This field should be populated with the approved fiscal document number when it is generated by Government.	
docPrintUrl	No	String	Url to access or download the fiscal document printable version (pdf).	
additionalproperties	No	List<RestFdgExt>	Collection of additional properties.	
errors	No	List<RestFdgError>	Collection of errors.	

Table 5-38 RestFdgExt

Parameter Name	Required	Data Type	Description	Valid values
key	No	String	Key value of the additional property data.	
value	No	String	Value of the additional property data.	
type	No	String	Data type of the additional property.	VARCHAR 2, NUMBER or DATE
description	No	String	Description of additional property data.	

Table 5-39 RestFdgError

Parameter Name	Required	Data Type	Description	Valid values
errorCode	Yes	String	Error code defined by 3rd party.	
errorDesc	Yes	String	Error description defined by 3rd party. This description must have enough detail so the problem can be identified.	

JSON Input Structure

```
{
  "integrationSystemName": "3RDPARNER",
  "integrationSystemId": 777,
  "requestAction": "A",
  "docId": 669854788524448800,
  "docNo": 1235489884,
  "operationStatus": "S",
  "extDocNo": 999999999999,
  "docPrintUrl": "www.doc.com/print?code=123",
  "additionalproperties": [
    {
      "key": "EXTID1",
      "value": "ExtensionValue",
      "type": "VARCHAR",
      "description": "Description"
    }
  ],
  "errors": [
    {
      "errorCode": "ERR87896",
      "errorDesc": "Unknow Error"
    }
  ]
}
```

```
    }}
  }"
```

Output

Parameter Name	Data Type	Description
status	String	Request process status. Can be either successfully or with error.
errors	List<RestFdgError>	Collection of errors.

Table 5-40 RestFdgError

Parameter Name	Data Type	Description
errorCode	String	This field contains the error code.
errorDesc	String	This field contains the error description.

JSON Structure

```
"{
  "status": "E",
  "errors": [
    {
      "errorCode": "FDG_DOC_NOT_FOUND",
      "errorDesc": "Fiscal document with id 669854788524448800 not found."
    }
  ]
}"
```

Table Impact

TABLE	SELECT	INSERT	UPDATE	DELETE
SVC_FDG_DTL	Yes	No	No	Yes
SVC_FDG_DTL_PACK	Yes	No	No	Yes
SVC_FDG_ERROR	Yes	Yes	No	Yes
SVC_FDG_ETT	Yes	No	No	Yes
SVC_FDG_EXT	Yes	Yes	No	Yes
SVC_FDG_HDR	Yes	No	Yes	Yes
SVC_FDG_NON_MERCH	Yes	No	No	Yes
SVC_FDG_REF	Yes	No	No	Yes
SVC_FDG_TAX	Yes	No	No	Yes
SVC_FDG_TEXT	Yes	No	No	Yes
FDG_DTL	No	Yes	No	No
FDG_DTL_PACK	No	Yes	No	No
FDG_ERROR	No	Yes	No	No
FDG_ETT	No	Yes	No	No

TABLE	SELECT	INSERT	UPDATE	DELETE
FDG_EXT	No	Yes	No	No
FDG_HDR	Yes	Yes	No	No
FDG_NON_MERCH	No	Yes	No	No
FDG_REF	No	Yes	No	No
FDG_TAX	No	Yes	No	No
FDG_TEXT	No	Yes	No	No
FDG_SYSTEM_OPTIONS	No	No	No	No

Fiscal Document Cancel Request Service

Functional Area

Finance

Business Overview

In both scenarios for shipment and Inbound, it will be the requestor's option to cancel a fiscal document request. This process will be made by calling the Document Cancel Request Service.

Requestor systems can request cancellation for a specific document. This request will be consumed by Fiscal Document Generation and, depending on the current status of the document, different actions will take place, including simply changing the status of the document to Canceled, or even the integration of the cancellation request to the 3rd party system that communicates this request to local Governments. In this scenario, Fiscal Document Generation will expect the 3rd party to confirm the cancellation.

Service Type

Post

ReST URL

```
{{baseUrl}}/services/private/fdg/processDocCancel
```

Input

Parameter Name	Required	Data Type	Description	Valid values
docId	Yes	String	This column contains the unique document ID.	

Parameter Name	Required	Data Type	Description	Valid values
requestAction	Yes	String	This field contains the current action for the request. This action is to third-party know the action to perform with this request.	A (Approve) or C (Cancel).
additionalproperties	No	List<RestFdgExt>	Collection of additional properties.	
comments	No	List<RestFdgText>	Collection of comments.	

Table 5-41 RestFdgExt

Parameter Name	Required	Data Type	Description	Valid values
key	No	String	Key value of the additional property data.	
value	No	String	Value of the additional property data.	
type	No	String	Data type of the additional property.	VARCHAR 2, NUMBER or DATE
description	No	String	Description of additional property data.	

Table 5-42 RestFdgText

Parameter Name	Required	Data Type	Description	Valid values
text	Yes	String	This field contains additional comments at item level. It is used to provide any additional information or legal message important to fiscal document.	
textGroupId	No	String	Text group ID. It is used to group related comments when there is more than one or when the size of a message is bigger than the size of the text field and it is necessary to generate more than one line.	
textGroupSeq	No	String	Sequence of the text for a given group ID. It is used in case comments information has more than one line.	

JSON Input Parameters

```
"{
  "docId": "12345",
  "requestAction": "C",
  "additionalproperties": [
    {
      "key": "EXTID1",
      "value": "ExtensionValue",
      "type": "VARCHAR2"
    }
  ],
  "comments":
  [
    {
      "text": "additional text"
    }
  ]
}"
```

Output

Parameter Name	Data Type	Description
processStatus	String	This column contains the status of the document
errors	List<RestFdgError>	Collection of errors.

Table 5-43 RestFdgError

Parameter Name	Data Type	Description
errorCode	String	This field contains the error code.
errorDesc	String	This field contains the error description.

JSON Structure

```
"{
  "status": "E",
  "errors": [
    {
      "errorCode": "FDG_DOC_NOT_FOUND",
      "errorDesc": "Fiscal document with id 12345 not found."
    }
  ]
}"
```

Table Impact

TABLE	SELECT	INSERT	UPDATE	DELETE
SVC_FDG_HDR	No	No	Yes	No

TABLE	SELECT	INSERT	UPDATE	DELETE
FDG_HDR	Yes	No	Yes	No
FDG_EXT	No	Yes	No	No
FDG_TEXT	No	Yes	No	No

Fiscal Document Detail Request Service

Functional Area

Finance

Business Overview

This web service is to be called by 3rd party systems in charge of communicating fiscal document requests to local Governments when they need to fetch data for a particular document. The response to this service will have the full payload of a fiscal document.

Service Type

Get

ReST URL

```
{{baseUrl}}/services/private/fdg/documentDetail?
docId=<string>&entityType=<string>&entityInternalType=<string>&entityCode=<string>
```

Input Parameters

Parameter Name	Required	Data Type	Description	Valid values
docId	No	String	Internal fiscal document ID generated by FDG. This ID is unique to each fiscal document created.	
entityType	No	String	Filter criteria to fetch documents for specific entity types in a fiscal document.	SHIPP, DEST, CR, VENDOR, INV, LOC.
entityInternalType	No	String	Filter criteria to fetch documents based on specific entity types in RMS irrespective of their role in the fiscal documents.	WH, ST, SUPP, PTNR, CUST.

Parameter Name	Required	Data Type	Description	Valid values
entityCode	No	String	Filter criteria to fetch documents based on a specific entity code. This field contains the entity's RMS internal code and must sent in conjunction with entityInternatType tag. Ex. ID of store or warehouse.	

Output

Parameter Name	Data Type	Description
requestId	String	Internal fiscal document REQUEST ID generated by FDG. This ID can be used to identify the fiscal document(s) being processed.
requestSystem	String	Name of the system that performed the request for a fiscal document.
requestAction	String	Action code to be passed along with the request for a fiscal document process.
docId	BigDecimal	Internal fiscal document ID generated by FDG. This ID is unique to each fiscal document created.
docNo	String	Official fiscal document number. This number is given to approved documents.
status	String	Internal FDG status of the fiscal document.
docType	String	Valid values: INV (Invoice), WBL (eWaybill), NFe (Nota Fiscal Eletronica).
docTypeCode	String	Extended description of a document type.
extReqId	String	External system's request ID used to link FDG internal ID with the requestor system's transaction ID.
transactionType	String	Identification of the transaction associated with the fiscal document.
issueDate	OffsetDateTime	Issue date of the fiscal document. Date defined at the time the document is submitted for approval.
dueDate	date	Payment date of fiscal document or invoice.
docLang	String	Language of fiscal document. It is based on language defined in source location.

Parameter Name	Data Type	Description
printerId	String	Printer ID that is designated to print fiscal document.
docPrintUrl	String	Url of the fiscal document report.
currencyCode	String	Currency code for the fiscal document. It is based on currency code defined in source location.
totalTaxAmt	BigDecimal	Total TAX amount of the fiscal document. This field is the sum of totalTaxAmt of all itens of the document. Rounding applied based on system option.
totalNonMerchTaxAmt	BigDecimal	Total TAX amount of allowances/ charges of the fiscal document. Rounding applied based on system option.
totalAmtInclTax	BigDecimal	Total value of fiscal document includind taxes. This field is the sum of totalAmtInclTax of all itens of the document. Rounding applied based on system option.
totalAmtExcTax	BigDecimal	Total value of fiscal document excluding taxes. This field has the totalAmtInclTax minus totalTaxAmt. Rounding applied based on system option.
totalAllowance	BigDecimal	Total value of all allowances in fiscal document. This field is the sum of totalAllowance of all itens of the document. Rounding applied based on system option.
totalCharge	BigDecimal	Total value of all charges in fiscal document. This field is the sum of totalCharge of all itens of the document. Rounding applied based on system option.
totalRounding	BigDecimal	The rounding amount (positive or negative) added to produce the line extension totalAmt. This field is the sum of totalRounding of all itens of the document.
terms	String	Payment terms used in this transaction
deduceTax	String	Flag that indicates if taxes were calculated by the system or informed at the fiscal document request.
createBy	String	User ID that created the record.
createDatetime	date	Record creation date.
lastUpdateBy	String	User ID that did last update of fiscal document.
lastUpdateDatetime	date	Date of last update of fiscal document.

Parameter Name	Data Type	Description
entities	List<RestFdgEtt>	Collection of entities.
details	List<RestFdgDtl>	Collection of item details

Table 5-44 RestFdgEtt

Parameter Name	Data Type	Description
entityType	String	Entity type.
entityInternalPtnrType	String	RMS partner type.
entityInternalType	String	RMS entity type.
entityCode	String	Internal RMS code of the entity. Ex. ID of store or warehouse
name	String	Entity name.
legalName	String	Entity's legal name.
carrierVehicleRn	String	Carrier vehicle's registration number.
carrierVehicleState	String	Carrier vehicle's state.
carrierDriverName	String	Carrier driver's name.
carrierDriverLn	String	Carrier driver's license number.
address	RestFdgEttAddr	Collection of address details.
identification	List<RestFdgExt>	Collection of Entity Identification Additional Properties. Fiscal Attributes will be send in this group.

Table 5-45 RestFdgEttAddr

Parameter Name	Data Type	Description
street	String	Street name of the entity's address.
additionalStreetName	String	Extension of street name of the entity's address.
complement	String	Additional information of entity's address, such as neighborhood, county, etc.
city	String	City name of entity's address.
jurisdictionCode	String	Jurisdiction code of entity's address. This can be legal code associated to city, county or any other level.
postalCode	String	Zip or postal code of entity's address.
country	String	Country name of entity's address.
telephone	String	Telephone number of the person who is the contact point.
email	String	Email address of the person who is the contact point.
state	String	State name of entity's address.

Table 5-45 (Cont.) RestFdgEttAddr

Parameter Name	Data Type	Description
contactName	String	Contact name.
additionalproperties	List<RestFdgExt>	Collection of address additional properties.

Table 5-46 RestFdgExt

Parameter Name	Data Type	Description
key	String	This column contains the key value of the entity additional properties at entity address level.
value	String	This column contains the value of the entity additional properties at entity address level.
type	String	This column contains the extension type at entity address level.
description	String	Description of additional property data.

Table 5-47 RestFdgExt (identification)

Parameter Name	Data Type	Description
key	String	This column contains the key value of the entity additional properties at entity address level.
value	String	This column contains the value of the entity additional properties at entity address level.
type	String	This column contains the extension type at entity address level.
description	String	Description of additional property data.
references	List<RestFdgRef>	Collection of references.
totaltaxes	List<RestFdgTax>	Collection of taxes.
errors	List<RestFdgError>	Collection of errors.
allowanceCharges	List<RestFdgNonMerc h>	Collection of allowance charges.
additionalproperties	List<RestFdgExt>	Collection of additional properties.
comments	List<RestFdgText>	Collection of comments.

Table 5-48 RestFdgDtl

Parameter Name	Data Type	Description
lineNo	long	Sequential number of the item in fiscal document. This number is a sequence starts with number one. Any gap is not allowed.
Item	String	Unique code ID that identifies the item.
itemDesc	String	Long description of the item.
itemShortDesc	String	Short description of the item.
totalAmt	BigDecimal	Total amount for the item inclusive of allowances and charges. Calculation formula is $((unitAmt * qty) - totalAllowance + totalCharge)$. Rounding applied based on system option.
unitAmt	BigDecimal	Item unit amount. No rounding applied.
uom	String	Unit of measure used in item in the fiscal document.
qty	BigDecimal	Quantity of item in the fiscal document.
totalTaxAmt	BigDecimal	Total amount of taxes applied to the item. This field is sum of all taxAmt at tax level for this item. Rounding applied based on system option.
totalAmtInclTax	BigDecimal	Total amount for item inclusive of taxes. Calculation formula is $totalAmt + totalTaxAmt$. Rounding applied based on system option.
totalRounding	BigDecimal	The rounding amount (positive or negative) added to produce the line extension totalAmt. Calculation formula $((unitAmt * qty) - totalAllowance + totalCharge) - totalAmt$.
totalAllowance	BigDecimal	Total amount of all allowances at item level. This field is sum of all allowances at allowanceCharges level for this item. Rounding applied based on system option.
totalCharge	BigDecimal	Total amount of all charges at item level. This field is sum of all charges at allowanceCharges level for this item. Rounding applied based on system option.
packInd	String	Pack indicator. Valid values: Y (this item is a pack) or N (this item is not a pack).
gtin	String	GTIN number (Global Trade Item Number) of item.

Table 5-48 (Cont.) RestFdgDtl

Parameter Name	Data Type	Description
gtinType	String	GTIN Type. Valid values: GTIN-8, GTIN-12, GTIN-13, GTIN-14
vpn	String	Vendor Product Number (VPN) associated with the item.
taxes	List<RestFdgTax>	Collection of item taxes.
references	List<RestFdgRef>	Collection of item references
allowanceCharges	List<RestFdgNonMerch>	Collection of item allowance charges.
packdetails	List<RestFdgDtlPack>	Collection of item pack details.
additionalproperties	List<RestFdgExt>	Collection of item additional properties.
comments	List<RestFdgText>	Collection of item comments.

Table 5-49 RestFdgTax

Parameter Name	Data Type	Description
taxType	String	Tax type.
taxCode	String	Tax identification code.
taxRate	BigDecimal	Tax rate when tax type is set as percentage.
taxAmt	BigDecimal	Total tax amount for the tax code. Rounding applied based on system option.
unitTaxAmt	BigDecimal	Tax amount per unit for the tax code.
taxBasis	BigDecimal	Tax basis amount used to calculate tax value based on taxRate field. Rounding applied based on system option.
taxCodeDesc	String	Tax code description. This field can be used to indicate extra code or detailed information about tax code.
additionalproperties	List<RestFdgExt>	Collection of taxes addition properties.
comments	List<RestFdgText>	Collection of taxes comments.

Table 5-50 RestFdgText

Parameter Name	Data Type	Description
text	String	Additional information and/or comments. It is used to provide any additional information or legal message important to the fiscal document.

Table 5-50 (Cont.) RestFdgText

Parameter Name	Data Type	Description
textGroupId	String	Text group ID. It is used to group related comments when there is more than one or when the size of a message is bigger than the size of the text field and it is necessary to generate more than one line.
textGroupSeq	String	Sequence of the text for a given group ID. It is used in case comments information has more than one line.

Table 5-51 RestFdgRef

Parameter Name	Data Type	Description
refType	String	Type of the reference information. It is used when there is any reference that it is necessary to add in the fiscal document, as shipment number, contract number, order, id, etc. Valid values are pre-defined in RMS System.
refCode	String	Reference code/value of reference type.

Table 5-52 RestFdgNonMerch

Parameter Name	Data Type	Description
chargeInd	String	Charge indicator.
nonMerchCode	String	Non-merchandise code that identifies a non-merchandise cost used as a charge or allowance.
nonMerchCodeDesc	String	Non-merchandise code description. It is used to provide more detailed about non-merchandise code, for example: allowance details, types of non-merchandise code such as Freight, Insurance, etc.
nonMerchAmt	BigDecimal	Non-merchandise amount.
nonMerchType	String	Non-merchandise type exclusive for Charges.
taxes	List<RestFdgTax>	Collection of taxes for non-merchandise.
additionalproperties	List<RestFdgExt>	Collection of additional properties for non-merchandise.
comments	List<RestFdgText>	Collection of comments for non-merchandises.

Table 5-53 packdetails

Parameter Name	Data Type	Description
item	String	Unique alphanumeric ID of a pack component that is part of item pack.
itemQty	BigDecimal	Quantity of the component item that is part of the pack.
itemDesc	String	Description of the component item component.
uom	String	Unit of measure of the component item.

Table 5-54 totaltaxes

Parameter Name	Data Type	Description
taxType	String	Not applicable for totalTaxes.
taxCode	String	Tax identification code.
taxRate	BigDecimal	Not applicable for totalTaxes.
taxAmt	BigDecimal	Sum of taxAmt for all items for the tax code. Rounding applied based on system option
unitTaxAmt	BigDecimal	Not applicable for totalTaxes.
taxBasis	BigDecimal	Sum of taxBasis for all items for the tax code. Rounding applied based on system option.
taxCodeDesc	String	Tax code description. This field can be used to indicate extra code or detailed information about tax code.

JSON Structure

```
{
  "docId": 17200,
  "requestId": 17207,
  "requestSystem": "OMNI",
  "requestAction": "A",
  "status": "S",
  "docNo": "17200",
  "docType": "WBL",
  "docTypeCode": null,
  "extReqId": "10065007",
  "transactionType": "SALES",
  "issueDate": "2021-03-31T08:19:42-05:00",
  "dueDate": null,
  "printerId": null,
  "docPrintUrl": null,
  "currencyCode": "USD",
  "totalTaxAmt": null,
  "totalNonMerchTaxAmt": null,
  "totalAmtInclTax": null,
}
```

```
"totalAmtExclTax": null,
"totalAllowance": null,
"totalCharge": null,
"totalRounding": null,
"terms": null,
"docLang": "EN",
"deduceTax": null,
"createBy": "RMS_ADMIN",
"createDatetime": "2021-06-16",
"lastUpdateBy": "RFM01",
"lastUpdateDatetime": "2021-06-16",
"entities": [
  {
    "entityType": "SHIPP",
    "entityInternalType": "ST",
    "entityInternalPtnrType": null,
    "entityCode": "990011",
    "name": "QABR-Store Test Washington",
    "legalName": "QABR-Store Test Washington",
    "carrierVehicleRn": null,
    "carrierVehicleState": null,
    "carrierDriverName": null,
    "carrierDriverLn": null,
    "address": {
      "street": "R da Juta",
      "additionalStreetName": "1512",
      "complement": null,
      "city": "Washijngton",
      "jurisdictionCode": null,
      "postalCode": "03308070",
      "country": "US",
      "telephone": "2360873",
      "email": "washington.dias@oracle.com",
      "state": "DC",
      "contactName": "Washington Dias",
      "additionalproperties": []
    },
    "identification": []
  },
  {
    "entityType": "DEST",
    "entityInternalType": "CUST",
    "entityInternalPtnrType": null,
    "entityCode": "75",
    "name": "wdias losnisk",
    "legalName": "JADLOG",
    "carrierVehicleRn": null,
    "carrierVehicleState": null,
    "carrierDriverName": null,
    "carrierDriverLn": null,
    "address": {
      "street": "end10",
      "additionalStreetName": "end20",
      "complement": "end30",
      "city": "Washington",
```

```
        "jurisdictionCode": "3100104",
        "postalCode": "08573160",
        "country": "US",
        "telephone": null,
        "email": null,
        "state": "DC",
        "contactName": "wdias losnisk",
        "additionalproperties": []
    },
    "identification": []
},
{
    "entityType": "CR",
    "entityInternalType": "PTNR",
    "entityInternalPtnrType": "FF",
    "entityCode": "11",
    "name": "Carrier",
    "legalName": "Carrier",
    "carrierVehicleRn": "CJU0939",
    "carrierVehicleState": "São Paulo",
    "carrierDriverName": "Djalma da Carreta",
    "carrierDriverLn": "77799573959957",
    "address": {
        "street": "15333 JFK Blvd",
        "additionalStreetName": null,
        "complement": null,
        "city": "Houston",
        "jurisdictionCode": null,
        "postalCode": "77032",
        "country": "US",
        "telephone": null,
        "email": "juliano.costa@oracle.com",
        "state": "TX",
        "contactName": "Juliano Costa",
        "additionalproperties": [
            {
                "key": "NIP",
                "value": "NIP09",
                "type": "VARCHAR2",
                "description": "NIP"
            },
            {
                "key": "POSTALCODE",
                "value": "POST09",
                "type": "VARCHAR2",
                "description": "Postal Code"
            }
        ]
    },
    "identification": [
        {
            "key": "TAX_ID",
            "value": "332424324",
            "type": "VARCHAR2",
            "description": "TAX ID"
        }
    ]
}
```

```
    }
  ]
}],
"details": [
  {
    "lineNo": 1,
    "item": "100900006",
    "itemDesc": "item regular scenario 3",
    "itemShortDesc": "regular item regular scenario 3",
    "totalAmt": null,
    "unitAmt": null,
    "uom": "EA",
    "qty": 5,
    "totalTaxAmt": null,
    "totalAmtInclTax": null,
    "packInd": "N",
    "gtin": null,
    "gtinType": "ITEM",
    "vpn": null,
    "totalRounding": null,
    "totalAllowance": null,
    "totalCharge": null,
    "taxes": [],
    "references": [],
    "allowanceCharges": [],
    "packdetails": [],
    "additionalproperties": [],
    "comments": []
  },
  {
    "lineNo": 2,
    "item": "100900065",
    "itemDesc": "itemm simple pack scenario 3",
    "itemShortDesc": "itemm simple pack scenario 3",
    "totalAmt": null,
    "unitAmt": null,
    "uom": "EA",
    "qty": 5,
    "totalTaxAmt": null,
    "totalAmtInclTax": null,
    "packInd": "Y",
    "gtin": null,
    "gtinType": "ITEM",
    "vpn": null,
    "totalRounding": null,
    "totalAllowance": null,
    "totalCharge": null,
    "taxes": [],
    "references": [],
    "allowanceCharges": [],
    "packdetails": [
      {
        "item": "100900057",
        "itemQty": 10,
        "itemDesc": "item componente de simple pack",
```


TABLE	SELECT	INSERT	UPDATE	DELETE
FDG_EXT	Yes	No	No	No
FDG_HDR	Yes	No	No	No
FDG_NON_MERCH	Yes	No	No	No
FDG_REF	Yes	No	No	No
FDG_TAX	Yes	No	No	No
FDG_TEXT	Yes	No	No	No

Sales Audit ReSTful Web Services

Data Privacy Access Service

This section describes the Data Privacy Access service for Sales Audit.

Business Overview

This query service provides access to data stored in Sales Audit that contain personally identifiable information.

Service Type

GET

ReSTURL

```
https://<host:port>/RetailAppsDataPrivServicesRESTApp/rest/privatedata/  
getPersonallInfo?customer_id={entityName}::{entityType}::{entityId}::{fullName}::  
{phone}::{email}
```

Accept

- application/json
- application/xml

Query Parameters

- customer_id (required): The customer ID string containing the parameters to be used in looking up data. The format of this string is as follows:
 - {entity name}::{entity type}::{entity id}::{full name}::{phone}::{email}

Path Parameters

Parameter	Description
Entity Name	The query group type for which data is to be retrieved. The available group types for Sales Audit are: <ul style="list-style-type: none"> EMPLOYEE CUSTOMER
Entity Type	Used if the entity name is CUSTOMER. The value here should indicate the type of customer being queried. Valid values for this input can be found on the Codes table where code type is 'CIDT'.
Entity ID	The ID of the entity being queried. For example, the employee ID.
Full Name	The full name to be searched for the selected entity.
Phone	The phone number to be searched for the selected entity.
Email	The email to be searched for the selected entity.

Default Response

The response will return all instances of the data being searched that occur in the requested entity. For example, if the entity requested was EMPLOYEE, all instances where the employee, name, phone, and email match the data sent will be returned. If any of these parameters are not sent (e.g. employee), then it will not be used as part of the search. The following data is included in the response:

Parameter	Description
Entity Name	Valid values are <ul style="list-style-type: none"> EMPLOYEE CUSTOMER
Entity Type	If the entity name is CUSTOMER, the value here indicates the type of customer being queried. Valid values for this input can be found on the Codes table where code type is 'CIDT'. For other entity types, this will be null.
Entity ID	The ID of the entity where the data was found.
Full Name	The name associated with the entity.
Phone	The phone number associated with the entity.
Fax	The fax number associated with the entity.
Telex	The telex number associated with the entity.
Pager	The pager number associated with the entity.
Email	The email address associated with the entity.

Sample Response

```
{
  "Personal Information": {
    "list": [],
    "Get Personal Information": {
      "list": [
        [
          {
            "ENTITY_NAME": "EMPLOYEE",
            "ENTITY_TYPE": "null",

```

```
        "ENTITY_ID": "1414",  
        "FULL_NAME": "Harry Adams",  
        "PHONE": "2349989",  
        "FAX": "null",  
        "TELEX": "null",  
        "PAGER": "null",  
        "EMAIL": "hadams@email.com"  
    }  
  ]  
}
```

Response Codes and Error Messages

- 200 - Success
- 400 - Bad Request - for the following situations:
 - Customer ID does not match the required format
 - Invalid input type
 - Missing customer ID
 - Invalid jsonFormat
- 500 - Internal Server Errors - for all other types of errors (for example, configuration errors, SQL errors, and so on)

Success Payloads

- When Accept=application/json, this API will return data in JSON format
- When Accept=application/xml, this API will return data formatted as an HTML page

Data Privacy Forget Service

This section describes the Data Privacy Forget service for Sales Audit.

Business Overview

This service supports updating personal information stored in Sales Audit. When the service is invoked with mask strings as inputs, it overwrites the fields with mask strings, which effectively removes the personal information from the system.

Service Type

DELETE

ReSTURL

https://<host:port>/RetailAppsDataPrivServicesRESTApp/rest/privatedata/updatePersonalInfo?customer_id={entityName}::{entityType}::{entityId}::{fullName}::

```
{phone}::{fax}::{telex}::{pager}::{email}::{addr1}::{addr2}::{addr3}::{county}::{city}::{state}::{countryId}::{postalCode}
```

Accept

- application/json
- application/xml

Query Parameters

- customer_id (required): The customer ID string containing the parameters to be used in updating data. The format of this string is as follows:
 - {entityName}::{entityType}::{entityId}::{fullName}::{phone}::{fax}::{telex}::{pager}::{email}::{addr1}::{addr2}::{addr3}::{county}::{city}::{state}::{countryId}::{postalCode}

Path Parameters

Parameter	Description
Entity Name (required)	The group type for which data is to be updated. The available group types for Sales Audit are: <ul style="list-style-type: none"> • EMPLOYEE • CUSTOMER
Entity Type	Required if the entity name is CUSTOMER. The value here should indicate the type of customer. Valid values for this input can be found on the Codes table where code type is 'CIDT'.
Entity ID (required)	The ID of the entity to be updated. For example, the employee ID.
Full Name	The value to update the full name with. If a null value is passed to this parameter that is a required field in the table, the field will be updated to 'XXXXX'.
Phone	The value to update the phone number with. If a null value is passed to this parameter that is a required field in the table, the field will be updated to 'XXXXX'.
Fax	The value to update the fax number with.
Telex	The value to update the telex number with.
Pager	The value to update the pager number with.
Email	The value to update the email address with.
Addr1	The value to update the address 1 with.
Addr2	The value to update the address 2 with.
Addr3	The value to update the address 3 with.
County	The value to update the county with.
City	The value to update the city with.
State	The value to update the state with.
Country	The value to update the country with.
Postal Code	The value to update the postal code with.

Default Response

This service only returns a response code to signify if the request is successful or not. If no record is updated, the service returns an error.

Response Codes and Error Messages

- 200 - Success
- 400 - Bad Request - for the following situations:
 - Customer ID does not match the required format
 - Invalid input type
 - Missing customer ID
 - Invalid jsonFormat
- 500 - Internal Server Errors - for all other types of errors (for example, configuration errors, SQL errors, and so on).

Success Payloads

N/A

Get Rejected Transactions Since Previous Query

Functional Area

Financial

Business Overview

The purpose of this service is to return transactions that were rejected during processing to the calling process or application, for use in troubleshooting and error resolution.

Sales Service

This services fetches and returns transactions from the staging table SVC_RTLOG_DATA_REJECT that meet the input criteria. Transactions failing validations in the `sacreatertlog.ksh` process are inserted into this staging table during processing

Service Type

Put

ReST URL

Resa/getRejectedTransactions

Input Parameters

Table 5-55 SvcprovRejectedRecRDO

Parameter Name	Data Type
pageSize	Number
pageNumber	Number
readUnreadErrorInd	String
startDate	String (Format :YYYYMMDD)
endDate	String (Format :YYYYMMDD)
svcprovRejStoreRDOs	List<SvcprovRejStoreRecRDO>
svcprovRejErrorcodeRDOs	List<SvcprovRejErrorcodeRecRDO>
svcprovRejTransactionidRDOs	List<SvcprovRejTransactionidRecRDO>

Table 5-56 SvcprovRejStoreRecRDO

Parameter Name	Data Type
store	String

Table 5-57 SvcprovRejErrorcodeRecRDO

Parameter Name	Data Type
errorCode	String

Table 5-58 SvcprovRejTransactionidRecRDO

Parameter Name	Data Type
transactionId	String

JSON Input Structure

```
[
  {
    "pageSize": null,
    "pageNumber": null,
    "readUnreadErrorInd": null,
    "startDate": null,
    "endDate": null,
    "svcprovRejStoreRDOs": [
      {
        "store": null
      }
    ],
    "svcprovRejErrorcodeRDOs": [
      {
```

```

        "errorCode": null
      }
    ],
    "svcprovRejTransactionidRDOs": [
      {
        "transactionId": null
      }
    ]
  }
]

```

Output

The output will contain the status of the request including validation errors, if any.

Table 5-59 Output Record

Parameter Name	Data Type
totalRecordCount	Number
results	Table of SVCPROV_REJ_THEAD_REC

Table 5-60 SVCPROV_REJ_THEAD_REC

Parameter Name	Data Type
Store	Number
businessDate	Date
errorCode	String
errorDesc	String
recType	String
register	String
transactionDate	Date
transactionNo	Number
cashier	String
salesperson	String
transactionType	String
subTransactionType	String
origSalesNo	Number
origRegNo	String
reasonCode	String
vendorNo	String
vendorInvcNo	String
paymentRefNo	String
prfDlvyNo	String
refNo1	String
refNo2	String

Table 5-60 (Cont.) SVCPROV_REJ_THEAD_REC

Parameter Name	Data Type
refNo3	String
refNo4	String
value	BigDecimal
bannerId	String
roundedAmt	BigDecimal
roundedOffAmt	BigDecimal
creditPromold	String
refNo25	String
refNo26	String
refNo27	String
tranProcessSys	String
refNo28	String
refNo29	String
refNo30	String
refNo31	String
salesThattTbl	List<SvcprovSalesAttrRecRDO>
salesTcustTbl	List<SvcprovSalesTcustRecRDO>
salesCattTbl	List< SvcprovSalesAttrRecRDO>
salesTitemTbl	List< SvcprovSalesTitemRecRDO>
salesTtaxTbl	List< SvcprovSalesTtaxRecRDO>
salesTpymtTbl	List< SvcprovSalesTpymtRecRDO>
salesTtendTbl	List<SvcprovSalesTtendRecRDO>
salesTtailTbl	List<SvcprovSalesCounterRecRDO>

Table 5-61 SvcprovSalesAttrRecRDO

Parameter Name	Data Type
recType	String
attributeType	String
attributeValue	String

Table 5-62 SvcprovSalesTcustRecRDO

Parameter Name	Data Type
recType	String
customerId	String
customerType	String
customerName	String

Table 5-62 (Cont.) SvcprovSalesTcustRecRDO

Parameter Name	Data Type
address1	String
address2	String
city	String
state	String
zipCode	String
country	String
homePhone	String
workPhone	String
email	String
birthdate	Date

Table 5-63 SvcprovSalesTitemRecRDO

Parameter Name	Data Type
recType	String
itemStatus	String
itemType	String
itemNoType	String
formatId	String
item	String
refItem	String
nonMerchItem	String
voucherNo	String
quantity	String
sellingUom	String
unitRetail	BigDecimal
overrideReason	String
originalUnitRetail	BigDecimal
taxableIndicator	String
pump	String
refNo5	String
refNo6	String
refNo7	String
refNo8	String
itemSwipedInd	String
returnReasonCode	String
salesperson	String
expirationDate	Date

Table 5-63 (Cont.) SvcprovSalesTitemRecRDO

Parameter Name	Data Type
dropShipInd	String
uomQty	BigDecimal
catchWeightInd	String
sellingItem	String
custOrdLineNo	Number
mediaId	Number
totalIgtaxAmount	BigDecimal
uniqueId	String
custOrdNo	String
custOrdDate	Date
fulfillmentOrdNo	String
noInventoryReturn	String
salesType	String
returnWh	Number
returnDeposition	String
originalStore	Number
originalTransactionNo	Number
fulfillmentLocType	String
fulfillmentLoc	Number
postingStore	String
salesIattTbl	List<SvcprovSalesAttrRecRDO>
salesIdiscTbl	List<SvcprovSalesIdiscRecRDO>
salesIgtaxTbl	List<SvcprovSalesIgtaxRecRDO>

Table 5-64 SvcprovSalesIdiscRecRDO

Parameter Name	Data Type
recType	String
merchPromoNo	String
discountRefNo	String
discountType	String
couponNo	String
couponRefNo	String
quantity	BigDecimal
unitDiscountAmount	BigDecimal
refNo13	String
refNo14	String
refNo15	String

Table 5-64 (Cont.) SvcprovSalesIdiscRecRDO

Parameter Name	Data Type
refNo16	String
uomQty	BigDecimal
catchWeightInd	String
promoComponent	Number
salesTdattTbl	List<SvcprovSalesAttrRecRDO>

Table 5-65 SvcprovSalesIgtaxRecRDO

Parameter Name	Data Type
recType	String
taxAuthority	String
igtaxCode	String
igtaxRate	BigDecimal
igtaxAmount	BigDecimal
refNo21	String
refNo22	String
refNo23	String
refNo24	String
salesIxattTbl	List<SvcprovSalesAttrRecRDO>

Table 5-66 SvcprovSalesTtaxRecRDO

Parameter Name	Data Type
recType	String
taxCode	String
taxAmount	BigDecimal
refNo17	String
refNo18	String
refNo19	String
refNo20	String
transactionTaxAttributeTbl	List<SvcprovSalesAttrRecRDO>

Table 5-67 SvcprovSalesTpymtRecRDO

Parameter Name	Data Type
recType	String
paymentAmount	BigDecimal

Table 5-68 SvcprovSalesTtendRecRDO

Parameter Name	Data Type
recType	String
tendTypeGroup	String
tendTypeId	Number
tenderAmount	BigDecimal
ccNo	String
ccAuthNo	String
ccAuthSource	String
ccCardholderVerification	String
ccExpirationDate	Date
ccEntryMode	String
ccTerminalId	String
ccSpecialCondition	String
ccToken	String
voucherNo	String
couponNbr	String
couponRefNo	String
chequeAccNo	String
chequeNo	Number
identificationMethod	String
identificationId	String
originalCurrency	String
originalCurrencyAmount	BigDecimal
refNbr9	String
refNbr10	String
refNbr11	String
refNbr12	String
transactionTenderAttributeTbl	List<SvcprovSalesAttrRecRDO>

Table 5-69 SvcprovSalesCounterRecRDO

Parameter Name	Data Type
recType	String
recCounter	Number

JSON Structure

```
{
  "totalRecordCount": null,
  "results": [
    {
```

```
"store": null,
"businessDateDisplay": null,
"businessDate": null,
"errorCode": null,
"errorDesc": null,
"recType": "THEAD",
"register": null,
"transactionDateDisplay": null,
"transactionDate": null,
"transactionNo": null,
"cashier": null,
"salesperson": null,
"transactionType": null,
"subTransactionType": null,
"origSalesNo": null,
"origRegNo": null,
"reasonCode": null,
"vendorNo": null,
"vendorInvcNo": null,
"paymentRefNo": null,
"prfDlvvyNo": null,
"refNo1": null,
"refNo2": null,
"refNo3": null,
"refNo4": null,
"value": null,
"bannerId": null,
"roundedAmt": null,
"roundedOffAmt": null,
"creditPromoId": null,
"refNo25": null,
"refNo26": null,
"refNo27": null,
"tranProcessSys": null,
"refNo28": null,
"refNo29": null,
"refNo30": null,
"refNo31": null,
"salesThattTbl": [
  {
    "recType": "THATT",
    "attributeType": null,
    "attributeValue": null
  }
],
"salesTcustTbl": [
  {
    "recType": "TCUST",
    "customerId": null,
    "customerType": null,
    "customerName": null,
    "address1": null,
    "address2": null,
    "city": null,
    "state": null,
```

```
    "zipCode": null,  
    "country": null,  
    "homePhone": null,  
    "workPhone": null,  
    "email": null,  
    "birthdateDisplay": null,  
    "birthdate": null  
  },  
  {  
    "recType": "TCUST",  
    "customerId": null,  
    "customerType": null,  
    "customerName": null,  
    "address1": null,  
    "address2": null,  
    "city": null,  
    "state": null,  
    "zipCode": null,  
    "country": null,  
    "homePhone": null,  
    "workPhone": null,  
    "email": null,  
    "birthdateDisplay": null,  
    "birthdate": null  
  }  
],  
"salesCattTbl": [  
  {  
    "recType": "CATT",  
    "attributeType": null,  
    "attributeValue": null  
  }  
],  
"salesTitemTbl": [  
  {  
    "recType": "TITEM",  
    "itemStatus": null,  
    "itemType": null,  
    "itemNoType": null,  
    "formatId": null,  
    "item": null,  
    "refItem": null,  
    "nonMerchItem": null,  
    "voucherNo": null,  
    "quantity": null,  
    "sellingUom": null,  
    "unitRetail": null,  
    "overrideReason": null,  
    "originalUnitRetail": null,  
    "taxableIndicator": "Y",  
    "pump": null,  
    "refNo5": null,  
    "refNo6": null,  
  }  
]
```

```
"refNo7": null,  
"refNo8": null,  
"itemSwipedInd": null,  
"returnReasonCode": null,  
"salesperson": null,  
"expirationDate": null,  
"dropShipInd": null,  
"uomQty": null,  
"catchWeightInd": null,  
"sellingItem": null,  
"custOrdLineNo": null,  
"mediaId": null,  
"totalIgtaxAmount": null,  
"uniqueId": null,  
"custOrdNo": null,  
"custOrdDateDisplay": null,  
"custOrdDate": null,  
"fulfillmentOrdNo": null,  
"noInventoryReturn": null,  
"salesType": null,  
"returnWh": null,  
"returnDeposition": null,  
"originalStore": null,  
"originalTransactionNo": null,  
"fulfillmentLocType": null,  
"fulfillmentLoc": null,  
"postingStore": null,  
"salesItattTbl": [  
  {  
    "recType": "ITATT",  
    "attributeType": null,  
    "attributeValue": null  
  },  
  {  
    "recType": "ITATT",  
    "attributeType": null,  
    "attributeValue": null  
  }  
],  
"salesIdiscTbl": [  
  {  
    "recType": "IDISC",  
    "merchPromoNo": null,  
    "discountRefNo": null,  
    "discountType": null,  
    "couponNo": null,  
    "couponRefNo": null,  
    "quantity": null,  
    "unitDiscountAmount": null,  
    "refNo13": null,  
    "refNo14": null,  
    "refNo15": null,  
    "refNo16": null,  
  }  
]
```

```
        "uomQty": null,
        "catchWeightInd": null,
        "promoComponent": null,
        "salesTdattTbl":
    }
],
"salesIgtaxTbl": [
    {
        "recType": "IGTAX",
        "taxAuthority": null,
        "igtaxCode": null,
        "igtaxRate": null,
        "igtaxAmount": null,
        "refNo21": null,
        "refNo22": null,
        "refNo23": null,
        "refNo24": null,
        "salesIxattTbl": [
            {
                "recType": "IXATT",
                "attributeType": null,
                "attributeValue": null
            }
        ]
    }
]
},
"salesTtaxTbl": [
    {
        "recType": "TTAX",
        "taxCode": null,
        "taxAmount": null,
        "refNo17": null,
        "refNo18": null,
        "refNo19": null,
        "refNo20": null,
        "salesTxattTbl": [
            {
                "recType": "TXATT",
                "attributeType": null,
                "attributeValue": null
            }
        ]
    }
],
"salesTpymtTbl": [
    {
        "recType": "TPYMT",
        "paymentAmount": null
    }
],
"salesTtendTbl": [
    {
```

```

    "recType": "TTEND",
    "tendTypeGroup": null,
    "tendTypeId": null,
    "tenderAmount": null,
    "ccNo": null,
    "ccAuthNo": null,
    "ccAuthSource": null,
    "ccCardholderVerification": null,
    "ccExpirationDateDisplay": null,
    "ccExpirationDate": null,
    "ccEntryMode": null,
    "ccTerminalId": null,
    "ccSpecialCondition": null,
    "ccToken": null,
    "voucherNo": null,
    "couponNbr": null,
    "couponRefNo": null,
    "chequeAccNo": null,
    "chequeNo": null,
    "identificationMethod": null,
    "identificationId": null,
    "originalCurrency": null,
    "originalCurrencyAmount": null,
    "refNbr9": null,
    "refNbr10": null,
    "refNbr11": null,
    "refNbr12": null,
    "salesTtattTbl": [
      {
        "recType": "TTATT",
        "attributeType": null,
        "attributeValue": null
      }
    ]
  },
  "salesTtailTbl": []
}

```

Table Impact

TABLE	SELECT	INSERT	UPDATE	DELETE
SVC_RTLOG_DATA_REJECT	Yes	No	Yes	No

Get Store Aggregations

Business Overview

Retrieves aggregated store day information for all dates or store days older than vdate -5.

Service Type

Get

ReST URL

```
/getStoreAggregations?  
allOlderInd={allOlderInd}&stores={stores}&sortAttrib={sortAttrib}&sortDirection={sortDirection  
&pageSize={pageSize}&pageNumber={pageNumber}
```

Input Parameters

Parameter Name	Required	Description	Valid values
AllOlderInd	Yes	Search string for locations ID or Name	ALL, OLDER
Stores	No	Comma-separated values for stores	NA
SortAttrib	No	Sort Attribute	STORENAME, AUDITOR, OSVALUE, ERRORCNT, DATASTATUS, OPENDAYS, OSDAYS and OSSUMS
SortDirection	No	Sort Direction	ASC, DESC
PageSize	No	Maximum number of locations to retrieve per page	NA
PageNumber	No	Result page to retrieve	NA

Output

Store

Store Name

Chain

Chain Name

Auditors

Open Days

Over Days

Short Days

Over Amount

Short Amount

Currency Code

Error Count

Table Impact

TABLE	SELECT	INSERT	UPDATE	DELETE
SA_ERROR	Yes	No	No	No
SA_HQ_VALUE	Yes	No	No	No
SA_POS_VALUE	Yes	No	No	No
SA_STORE_DATA	Yes	No	No	No
SA_STORE_DAY	Yes	No	No	No
SA_SYS_VALUE	Yes	No	No	No
V_CHAIN	Yes	No	No	No
V_STORE	Yes	No	No	No

Get Store Day Date Indicator

Business Overview

This web service allows the user to find which store days have records that needs attention.

Service Type

Get

ReST URL

/getStoreDateInd?store={store}

Input Parameters

Parameter Name	Required	Description	Valid values
store	Yes	Store ID	NA

Output

Record Type --DATE, OLDER, ALL

- **For record type DATE:** five records of type date are displayed for today minus 1 through today minus 5
- **One record type OLDER:** is for store days older than today minus 5
- **One record type ALL:** for all store days

Record Date --Date of date type rows

Store Has Value indicator

Table Impact

TABLE	SELECT	INSERT	UPDATE	DELETE
SA_STORE_DAY	Yes	No	No	No
V_STORE	Yes	No	No	No

Get Store Days

Business Overview

The service displays a list of open stores to which the user is assigned, for a single day, for 'OLDER' days, or for 'ALL' days.

Service Type

Get

ReST URL

```
/getStoreDays?
store={store}&recordType={recordType}&recordDate={recordDate}&sortAttrib={sortAttrib}&sortDirection={sortDirection}&pageSize={pageSize}&pageNumber={pageNumber}
```

Input Parameters

Parameter Name	Required	Description	Valid values
RecordType	Yes	Record Type	ALL, OLDER, DATE
RecordDate	No	Record Date, required when recordType is DATE	NA
Store	No	Store ID	NA
SortAttrib	No	Sort Attribute	STORENAME, AUDITOR, OSVALUE, ERRORCNT, DATASTATUS, OPENDAYS, OSDAYS and OSSUMS
SortDirection	No	Sort Direction	ASC, DESC
PageSize	No	Maximum number of locations to retrieve per page	NA
PageNumber	No	Result page to retrieve	NA

Output

Store

Store Day Seq No

Auditors

Business Date
 Store Name
 Chain
 Chain Name
 Data Status
 Data Status Description
 Audit Status
 Audit Status Description
 Audit Changed Datetime
 Fuel Status
 Fuel Status Description
 Over Short Amount
 Currency Code
 Error Count
 Transaction Count
 Loaded File Count
 Expected File Count

Table Impact

TABLE	SELECT	INSERT	UPDATE	DELETE
LOC_TRAITS_MATRIX	Yes	No	No	No
SA_ERROR	Yes	No	No	No
SA_HQ_VALUE	Yes	No	No	No
SA_POS_VALUE	Yes	No	No	No
SA_STORE_DATA	Yes	No	No	No
SA_STORE_DAY	Yes	No	No	No
SA_SYS_VALUE	Yes	No	No	No
SA_SYSTEM_OPTIONS	Yes	No	No	No
SA_TOTAL	Yes	No	No	No
SA_TRAN_HEAD	Yes	No	No	No
SA_USER_LOC_TRAITS	Yes	No	No	No
V_CHAIN	Yes	No	No	No
V_CODE_DETAIL	Yes	No	No	No
V_STORE	Yes	No	No	No

Get Store Errors

Business Overview

Retrieves summary of store day errors.

Service Type

Get

ReST URL

/getStoreErrors?store={store}&recordType={recordType}&recordDate={recordDate}

Input Parameters

Parameter Name	Required	Description	Valid values
RecordType	Yes	Record Type	ALL, OLDER, DATE
RecordDate	No	Record Date, required when recordType is DATE	NA
Store	No	Store ID	NA

Output

Store

Error Code

Error Description

Error Percentage

Table Impact

TABLE	SELECT	INSERT	UPDATE	DELETE
SA_ERROR	Yes	No	No	No
SA_STORE_DAY	Yes	No	No	No
V_SA_ERROR	Yes	No	No	No
V_STORE	Yes	No	No	No

Load Transactions

Functional Area

Financial

Business Overview

This service accepts transactions from a Point of Sale (POS) or Order Management Solution (OMS) into Sales Audit. This service provides an alternative to transactions processed from an RTLOG file through the Sales Audit sales process documented under the Scheduled Integration section.

This service accepts transactions from a POS or OMS system in a JSON format and loads them into the staging table `SVC_RTLOG_DATA_LOAD` for further processing. This service performs minimal data validation, as the majority of the sales data validation remains in the Sales Audit import process. The transactions staged through this service are extracted into the RTLOG format by the `sacreatertlog.ksh` process and processed by the `saimptlog` process.

In addition, this service supports a near-real-time update of inventory impacted by sales transactions by recording sales units in the `SA_INPROGRESS_SALES` table. The units record in this table for an item/location can be netted out of stock on hand for a more accurate and real-time view of available inventory. When these sales are processed completely through auditing, the in-progress units will be negated.

Service Type

Post

ReST URL

Resa/salesService

Input Parameters

Table 5-70 SvcprovSalesServiceRecRDO

Parameter Name	Data Type
recType	String
fileCreateDate	String (Format :YYYYMMDDHHMMSS)
businessDate	String(Format: YYYYMMDD)
store	Number
referenceNbr	String
rtlogOrgSys	String
transactionHeadTbl	List<SvcprovSalesTheadRecRDO>
salesFtailTbl	List<SvcprovSalesCounterRecRDO>

Table 5-71 SvcprovSalesTheadRecRDO

Parameter Name	Data Type
recType	String
register	String
transactionDate	String (Format :YYYYMMDDHHMMSS)

Table 5-71 (Cont.) SvcprovSalesTheadRecRDO

Parameter Name	Data Type
transactionNo	Number
cashier	String
salesperson	String
transactionType	String
subTransactionType	String
origSalesNo	Number
origRegNo	String
reasonCode	String
vendorNo	String
vendorInvNo	String
paymentRefNo	String
prfDlvyNo	String
refNo1	String
refNo2	String
refNo3	String
refNo4	String
value	BigDecimal
bannerId	String
roundedAmt	BigDecimal
roundedOffAmt	BigDecimal
creditPromold	String
refNo25	String
refNo26	String
refNo27	String
tranProcessSys	String
refNo28	String
refNo29	String
refNo30	String
refNo31	String
transactionHeadAttributeTbl	List<SvcprovSalesAttrRecRDO>
transactionCustomerTbl	List< SvcprovSalesTcustRecRDO>
customerAttributeTbl	List< SvcprovSalesAttrRecRDO>
transactionItemTbl	List< SvcprovSalesTitemRecRDO>
transactionTaxTbl	List< SvcprovSalesTtaxRecRDO>
transactionPaymentTbl	List< SvcprovSalesTpymtRecRDO>
transactionTenderTbl	List<SvcprovSalesTtendRecRDO>
transactionTrailerTbl	List<SvcprovSalesCounterRecRDO>

Table 5-72 SvcprovSalesAttrRecRDO

Parameter Name	Data Type
recType	String
attributeType	String
attributeValue	String

Table 5-73 SvcprovSalesTcustRecRDO

Parameter Name	Data Type
recType	String
customerId	String
customerType	String
customerName	String
address1	String
address2	String
city	String
state	String
zipCode	String
country	String
homePhone	String
workPhone	String
email	String
birthdate	String (Format: YYYYMMDD)

Table 5-74 SvcprovSalesTitemRecRDO

Parameter Name	Data Type
recType	String
itemStatus	String
itemType	String
itemNoType	String
formatId	String
item	String
refItem	String
nonMerchItem	String
voucherNo	String
quantity	String
sellingUom	String
unitRetail	BigDecimal
overrideReason	String

Table 5-74 (Cont.) SvcprovSalesTitemRecRDO

Parameter Name	Data Type
originalUnitRetail	BigDecimal
taxableIndicator	String
pump	String
refNo5	String
refNo6	String
refNo7	String
refNo8	String
itemSwipedInd	String
returnReasonCode	String
salesperson	String
expirationDate	String (Format: YYYYMMDD)
dropShipInd	String
uomQty	BigDecimal
catchWeightInd	String
sellingItem	String
custOrdLineNo	Number
mediaId	Number
totalIgtaxAmount	BigDecimal
uniqueId	String
custOrdNo	String
custOrdDate	String(Format: YYYYMMDDHHMMSS)
fulfillmentOrdNo	String
noInventoryReturn	String
salesType	String
returnWh	Number
returnDeposition	String
originalStore	Number
originalTransactionNo	Number
fulfillmentLocType	String
fulfillmentLoc	Number
postingStore	String
transactionItemAttributeTbl	List<SvcprovSalesAttrRecRDO>
transactionItemDiscountTbl	List<SvcprovSalesIdiscRecRDO>
transactionItemTaxTbl	List<SvcprovSalesIgtaxRecRDO>

Table 5-75 SvcprovSalesIdiscRecRDO

Parameter Name	Data Type
recType	String
merchPromoNo	String
discountRefNo	String
discountType	String
couponNo	String
couponRefNo	String
quantity	BigDecimal
unitDiscountAmount	BigDecimal
refNo13	String
refNo14	String
refNo15	String
refNo16	String
uomQty	BigDecimal
catchWeightInd	String
promoComponent	Number
transactionItemDiscountAttributeTbl	List<SvcprovSalesAttrRecRDO>

Table 5-76 SvcprovSalesIgtaxRecRDO

Parameter Name	Data Type
recType	String
taxAuthority	String
igtaxCode	String
igtaxRate	BigDecimal
igtaxAmount	BigDecimal
refNo21	String
refNo22	String
refNo23	String
refNo24	String
transactionItemTaxAttributeTbl	List<SvcprovSalesAttrRecRDO>

Table 5-77 SvcprovSalesTtaxRecRDO

Parameter Name	Data Type
recType	String
taxCode	String
taxAmount	BigDecimal
refNo17	String

Table 5-77 (Cont.) SvcprovSalesTtaxRecRDO

Parameter Name	Data Type
refNo18	String
refNo19	String
refNo20	String
transactionTaxAttributeTbl	List<SvcprovSalesAttrRecRDO>

Table 5-78 SvcprovSalesTpymtRecRDO

Parameter Name	Data Type
recType	String
paymentAmount	BigDecimal

Table 5-79 SvcprovSalesTtendRecRDO

Parameter Name	Data Type
recType	String
tendTypeGroup	String
tendTypeId	Number
tenderAmount	BigDecimal
ccNo	String
ccAuthNo	String
ccAuthSource	String
ccCardholderVerification	String
ccExpirationDate	String(Format: YYYYMMDD)
ccEntryMode	String
ccTerminalId	String
ccSpecialCondition	String
ccToken	String
voucherNo	String
couponNbr	String
couponRefNo	String
chequeAccNo	String
chequeNo	Number
identificationMethod	String
identificationId	String
originalCurrency	String
originalCurrencyAmount	BigDecimal
refNbr9	String
refNbr10	String

Table 5-79 (Cont.) SvcprovSalesTtendRecRDO

Parameter Name	Data Type
refNbr11	String
refNbr12	String
transactionTenderAttributeTbl	List<SvcprovSalesAttrRecRDO>

Table 5-80 SvcprovSalesCounterRecRDO

Parameter Name	Data Type
recType	String
recCounter	Number

JSON Structure

```
[
  {
    "recType": null,
    "fileCreateDate": null,
    "businessDate": null,
    "store": null,
    "referenceNbr": null,
    "rtlogOrgSys": null,
    "transactionHeadTbl": [
      {
        "recType": null,
        "register": null,
        "transactionDate": null,
        "transactionNo": null,
        "cashier": null,
        "salesperson": null,
        "transactionType": null,
        "subTransactionType": null,
        "origSalesNo": null,
        "origRegNo": null,
        "reasonCode": null,
        "vendorNo": null,
        "vendorInvcNo": null,
        "paymentRefNo": null,
        "prfDlvyNo": null,
        "refNo1": null,
        "refNo2": null,
        "refNo3": null,
        "refNo4": null,
        "value": null,
        "bannerId": null,
        "roundedAmt": null,
        "roundedOffAmt": null,
        "creditPromoId": null,
        "refNo25": null,
      }
    ]
  }
]
```

```
"refNo26": null,  
"refNo27": null,  
"tranProcessSys": null,  
"refNo28": null,  
"refNo29": null,  
"refNo30": null,  
"refNo31": null,  
"transactionHeadAttributeTbl": [  
  {  
    "recType": null,  
    "attributeType": null,  
    "attributeValue": null  
  }  
],  
"transactionCustomerTbl": [  
  {  
    "recType": null,  
    "customerId": null,  
    "customerType": null,  
    "customerName": null,  
    "address1": null,  
    "address2": null,  
    "city": null,  
    "state": null,  
    "zipCode": null,  
    "country": null,  
    "homePhone": null,  
    "workPhone": null,  
    "email": null,  
    "birthdate": null  
  }  
],  
"customerAttributeTbl": [  
  {  
    "recType": null,  
    "attributeType": null,  
    "attributeValue": null  
  }  
],  
"transactionItemTbl": [  
  {  
    "recType": null,  
    "itemStatus": null,  
    "itemType": null,  
    "itemNoType": null,  
    "formatId": null,  
    "item": null,  
    "refItem": null,  
    "nonMerchItem": null,  
    "voucherNo": null,  
    "quantity": null,  
    "sellingUom": null,  
    "unitRetail": null,  
    "overrideReason": null,  
    "originalUnitRetail": null,  
  }  
]
```

```
"taxableIndicator": null,
"pump": null,
"refNo5": null,
"refNo6": null,
"refNo7": null,
"refNo8": null,
"itemSwipedInd": null,
"returnReasonCode": null,
"salesperson": null,
"expirationDate": null,
"dropShipInd": null,
"uomQty": null,
"catchWeightInd": null,
"sellingItem": null,
"custOrdLineNo": null,
"mediaId": null,
"totalIgtaxAmount": null,
"uniqueId": null,
"custOrdNo": null,
"custOrdDate": null,
"fulfillmentOrdNo": null,
"noInventoryReturn": null,
"salesType": null,
"returnWh": null,
"returnDeposition": null,
"originalStore": null,
"originalTransactionNo": null,
"fulfillmentLocType": null,
"fulfillmentLoc": null,
"postingStore": null,
"transactionItemAttributeTbl": [
  {
    "recType": null,
    "attributeType": null,
    "attributeValue": null
  }
],
"transactionItemDiscountTbl": [
  {
    "recType": null,
    "merchPromoNo": null,
    "discountRefNo": null,
    "discountType": null,
    "couponNo": null,
    "couponRefNo": null,
    "quantity": null,
    "unitDiscountAmount": null,
    "refNo13": null,
    "refNo14": null,
    "refNo15": null,
    "refNo16": null,
    "uomQty": null,
    "catchWeightInd": null,
    "promoComponent": null,
    "transactionItemDiscountAttributeTbl": [
```

```
        {
          "recType": null,
          "attributeType": null,
          "attributeValue": null
        }
      ]
    }
  ],
  "transactionItemTaxTbl": [
    {
      "recType": null,
      "taxAuthority": null,
      "igtaxCode": null,
      "igtaxRate": null,
      "igtaxAmount": null,
      "refNo21": null,
      "refNo22": null,
      "refNo23": null,
      "refNo24": null,
      "transactionItemTaxAttributeTbl": [
        {
          "recType": null,
          "attributeType": null,
          "attributeValue": null
        }
      ]
    }
  ]
},
"transactionTaxTbl": [
  {
    "recType": null,
    "taxCode": null,
    "taxAmount": null,
    "refNo17": null,
    "refNo18": null,
    "refNo19": null,
    "refNo2null": null,
    "transactionTaxAttributeTbl": [
      {
        "recType": null,
        "attributeType": null,
        "attributeValue": null
      }
    ]
  }
],
"transactionPaymentTbl": [
  {
    "recType": null,
    "paymentAmount": null
  }
],
"transactionTenderTbl": [
```

```
{
  "recType": null,
  "tendTypeGroup": null,
  "tendTypeId": null,
  "tenderAmount": null,
  "ccNo": null,
  "ccAuthNo": null,
  "ccAuthSource": null,
  "ccCardholderVerification": null,
  "ccExpirationDate": null,
  "ccEntryMode": null,
  "ccTerminalId": null,
  "ccSpecialCondition": null,
  "ccToken": null,
  "voucherNo": null,
  "couponNbr": null,
  "couponRefNo": null,
  "chequeAccNo": null,
  "chequeNo": null,
  "identificationMethod": null,
  "identificationId": null,
  "originalCurrency": null,
  "originalCurrencyAmount": null,
  "refNbr9": null,
  "refNbr10": null,
  "refNbr11": null,
  "refNbr12": null,
  "transactionTenderAttributeTbl": [
    {
      "recType": null,
      "attributeType": null,
      "attributeValue": null
    }
  ]
},
"transactionTrailerTbl": [
  {
    "recType": null,
    "recCounter": null
  }
]
},
"salesFtailTbl": [
  {
    "recType": null,
    "recCounter": null
  }
]
}
```


Output

Table 5-81 SVCPROV_SALES_STATUS_REC

Parameter Name	Data Type
statusMsg	String
salesErrTbl	List< SVCPROV_SALES_FAIL_REC >

Table 5-82 SVCPROV_SALES_FAIL_REC

Parameter Name	Data Type
store	BigDecimal
transactionNo	BigDecimal
recType	String
businessDate	Date
errorMsg	String

The output will contain the status of the request including validation errors, if any.

JSON Structure

```
{
  "statusMsg": null,
  "salesErrTbl": [
    {
      "store": null,
      "transactionNo": null,
      "recType": null,
      "businessDateDisplay": null,
      "businessDate": null,
      "errorMsg": null,
      "links": [],
      "hyperMediaContent": {
        "linkRDO": []
      }
    }
  ],
  "links": [],
  "hyperMediaContent": {
    "linkRDO": []
  }
}
```

Table Impact

TABLE	SELECT	INSERT	UPDATE	DELETE
SVC_RTLOG_DATA_LOAD	No	Yes	No	No

TABLE	SELECT	INSERT	UPDATE	DELETE
ITEM_LOC_SOH	Yes	No	Yes	No
SA_INPROGRESS_SALES	Yes	Yes	Yes	No
SVC_INPROGRESS_SALES	No	Yes	No	No
SVC_PROCESS_TRACKER	No	Yes	No	No

Store Search

Business Overview

This web service enables store search and returns aggregated store information.

Service Type

Get

ReST URL

/storeSearch?
searchString={searchString}&searchFilter={searchFilter}&sortAttrib={sortAttrib}&sortDir
ection={sortDirection}&pageSize={pageSize}&pageNumber={pageNumber}

Input Parameters

Parameter Name	Required	Description	Valid values
SearchString	Yes	Search string for locations ID or Name	NA
SearchFilter	Yes	Search all stores or assigned stores	ALL, ASSIGN
SortAttrib	No	Sort Attribute	STORENAME, AUDITOR, OSVALUE, ERRORCNT, DATASTATUS, OPENDAYS, OSDAYS and OSSUMS
SortDirection	No	Sort Direction	ASC, DESC
PageSize	No	Maximum number of locations to retrieve per page	NA
PageNumber	No	Result page to retrieve	NA

Output

Store

Store Name

Chain

Chain Name

Auditors
 Open Days
 Over Days
 Short Days
 Over Amount
 Short Amount
 Currency Code
 Error Count

Table Impact

TABLE	SELECT	INSERT	UPDATE	DELETE
LOC_TRAITS_MATRIX	Yes	No	No	No
SA_ERROR	Yes	No	No	No
SA_HQ_VALUE	Yes	No	No	No
SA_POS_VALUE	Yes	No	No	No
SA_STORE_DATA	Yes	No	No	No
SA_STORE_DAY	Yes	No	No	No
SA_SYS_VALUE	Yes	No	No	No
SA_TOTAL	Yes	No	No	No
SA_TRAN_HEAD	Yes	No	No	No
SA_USER_LOC_TRAITS	Yes	No	No	No
V_CHAIN	Yes	No	No	No
V_STORE	Yes	No	No	No

Summary of Open Store Days

Business Overview

This service provides, at a glance, the number of open stores for which the sales audit manager is responsible. The stores for which the user is responsible are those associated with the user in Sales Audit's employee maintenance via location traits.

Service Type

Get

ReSTURL

/summaryOpenStoreDay

Input Parameters

No input.

Output

Record Type --DATE, OLDER, ALL

- **For record type DATE:** five records of type date are displayed for today minus 1 through today minus 5
- **One record type OLDER:** is for store days older than today minus 5
- **One record type ALL:** for all store days

Record Date --Date of date type rows

Open Store Count

Table Impact

TABLE	SELECT	INSERT	UPDATE	DELETE
LOC_TRAITS_MATRIX	Yes	No	No	No
SA_STORE_DAY	Yes	No	No	No
SA_USER_LOC_TRAITS	Yes	No	No	No

Summary of Errors

Business Overview

This service provides, at a glance, the number outstanding errors on the specified days for stores for which the sales audit manager is responsible. An outstanding error is defined as an error that exists against a store day that has not been overridden.

Service Type

Get

ReST URL

/summaryError

Input Parameters

No input.

Output

Record Type --DATE, OLDER, ALL

- **For record type DATE:** five records of type date are displayed for today minus 1 through today minus 5
- **One record type OLDER:** is for store days older than today minus 5
- **One record type ALL:** for all store days

Record Date --Date of date type rows

Error Count

Table Impact

TABLE	SELECT	INSERT	UPDATE	DELETE
LOC_TRAITS_MATRIX	Yes	No	No	No
SA_ERROR	Yes	No	No	No
SA_STORE_DAY	Yes	No	No	No
SA_USER_LOC_TRAITS	Yes	No	No	No

Summary of Over/Short Amount

Business Overview

This service provides at a glance the sums of all overages and all shortages for all open stores on a given day for which the sales audit manager is responsible. If all locations to which the user is responsible have the same local currency, all monetary values will be displayed in the local currency. Otherwise, all monetary values will be displayed in the retailer's primary currency. If the Over/Short value for the store day is a positive value it is considered an overage, if the Over/Short value for the store day is a negative value it is a shortage.

Service Type

Get

ReST URL

/summaryOverShortAmount

Input Parameters

No input.

Output

Record Type --DATE, OLDER, ALL

- **For record type DATE:** Five records of type date are displayed for today minus 1 through today minus 5
- **One record type OLDER:** is for store days older than today minus 5
- **One record type ALL:** for all store days

Record Date --Date of date type rows
Over Amount
Short Amount
Currency Code

Table Impact

TABLE	SELECT	INSERT	UPDATE	DELETE
LOC_TRAITS_MATRIX	Yes	No	No	No
MV_CURRENCY_CONVERSION_RATES	Yes	No	No	No
SA_HQ_VALUE	Yes	No	No	No
SA_POS_VALUE	Yes	No	No	No
SA_STORE_DAY	Yes	No	No	No
SA_SYS_VALUE	Yes	No	No	No
SA_TOTAL	Yes	No	No	No
SA_USER_LOC_TRAITS	Yes	No	No	No
STORE	Yes	No	No	No

Summary of Over/Short Count

Business Overview

This service provides, at a glance, the count of overages and the count of shortages for all open stores on a given day for which the sales audit manager is responsible. If the Over/Short value for the store day is a positive value it is considered an overage, if the Over/Short value for the store day is a negative value it is a shortage.

Service Type

Get

ReST URL

/summaryOverShortCount

Input Parameters

No input.

Output

Record Type --DATE, OLDER, ALL

- **For record type DATE:** five records of type date are displayed for today minus 1 through today minus 5
- **One record type OLDER:** is for store days older than today minus 5

- **One record type ALL:** for all store days

Record Date --Date of date type rows

Over Count

Short Count

Table Impact

TABLE	SELECT	INSERT	UPDATE	DELETE
LOC_TRAITS_MATRIX	Yes	No	No	No
SA_HQ_VALUE	Yes	No	No	No
SA_POS_VALUE	Yes	No	No	No
SA_STORE_DAY	Yes	No	No	No
SA_SYS_VALUE	Yes	No	No	No
SA_TOTAL	Yes	No	No	No
SA_USER_LOC_TRAITS	Yes	No	No	No
STORE	Yes	No	No	No

6

Scheduled Integration

This chapter provides a summary of integrations that are scheduled either to be run once per day or periodically throughout the day. There are generally two types of integrations - those that expect or produce files and those that move data between integration tables, also referred to as Bulk Data Integration (BDI).

File-based Integration Overview

Merchandising and Sales Audit require that all inbound and outbound file-based transactions adhere to standard file layouts. There are two types of file layouts: detail-only and master-detail, which are described in the sections below.

Standard File Layouts

The Merchandising interface library supports two standard file layouts: one for master/detail processing, and one for processing detail records only. True sub-details are not supported within the Merchandising base package interface library functions.

A 5-character identification code or record type identifies all records within an I/O file, regardless of file type. The following includes common record type values:

- FHEAD-File Header
- FDETL-File Detail
- FTAIL-File Tail
- THEAD-Transaction Header
- TDETL-Transaction Detail
- TTAIL-Transaction Tail

Each line of the file must begin with the record type code followed by a 10-character record ID.

Detail-Only Files

File layouts have a standard file header record, a detail record for each transaction to be processed, and a file trailer record. Valid record types are FHEAD, FDETL, and FTAIL.

Example 6-1 Detail-Only Files

```
FHEAD0000000001STKU1996010100000019960929
FDETL0000000002SKU100000040000011011
FDETL0000000003SKU100000050003002001
FDETL0000000004SKU100000050003002001
FTAIL00000000050000000003
```

Master and Detail Files

File layouts consists of:

- Standard file header record
- Set of records for each transaction to be processed
- File trailer record.

The transaction set consists of:

- Transaction set header record
- Transaction set detail for detail within the transaction
- Transaction trailer record

Valid record types are FHEAD, THEAD, TDETL, TTAIL, and FTAIL.

Example 6-2 Master and Detail Files

```
FHEAD0000000001RTV 19960908172000
THEAD000000000200000000000001199609091202000000000003R
TDETL000000000300000000000001000001SKU10000012
TTAIL0000000004000001
THEAD000000000500000000000002199609091202001215720131R
TDETL000000000600000000000002000001UPC400100002667
TDETL0000000007000000000000020000021UPC400100002643 0
TTAIL0000000008000002
FTAIL00000000090000000007
```

Record Name	Field Name	Field Type	Default Value	Description
File Header	File Type Record Descriptor	Char(5)	FHEAD	Identifies file record type.
	File Line Identifier	Number(10)	Specified by external system	Line number of the current file.
	File Type Definition	Char(4)	N/A	Identifies transaction type.
	File Create Date	Date	Create date	Date file was written by external system.
Transaction Header	File Type Record Descriptor	Char(5)	THEAD	Identifies file record type.
	File Line Identifier	Number(10)	Specified by external system	Line number of the current file.
	Transaction Set Control Number	Char(14)	Specified by external system	Used to force unique transaction check.
	Detail Sequence Number	Char(6)	Specified by external system	Sequential number assigned to detail records within a transaction.
Transaction Detail	File Type Record Descriptor	Char(5)	TDETL	Identifies file record type.
	File Line Identifier	Number(10)	Specified by external system	Line number of the current file.

Record Name	Field Name	Field Type	Default Value	Description
	Transaction Set Control Number	Char(14)	Specified by external system	Used to force unique transaction check.
	Detail Sequence Number	Char(6)	Specified by external system	Sequential number assigned to detail records within a transaction.
Transaction Trailer	File Type Record Descriptor	Char(5)	TTAIL	Identifies file record type.
	File Line Identifier	Number(10)	Specified by external system	Line number of the current file.
	Transaction Detail Line Count	Number(6)	Sum of detail lines	Sum of the detail lines within a transaction.
File Trailer	File Record Type Descriptor	Char(5)	FTAIL	Identifies file record type.
	File Line Identifier	Number(10)	Specified by external system	Line number of the current file.
	Total Transaction Line Count	Number(10)	Sum of all transaction lines	All lines in file less the file header and trailer records.

Integration Batch Wrapper FAQ

See also the "Batch Wrapper Overview" in Chapter 1 of the *Merchandising Operations Guide, Volume 1*.

For input files, are there any specific file format names?

Usually, the input file pattern is the same as the batch name. This can be seen in the `ParameterValue` column of the Job tab in the Batch Schedule spreadsheet. For example, `dealupld` has this parameter value:

```
dealupld #SysOpt.dbwallet dealupld
```

The input file pattern is the 3rd parameter (`dealupld`) so the filename should have `dealupld` in it (for example, `dealupld_input`, `input_dealupld`, and so on). There are some batch programs that should start with a specific pattern like `saimptlogi` - the input file should start with `RTLOG*`. This can be seen as well in the Batch Schedule spreadsheet:

```
#SysOpt.dbwallet RTLOG
```

How and where will files be moved?

The input file (zipped) should be in the SFTP batch/incoming folder.

There are two ways to SFTP the input files. For bulk upload, ensure that a COMPLETE file is present in the batch/incoming/COMMAND folder. For individual file upload, ensure that each

input file has a corresponding, complete file (for example, `RTLOG_1521.zip` will have a corresponding `RTLOG_1521.zip.complete` file in the batch/incoming folder).

The batch wrappers will take care of moving this to the input folder (`data/in`) given that the filename of the zip file corresponds to the expected file name pattern.

When will files be unzipped?

This will be unzipped when it is moved to the input folder (`data/in`).

What will happen to the files once they are processed?

Successfully processed files will be move to the processed folder (`data/processed`). Files with errors will remain in the input folder (`data/in`) and corresponding errors will be logged in the error log.

Will processed file be archived?

Yes. Input files will be archived in the processed folder (`data/processed`), while any output file will be archived in the archive folder (`data/archive`)

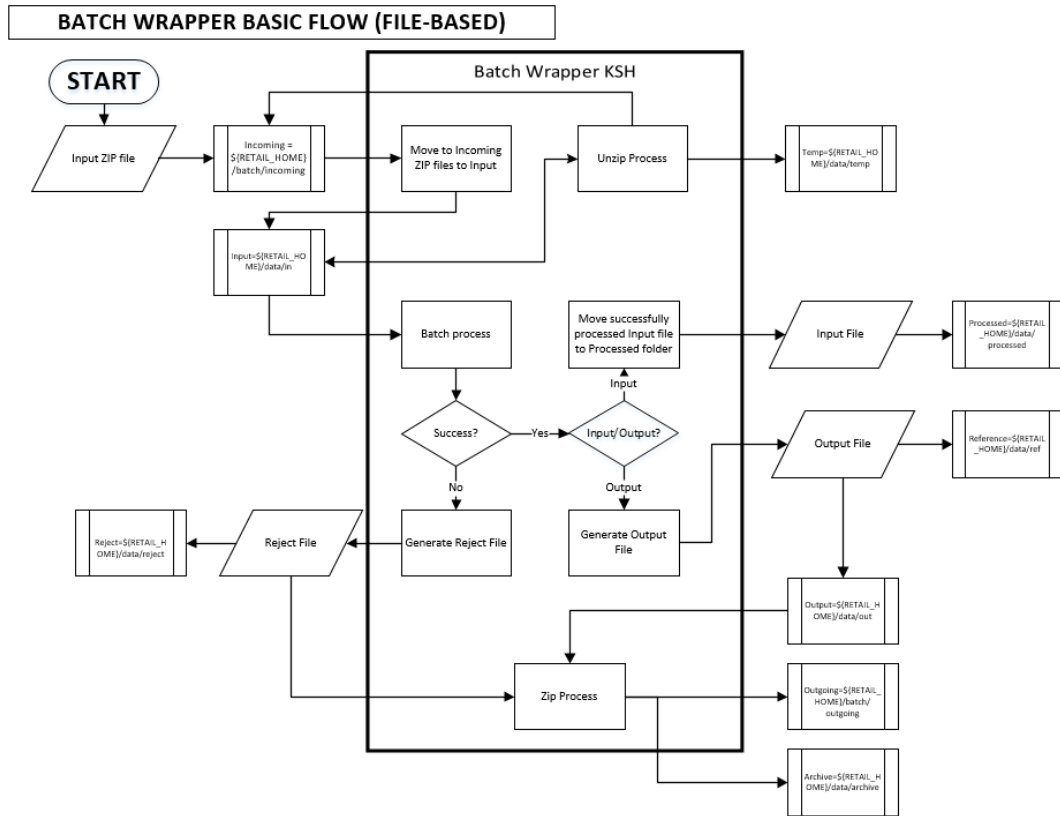
What will happen if the file is rejected?

Reject files will be created as applicable and they will be sent to the SFTP outgoing folder (`batch/outgoing`) as well as archived in the archive folder (`data/archive`). Depending on the batch, if the program completes successfully even with rejected records then the rejected input file will be moved to the processed folder (`data/processed`). But if the batch aborts, then the rejected input file will remain in the input folder (`data/in`).

How will rejected files be reprocessed?

Reprocessing the files would require manual intervention. Remove the files with errors from the input folder (`data/in`). Those input files would need to be corrected and resent.

File Movement Flow



Input/Output File Naming Convention

Batch	Input File Name	Zip File Name
dealupld	*dealupld*	*dealdupld*.zip
ediupack	*ediupack*	*ediupack*.zip
ediupavl	*ediupavl*	*ediupavl*.zip
lcup798	*lcup798*	*lcup798*.zip
lcupld	*lcupld*	*lcupld*.zip
cmpupld	*cmpupld*	*cmpupld*.zip
fcostmplupld.ksh	*fcostmplupld*	*fcostmplupld*.zip
htsupld	*htsupld*	*htsupld*.zip
otbupld	*otbupld*	*otbupld*.zip
tranupld	*tranupld*	*tranupld*.zip
fcustomerupload.ksh	*fcustomerupload*	*fcustomerupload*.zip

Batch	Input File Name	Zip File Name
iindbatch.ksh	Input file name is provided by user. Must end in *.xml Template name must correspond to S9T_TEMPLATE.TEMPLATE_KEY	*<input file name>*.zip
poindbatch.ksh	Input file name is provided by user. Must end in *.xml Template name must correspond to S9T_TEMPLATE.TEMPLATE_KEY	*<input file name>*.zip
replindbatch.ksh	Input file name is provided by user. Must end in *.xml Template name must correspond to S9T_TEMPLATE.TEMPLATE_KEY	*<input file name>*.zip
saimpadj	*saimpadj*	*saimpadj*.zip
load_item_forecast.ksh	*demand*	*demand*.zip
lcmt700	*lcmt700*	*lcmt700*.zip
lcmt707	*lcmt707*	*lcmt707*.zip
resa2sim	SIMT*	No zip file. Input file is from saexpsim.
resa2dw	RDWT* RDWF* RDWS* RDWC*	No zip file. Input file is from saexpdw.
lcmt730	*lcmt730*	*lcmt730*.zip
lcmt798	*lcmt798*	*lcmt798*.zip
lifstkup	*lifstkup*	*lifstkup*.zip
ordinvupld	ORIN*	ORIN*.zip
sa_rules_total_upload.ksh	sartexp_<table name>*	sartexp*.zip
saimptlogfin	storedayfile	No zip file. Input file is from sagetref.
saimptlogi	RTLOG_<store>_<datetime>.dat	RTLOG*.zip
saimptlogtdup_upd	storedayfile storeposfile	No zip file. Input file is from sagetref.
savouch	*savouch*	*savouch*.zip
stlgdnld	*stlgdnld*	*stlgdnld*.zip
stockcountupload.ksh	STK*.<file extension>	STK*.zip
trandataload.ksh	*trandatoload*	*trandatoload*.zip
uploadsales.ksh	POSU_<store>_<tran_date>_ <sysdate>.<thread_val>	POSU*.zip
wfslsupld.ksh	*wfslsupld*	*wfslsupld*.zip
wfordupld.ksh	wford*.dat	wford*.zip
wfretupld.ksh	wfreturn*.dat	wfreturn*.zip

Bulk Data Integration Overview

Oracle Bulk Data Integration (BDI) is a solution that is part of the Oracle Retail Integration Cloud Service that defines the architecture and infrastructure used to move bulk data among Oracle Retail applications.

In a Bulk Data Integration system, message families are represented as interface modules. Each interface module (for example, DiffGrp_Fnd) contains a Merchandising component that takes care of pulling and staging data for publication to the External BDI system. Interface modules are divided by functional entity (for example, Item Master, Stores, Diffs, and so on).

For more information on BDI, see the Oracle Retail Integration Cloud Service documentation on docs.oracle.com/retail.

Outbound Scheduled Integration

This section provides a summary of integrations that are scheduled either to be run once per day or periodically throughout the day to send data from Merchandising or Sales Audit to another solution. It includes both file-based and BDI-based integrations.

Foundation Data

Merchandising publishes foundation data for many other solution areas, including stores, warehouses, omni-channel, and so on.

The following scheduled outbound integrations are included in this functional area:

- Brand Publication API (BDI_Brand_Fnd_PF_From_RMS_EOW_JOB)
- Calendar Publication API (BDI_Calendar_Fnd_PF_From_RMS_EOW_JOB)
- Code Detail Publication API (BDI_CodeDetail_Fnd_PF_From_RMS_JOB)
- Code Head Publication API (BDI_CodeHead_Fnd_PF_From_RMS_JOB)
- Company-wide Closings and Company Closed Exceptions (BDI_CompanyClosed_Fnd_PF_From_RMS_JOB)
- Currency Conversion Rates Publication API (BDI_CurrConvRates_Fnd_PF_From_RMS_EOW_JOB)
- Delivery Slot Publication API (BDI_DeliverySlot_Fnd_PF_From_RMS_JOB)
- Diff Group Export (export_diffgrp.ksh)
- Diff Group Publication API (BDI_DiffGrp_Fnd_PF_From_RMS_JOB)
- Diff ID and Type Export (export_diffs.ksh)
- Diff ID Publication API (BDI_Diff_Fnd_PF_From_RMS_JOB)
- Finisher Address Publication API (BDI_FinisherAddr_Fnd_PF_From_RMS_JOB)
- Location Closed Publication API (BDI_LocClosed_Fnd_PF_From_RMS_JOB)
- Merch Hierarchy Publication API (BDI_MerchHier_Fnd_PF_From_RMS_JOB)
- Merchandise Hierarchy Export (export_merchhier.ksh)
- Organization Hierarchy Publication API (BDI_OrgHier_Fnd_PF_From_RMS_JOB)

- Organizational Hierarchy Export (export_orghier.ksh)
- Partner Address Publication API (BDI_PartnerAddr_Fnd_PF_From_RMS_JOB)
- Partner Org Unit Publication API (BDI_PartOrgUnit_Fnd_PF_From_RMS_JOB)
- Partner Publication API (BDI_Partner_Fnd_PF_From_RMS_JOB)
- Store Address Publication API (BDI_StoreAddr_Fnd_PF_From_RMS_JOB)
- Store Hours Publication API (BDI_StoreHours_Fnd_PF_From_RMS_JOB)
- Store Publication API (BDI_Store_Fnd_PF_From_RMS_JOB)
- Stores Export (export_stores.ksh)
- Supplier Address Publication API (BDI_SupplierAddr_Fnd_PF_From_RMS_JOB)
- Sups Publication API (BDI_Supplier_Fnd_PF_From_RMS_JOB)
- Tax Download - Brazil (taxdnld)
- Ticket Download (tcktdnld)
- UDA Publication API (BDI_Uda_Fnd_PF_From_RMS_JOB)
- UDA Values Publication API (BDI_UdaValues_Fnd_PF_From_RMS_JOB)
- UOM Class Publication API (BDI_UomClass_Fnd_PF_From_RMS_JOB)
- UOM Conversion Publication API (BDI_UomConversion_Fnd_PF_From_RMS_JOB)
- VAT Codes, Regions, and Rates Export (export_vat.ksh)
- VAT Publication API (BDI_Vat_Fnd_PF_From_RMS_JOB)
- Warehouse (BDI_Wh_Fnd_PF_From_RMS_JOB)
- Warehouse Address Publication API (BDI_WhAddr_Fnd_PF_From_RMS_JOB)

Brand Publication API (BDI_Brand_Fnd_PF_From_RMS_EOW_JOB)

This section describes the Brand Publication BDI.

Functional Area

Foundation

Business Overview

This API publishes the creation, update and deletion of brand information to the RIB such that all the downstream applications (including an external system) may subscribe to it and have information in sync with Merchandising. Publishing is provided synchronously in a near-real time manner.

New Brand

Creating a new brand triggers a message to be sent to notify external systems. Brand name and description are sent as part of the create message.

Table 6-1 Brand – Create and Update

Message Element	Included?	Notes
Brand name	Always	This field contains the brand name.
Brand Description	Always	This field contains the description of the brand.

Updated Brand

Updating a brand triggers a message to be sent to notify external systems. Brand name and description are sent as part of the update message.

Deleted Brand

Deleting a brand triggers a message to be sent to notify external systems. Brand name is sent as part of the delete message.

Table 6-2 Brand – Delete

Message Element	Included?	Notes
Brand name	Always	This field contains the brand name being deleted.

Message XSD

Here are the filenames that correspond with each message type. Please consult the RIB documentation for each message type to get a detailed picture of the composition of each message.

Message Types	Message Type Description	XML Schema Definition (XSD)
brandcre	Brand Create Message	BrandDesc.xsd
brandupd	Brand Modify Message	BrandDesc.xsd
branddel	Brand Delete Message	BrandRef.xsd

Calendar Publication API (BDI_Calendar_Fnd_PF_From_RMS_EOW_JOB)

This section describes the Calendar Publication BDI.

Functional Area

Foundation

Business Overview

BDI (Bulk Data Integration) is an integration layer that facilitates the bulk transfer of Calendar information (2 prior years, current year, 2 future years) from Merchandising to other Oracle Retail Applications. On this particular integration stream, the data flow is from Merchandising to BDI, and then BDI to downstream applications. To accomplish this data transfer, BDI will be calling a Merchandising-owned API that will pull data from Merchandising and deliver these to the BDI integration layer. This API will be in the form of a PLSQL function inside a PLSQL package.

Package Impact

Filename: **bdifoundationb.pls.pls**

```
BDI_FOUNDATION_SQL.CALENDAR_UP(O_error_message IN OUT
RTK_ERRORS.RTK_TEXT%TYPE,
NUMBER,
I_job_context IN VARCHAR2)
O_control_id IN OUT
```

This function begins by calling a BDI function that signals the start of the interface process. The BDI function will update the internal BDI control tables to track the progress of the API.

A DML insert statement is then executed to populate the BDI outbound table that resides in the BDI_RMS_INT_SCHEMA schema. This outbound table is loaded with records from the Merchandising V_BDI_DAY_LEVEL_CALENDAR view.

After the insert, another call to a BDI function is performed to signify the successful loading of records. This will update the internal BDI control tables.

A database commit is issued, and the control Id is returned by the API.

Data Definition XML

The BDI interface staging tables are generated based on the XML schema definition

Data Flow	Description	XML Schema Definition (XSD)
Calendar	Calendar upload to BDI	Calendar_Fnd_BdiInterfaceModule.xml

Table Impact

TABLE	SELECT	INSERT	UPDATE	DELETE
CALENDAR_OUT	No	Yes	No	No
V_BDI_DAY_LEVEL_CALENDAR	Yes	No	No	No

Code Detail Publication API (BDI_CodeDetail_Fnd_PF_From_RMS_JOB)

This section describes the Code Detail Publication BDI.

Functional Area

Cross Pillar

Business Overview

BDI (Bulk Data Integration) is an integration layer that facilitates the bulk transfer of Code Detail information from Merchandising to other Oracle Retail Applications. On this particular integration stream, the data flow is from Merchandising to BDI, and then BDI to downstream applications. To accomplish this data transfer, BDI will be calling a Merchandising-owned API that will pull data from Merchandising and deliver these to the BDI integration layer. This API will be in the form of a PLSQL function inside a PLSQL package.

Package Impact

Filename: **bdicrosspillarb.pls**

```
BDI_CROSS_PILLAR_SQL.CODE_DETAIL_UP(
    O_error_message IN OUT RTK_ERRORS.RTK_TEXT%TYPE,
    O_control_id    IN OUT NUMBER,
    I_job_context   IN     VARCHAR2)
```

This function begins by calling a BDI function that signals the start of the interface process. The BDI function updates the internal BDI control tables to track the progress of the API.

A DML insert statement is then executed to populate the BDI outbound table that resides in the BDI_RMS_INT_SCHEMA schema. This outbound table is loaded with records from the Merchandising CODE_DETAIL table.

After the insert, another call to a BDI function is performed to signify the successful loading of records. This updates the internal BDI control tables.

A database commit is issued, and the control ID is returned by the API.

Data Definition XML

The BDI interface staging tables are generated based on the XML schema definition

Data Flow	Description	XML Schema Definition (XSD)
Code Detail	Code Detail upload to BDI	CodeDetail_Fnd_BdiInterfaceModule.xml

Table Impact

TABLE	SELECT	INSERT	UPDATE	DELETE
CODE_DETAIL_OUT	No	Yes	No	No
CODE_DETAIL	Yes	No	No	No

Code Head Publication API (BDI_CodeHead_Fnd_PF_From_RMS_JOB)

This section describes the Code Head Publication BDI.

Functional Area

Cross Pillar

Business Overview

BDI (Bulk Data Integration) is an integration layer that facilitates the bulk transfer of Code Head information from Merchandising to other Oracle Retail Applications. On this particular integration stream, the data flow is from Merchandising to BDI, and then BDI to downstream applications. To accomplish this data transfer, BDI will be calling a Merchandising-owned API that will pull data from Merchandising and deliver these to the BDI integration layer. This API will be in the form of a PLSQL function inside a PLSQL package.

Package Impact

Filename: **bdicrosspillarb.pls**

```
BDI_CROSS_PILLAR_SQL.CODE_HEAD_UP(O_error_message IN OUT
RTK_ERRORS.RTK_TEXT%TYPE,
                                O_control_id   IN OUT NUMBER,
                                I_job_context  IN      VARCHAR2)
```

This function begins by calling a BDI function that signals the start of the interface process. The BDI function will update the internal BDI control tables to track the progress of the API.

A DML insert statement is then executed to populate the BDI outbound table that resides in the BDI_RMS_INT_SCHEMA schema. This outbound table is loaded with records from the Merchandising CODE_HEAD table.

After the insert, another call to a BDI function is performed to signify the successful loading of records. This will update the internal BDI control tables.

A database commit is issued, and the control Id is returned by the API.

Data Definition XML

The BDI interface staging tables are generated based on the XML schema definition

Data Flow	Description	XML Schema Definition (XSD)
Code Head	Code Head upload to BDI	CodeHead_Fnd_BdiInterfaceModule.xml

Table Impact

TABLE	SELECT	INSERT	UPDATE	DELETE
CODE_HEAD_OUT	No	Yes	No	No
CODE_HEAD	Yes	No	No	No

Company-wide Closings and Company Closed Exceptions (BDI_CompanyClosed_Fnd_PF_From_RMS_JOB)

This section describes the Company-wide Closings and Company Closed Exceptions Publication BDI.

Functional Area

Foundation

Design Overview

BDI (Bulk Data Integration) is an integration layer that facilitates the bulk transfer of Store information from Merchandising to other Oracle Retail Applications. On this particular integration stream, the data flow is from Merchandising to BDI, and then BDI to downstream applications. To accomplish this data transfer, BDI will be calling a

Merchandising-owned API that will pull data from Merchandising and deliver these to the BDI integration layer. This API will be in the form of a PLSQL function inside a PLSQL package.

Package Impact

The following packages are impacted:

Filename: **bdifoundations.pls**

```
BDI_FOUNDATION_SQL.COMPANY_CLOSED_UP (
    O_error_message IN OUT RTK_ERRORS.RTK_TEXT%TYPE,
    O_control_id    IN OUT NUMBER,
    I_job_context   IN     VARCHAR2)
```

Filename: **bdifoundationb.pls**

```
BDI_FOUNDATION_SQL.COMPANY_CLOSED_UP (
    O_error_message IN OUT RTK_ERRORS.RTK_TEXT%TYPE,
    O_control_id    IN OUT NUMBER,
    I_job_context   IN     VARCHAR2)
```

This function begins by calling a BDI function that signals the start of the interface process. The BDI function will update the internal BDI control tables to track the progress of the API.

A DML insert statement is then executed to populate the BDI outbound table that resides in the BDI_RMS_INT_SCHEMA schema. This outbound table is loaded with records from the Merchandising company closed and company closed exception table.

After the insert, another call to a BDI function is performed to signify the successful loading of records. This will update the internal BDI control tables.

A database commit is issued, and the control Id is returned by the API.

Data Definition XML

The BDI interface staging tables are generated based on the XML schema definition.

Data Flow	Description	XML Schema Definition (XSD)
Company Closed	Company Closed upload to BDI	CompanyClosed_Fnd_BdiInterfaceModule.xml
Company Closed Exceptions	Company Closed Exceptions upload to BDI	CompanyClosedExcep_Fnd_BdiInterfaceModule.xml

Tables

TABLE	SELECT	INSERT	UPDATE	DELETE
COMPANY_CLOSED_OUT	No	Yes	No	No
COMPANY_CLOSED_EXCEP_OUT	No	Yes	No	No
COMPANY_CLOSED_ECXEP	Yes	No	No	No
COMPANY_CLOSED	Yes	No	No	No

Currency Conversion Rates Publication API (BDI_CurrConvRates_Fnd_PF_From_RMS_EOW_JOB)

This section describes the Currency Conversion Rates Publication BDI.

Functional Area

Foundation

Business Overview

BDI (Bulk Data Integration) is an integration layer that facilitates the bulk transfer of Currency conversion rates information from Merchandising to other Oracle Retail Applications. On this particular integration stream, the data flow is from Merchandising to BDI, and then BDI to downstream applications. To accomplish this data transfer, BDI will be calling a Merchandising-owned API that will pull data from Merchandising and deliver these to the BDI integration layer. This API will be in the form of a PLSQL function inside a PLSQL package.

Package Impact

Filename: **bdifoundationb.pls.pls**

```
BDI_FOUNDATION_SQL.CURR_CONV_RATES_UP(
    O_error_message IN OUT RTK_ERRORS.RTK_TEXT%TYPE,
    O_control_id    IN OUT NUMBER,
    I_job_context   IN     VARCHAR2)
```

This function begins by calling a BDI function that signals the start of the interface process. The BDI function will update the internal BDI control tables to track the progress of the API.

A DML insert statement is then executed to populate the BDI outbound table that resides in the BDI_RMS_INT_SCHEMA schema. This outbound table is loaded with records from the Merchandising MV_CURRENCY_CONVERSION_RATES materialized view.

After the insert, another call to a BDI function is performed to signify the successful loading of records. This will update the internal BDI control tables.

A database commit is issued, and the control Id is returned by the API.

Data Definition XML

The BDI interface staging tables are generated based on the XML schema definition

Data Flow	Description	XML Schema Definition (XSD)
Currency Conversion Rates	Currency Conversion Rates upload to BDI	CurrConvRates_Fnd_BdiInterfaceModule.xml

Table Impact

TABLE	SELECT	INSERT	UPDATE	DELETE
CURR_CONV_RATES_OUT	No	Yes	No	No
MV_CURRENCY_CONVERSION_RATES	Yes	No	No	No
SYSTEM_OPTIONS	Yes	No	No	No
STORE	Yes	No	No	No
WH	Yes	No	No	No

Delivery Slot Publication API (BDI_DeliverySlot_Fnd_PF_From_RMS_JOB)

This section describes the Delivery Slot Publication BDI.

Functional Area

Cross Pillar

Business Overview

BDI (Bulk Data Integration) is an integration layer that facilitates the bulk transfer of Delivery Slot information from Merchandising to other Oracle Retail Applications. On this particular integration stream, the data flow is from Merchandising to BDI, and then BDI to downstream applications. To accomplish this data transfer, BDI will be calling a Merchandising-owned API that will pull data from Merchandising and deliver these to the BDI integration layer. This API will be in the form of a PLSQL function inside a PLSQL package.

Package Impact

Filename: **bdicrosspillarb.pls**

```
BDI_CROSS_PILLAR_SQL.DELIVERY_SLOT_UP (
    O_error_message IN OUT RTK_ERRORS.RTK_TEXT%TYPE,
    O_control_id    IN OUT NUMBER,
    I_job_context   IN     VARCHAR2)
```

This function begins by calling a BDI function that signals the start of the interface process. The BDI function will update the internal BDI control tables to track the progress of the API.

A DML insert statement is then executed to populate the BDI outbound table that resides in the BDI_RMS_INT_SCHEMA schema. This outbound table is loaded with records from the Merchandising DELIVERY_SLOT table.

After the insert, another call to a BDI function is performed to signify the successful loading of records. This will update the internal BDI control tables.

A database commit is issued, and the control Id is returned by the API.

Data Definition XML

The BDI interface staging tables are generated based on the XML schema definition

Data Flow	Description	XML Schema Definition (XSD)
Delivery Slot	Delivery Slot upload to BDI	DeliverySlot_Fnd_BdiInterfaceModule.xml

Table Impact

TABLE	SELECT	INSERT	UPDATE	DELETE
DELIVERY_SLOT_OUT	No	Yes	No	No
DELIVERY_SLOT	Yes	No	No	No

Diff Group Export (export_diffgrp.ksh)

Module Name	export_diffgrp.ksh
Description	Extraction of differentiator groups data.
Functional Area	Foundation
Module Type	Integration
Module Technology	ksh
Catalog ID	RMS255
Wrapper Script	rmswrap_shell_out.ksh

Schedule

Oracle Retail Merchandising Batch Schedule

Design Overview

This batch job will extract new, updated and deleted Merchandising diff group information into a flat file. Data to be extracted will be pulled off from the differentiator group tables. The mode (full vs. delta) will be an input parameter for this batch. The mode will allow a full extract (all diff group records in Merchandising) as well as delta processing (all diff group record changes in the time frame passed in the program) of data. For a full extract, records will be retrieved from the differentiator group tables. For a delta extract, the action type and diff group ID will be retrieved from the differentiator group staging export table and the attributes will be retrieved from the differentiator group tables.

Restart/Recovery

N/A

I/O Specification

Integration Type	Extract from Merchandising
File Name	diffgrphdr_date_[full/delta]_[#ofLines].dat diffgrpdtl_date_[full/delta]_[#ofLines].dat
Integration Contract	IntCon000212.html IntCon000213.html

Design Assumptions

N/A

Diff Group Publication API (BDI_DiffGrp_Fnd_PF_From_RMS_JOB)

This section describes the Diff Group Publication BDI.

Functional Area

Foundation

Business Overview

BDI (Bulk Data Integration) is an integration layer that facilitates the bulk transfer of Diff Groups from Merchandising to other Oracle Retail Applications. On this particular integration stream, the data flow is from Merchandising to BDI, and then BDI to downstream applications. To accomplish this data transfer, BDI will be calling a Merchandising-owned API that will pull data from Merchandising and deliver these to the BDI integration layer. This API will be in the form of a PLSQL function inside a PLSQL package.

Package Impact

Filename: **bdicrosspillarb.pls**

```
BDI_CROSS_PILLAR_SQL.DIFF_GROUP_UP(O_error_message IN OUT VARCHAR2,
                                     O_control_id   IN OUT NUMBER,
                                     I_job_context   IN      VARCHAR2)
```

This function begins by calling a BDI function that signals the start of the interface process. The BDI function will update the internal BDI control tables to track the progress of the API.

A DML insert statement is then executed to populate the BDI outbound tables that reside in the BDI_RMS_INT_SCHEMA schema. These outbound tables are loaded with records from the Merchandising Diff Group head and detail tables.

After the insert, another call to a BDI function is performed to signify the successful loading of records. This will update the internal BDI control tables.

A database commit is issued, and the control Id is returned by the API.

Data Definition XML

The BDI interface staging tables are generated based on the XML schema definition

Data Flow	Description	XML Schema Definition (XSD)
Diff Group	Diff Group upload to BDI	DiffGrp_Fnd_BdiInterfaceModule.xml

Table Impact

TABLE	SELECT	INSERT	UPDATE	DELETE
DIFF_GRP_OUT	No	Yes	No	No

TABLE	SELECT	INSERT	UPDATE	DELETE
DIFF_GRP_DTL_OUT	No	Yes	No	No
DIFF_GROUP_HEAD	Yes	No	No	No
DIFF_TYPE	Yes	No	No	No
DIFF_GROUP_DETAIL	Yes	No	No	No

Diff ID and Type Export (export_diffs.ksh)

Module Name	export_diffs.ksh
Description	Extraction of differentiator's data defined for a differentiator type.
Functional Area	Foundation
Module Type	Integration
Module Technology	ksh
Catalog ID	256
Wrapper Script	rmswrap_shell_out.ksh

Schedule

Oracle Retail Merchandising Batch Schedule

Design Overview

This batch job will extract new, updated and deleted Merchandising differentiator information into a flat file. Data to be extracted will be pulled off from the differentiator extract staging and the differentiator IDs table.

The mode (full vs. delta) will be an input parameter for this batch. The mode will allow a full extract (all differentiator records in Merchandising) as well as delta processing (all differentiator record changes in the time frame passed in the program) of data.

For a full extract, records will be solely retrieved from the differentiator IDs table. For a delta extract, the action type and differentiator ID will be retrieved from the differentiator export staging table and the attributes will be retrieved from the differentiator IDs table.

Restart/Recovery

N/A

I/O Specification

Integration Type	Extract from Merchandising
File Name	diffs_date_[full/delta]_[#ofLines].dat
Integration Contract	IntCon000206.html

Design Assumptions

N/A

Diff ID Publication API (BDI_Diff_Fnd_PF_From_RMS_JOB)

This section describes the Diff ID Publication BDI.

Functional Area

Foundation

Business Overview

BDI (Bulk Data Integration) is an integration layer that facilitates the bulk transfer of Diff IDs from Merchandising to other Oracle Retail Applications. On this particular integration stream, the data flow is from Merchandising to BDI, and then BDI to downstream applications. To accomplish this data transfer, BDI will be calling a Merchandising-owned API that will pull data from Merchandising and deliver these to the BDI integration layer. This API will be in the form of a PLSQL function inside a PLSQL package.

Package Impact

This section describes the package impact.

Bulk Interface Module

Filename: bdicrosspillarb.pls

```
BDI_CROSS_PILLAR_SQL.DIFF_UP(O_error_message IN OUT VARCHAR2,
                             O_control_id    IN OUT NUMBER,
                             I_job_context   IN     VARCHAR2)
```

This function begins by calling a BDI function that signals the start of the interface process. The BDI function will update the internal BDI control tables to track the progress of the API.

A DML insert statement is then executed to populate the BDI outbound table that resides in the BDI_RMS_INT_SCHEMA schema. This outbound table is loaded with records from the Merchandising Diff tables.

After the insert, another call to a BDI function is performed to signify the successful loading of records. This will update the internal BDI control tables.

A database commit is issued, and the control Id is returned by the API.

Data Definition XML

The BDI interface staging tables are generated based on the XML schema definition

Data Flow Type	Description	XML Schema Definition (XSD)
Diff Id	Diff Id upload to BDI	Diff_Fnd_BdiInterfaceModule.xml

Table Impact

TABLE	SELECT	INSERT	UPDATE	DELETE
DIFF_OUT	No	Yes	No	No

TABLE	SELECT	INSERT	UPDATE	DELETE
DIFF_IDS	Yes	No	No	No
DIFF_TYPE	Yes	No	No	No

Finisher Address Publication API (BDI_FinisherAddr_Fnd_PF_From_RMS_JOB)

This section describes the Finisher Address Publication BDI.

Functional Area

Foundation

Business Overview

BDI (Bulk Data Integration) is an integration layer that facilitates the bulk transfer of Finisher Address positions from Merchandising to other Oracle Retail Applications. On this particular integration stream, the data flow is from Merchandising to BDI, and then BDI to downstream applications. To accomplish this data transfer, BDI will be calling a Merchandising-owned API that will pull data from Merchandising and deliver these to the BDI integration layer. This API will be in the form of a PLSQL function inside a PLSQL package

Package Impact

Filename: bdfoundations/b.pls

```
BDI_FOUNDATION_SQL.FINISHER_ADDR_UP(O_error_message IN OUT VARCHAR2,
                                     O_control_id     IN OUT NUMBER,
                                     I_job_context    IN   VARCHAR2)
```

This function begins by calling a BDI function that signals the start of the interface process. The BDI function will update the internal BDI control tables to track the progress of the API.

A DML insert statement is then executed to populate the BDI outbound table that resides in the BDI_RMS_INT_SCHEMA schema. This outbound table is loaded with records from the Merchandising Finisher Address tables.

After the insert, another call to a BDI function is performed to signify the successful loading of records. This will update the internal BDI control tables.

A database commit is issued, and the control Id is returned by the API.

Data Definition XML

The BDI interface staging tables are generated based on the XML schema definition

Data Flow	Description	XML Schema Definition (XSD)
Finisher Address	Finisher Address upload to BDI	FinisherAddr_Fnd_BdiInterfaceModule.xml

Table Impact

TABLE	SELECT	INSERT	UPDATE	DELETE
FINISHER_ADDR_OUT	No	Yes	No	No
ADD_TYPE_MODULE	Yes	No	No	No
WH	Yes	No	No	No
V_ADD_TYPE_TL	Yes	No	No	No
COUNTRY	Yes	No	No	No
STATE	Yes	No	No	No
ADDR	Yes	No	No	No
PARTNER	Yes	No	No	No

Location Closed Publication API (BDI_LocClosed_Fnd_PF_From_RMS_JOB)

This section describes the Location Closed Publication BDI.

Functional Area

Foundation

Design Overview

BDI (Bulk Data Integration) is an integration layer that facilitates the bulk transfer of Store information from Merchandising to other Oracle Retail Applications. On this particular integration stream, the data flow is from Merchandising to BDI, and then BDI to downstream applications. To accomplish this data transfer, BDI will be calling a Merchandising-owned API that will pull data from Merchandising and deliver these to the BDI integration layer. This API will be in the form of a PLSQL function inside a PLSQL package.

Package Impact

The following packages are impacted by this BDI:

Bulk Interface Module

The following build interface module packages are impacted:

Filename: bdfoundations.pls

```
FUNCTION LOCATION_CLOSED_UP(O_error_message IN OUT RTK_ERRORS.RTK_TEXT%TYPE,
                           O_control_id     IN OUT NUMBER,
                           I_job_context    IN     VARCHAR2)
```

Filename: bdfoundationb.pls

```
FUNCTION LOCATION_CLOSED_UP(O_error_message IN OUT RTK_ERRORS.RTK_TEXT%TYPE,
                           O_control_id     IN OUT NUMBER,
                           I_job_context    IN     VARCHAR2)
```

This function begins by calling a BDI function that signals the start of the interface process. The BDI function will update the internal BDI control tables to track the progress of the API.

A DML insert statement is then executed to populate the BDI outbound table that resides in the BDI_RMS_INT_SCHEMA schema. This outbound table is loaded with records from the Merchandising Location closed table.

After the insert, another call to a BDI function is performed to signify the successful loading of records. This will update the internal BDI control tables.

A database commit is issued, and the control Id is returned by the API.

Data Definition XML

The BDI interface staging tables are generated based on the XML schema definition.

Data Flow	Description	XML Schema Definition (XSD)
Location Closed	Location Closed upload to BDI	LocationClosed_Fnd_BdiInterfaceModule.xml

Tables

TABLE	SELECT	INSERT	UPDATE	DELETE
LOCATION_CLOSED_OUT	No	Yes	No	No
LOCATION_CLOSED	Yes	No	No	No

Merch Hierarchy Publication API (BDI_MerchHier_Fnd_PF_From_RMS_JOB)

This section describes the Merch Hierarchy Publication BDI.

Functional Area

Foundation

Business Overview

BDI (Bulk Data Integration) is an integration layer that facilitates the bulk transfer of Merchandise Hierarchy information from Merchandising to other Oracle Retail Applications.

On this particular integration stream, the data flow is from Merchandising to BDI, and then BDI to downstream applications. To accomplish this data transfer, BDI will be calling a Merchandising-owned API that will pull data from Merchandising and deliver these to the BDI integration layer. This API will be in the form of a PLSQL function inside a PLSQL package.

Package Impact

This section describes the package impact.

Bulk Interface Module

Filename: bdimerchb.pls

```
BDI_MERCH_SQL.MERCH_HIER_UP(O_error_message IN OUT VARCHAR2,
                             O_control_id    IN OUT NUMBER,
                             I_job_context   IN     VARCHAR2)
```

This function begins by calling a BDI function that signals the start of the interface process. The BDI function will update the internal BDI control tables to track the progress of the API.

A DML insert statement is then executed to populate the BDI outbound table that resides in the BDI_RMS_INT_SCHEMA schema. This outbound table is loaded with records from the Merchandising Merchandise Hierarchy tables.

After the insert, another call to a BDI function is performed to signify the successful loading of records. This will update the internal BDI control tables.

A database commit is issued, and the control Id is returned by the API.

Data Definition XML

The BDI interface staging tables are generated based on the XML schema definition

Data Flow	Description	XML Schema Definition (XSD)
Merchandise Hierarchy	Merchandise Hierarchy upload to BDI	MerchHier_Fnd_BdiInterfaceModule.xml

Table Impact

TABLE	SELECT	INSERT	UPDATE	DELETE
MERCH_HIER_OUT	No	Yes	No	No
DIVISION	Yes	No	No	No
COMPHEAD	Yes	No	No	No
GROUPS	Yes	No	No	No
DEPS	Yes	No	No	No
CLASS	Yes	No	No	No
SUBCLASS	Yes	No	No	No

Merchandise Hierarchy Export (export_merchhier.ksh)

Module Name	export_merchhier.ksh
Description	Extraction of merchandise hierarchy data.
Functional Area	Foundation
Module Type	Integration
Module Technology	ksh
Catalog ID	260
Wrapper Script	rmswrap_shell_out.ksh

Schedule

Oracle Retail Merchandising Batch Schedule

Design Overview

This batch job will extract new, updated and deleted Merchandising merchandise hierarchy information from division to subclass into a flat file. Data to be extracted will be pulled off from the merchandise hierarchy export staging table and the main merchandise hierarchy tables. The mode (full vs. delta) will be an input parameter for this new batch. The mode will allow a full extract (all merchandise hierarchy records in Merchandising) as well as delta processing (all merchandise hierarchy changes since the last export) of data. For a full extract, records will be solely retrieved from the main merchandise hierarchy tables. For a delta extract, the action type and entity ID will be retrieved from the merchandise hierarchy export staging table and the attributes of the entities will be retrieved from their corresponding main entity tables.

Restart/Recovery

N/A

I/O Specification

Integration Type	Extract from Merchandising
File Name	merchierarchy_[Date]_[full/delta]_[#ofLines].dat
Integration Contract	IntCon000207

Design Assumptions

N/A

Organization Hierarchy Publication API (BDI_OrgHier_Fnd_PF_From_RMS_JOB)

This section describes Organization Hierarchy Publication BDI.

Functional Area

Foundation

Business Overview

BDI (Bulk Data Integration) is an integration layer that facilitates the bulk transfer of Org Hierarchy information from Merchandising to other Oracle Retail Applications. On this particular integration stream, the data flow is from Merchandising to BDI, and then BDI to downstream applications. To accomplish this data transfer, BDI will be calling a Merchandising-owned API that will pull data from Merchandising and deliver these to the BDI integration layer. This API will be in the form of a PLSQL function inside a PLSQL package.

Package Impact

This section describes the package impact.

Bulk Interface Module

Filename: bdiorgb.pls

```
BDI_ORG_SQL.ORG_HIER_UP(O_error_message IN OUT VARCHAR2,
                        O_control_id    IN OUT NUMBER,
                        I_job_context    IN     VARCHAR2)
```

This function begins by calling a BDI function that signals the start of the interface process. The BDI function will update the internal BDI control tables to track the progress of the API.

A DML insert statement is then executed to populate the BDI outbound table that resides in the BDI_RMS_INT_SCHEMA schema. This outbound table is loaded with records from the Merchandising Organization Hierarchy tables.

After the insert, another call to a BDI function is performed to signify the successful loading of records. This will update the internal BDI control tables.

A database commit is issued, and the control Id is returned by the API.

Data Definition XML

The BDI interface staging tables are generated based on the XML schema definition.

Data Flow	Description	XML Schema Definition (XSD)
Org Hierarchy	Org Hierarchy upload to BDI	OrgHier_Fnd_BdiInterfaceModule.xml

Table Impact

TABLE	SELECT	INSERT	UPDATE	DELETE
ORG_HIER_OUT	No	Yes	No	No
WH	Yes	No	No	No
AREA	Yes	No	No	No
CHAIN	Yes	No	No	No
COMPHEAD	Yes	No	No	No
DISTRICT	Yes	No	No	No
REGION	Yes	No	No	No
STORE	Yes	No	No	No
WH	Yes	No	No	No

Organizational Hierarchy Export (export_orghier.ksh)

Module Name	export_orghier.ksh
Description	Extraction of organizational hierarchy data.
Functional Area	Foundation
Module Type	Integration
Module Technology	Ksh
Catalog ID	RMS261

Wrapper Script rmswrap_shell_out.ksh

Schedule

Oracle Retail Merchandising Batch Schedule

Design Overview

This batch job will extract new, updated and deleted Merchandising organizational hierarchy information from company to stores and warehouses into a flat file. Data to be extracted will be pulled off from the organizational hierarchy export staging table and the main organizational hierarchy tables. The mode (full vs. delta) will be an input parameter for this batch. The mode will allow a full extract (all organizational hierarchy records in Merchandising) as well as delta processing (all organizational hierarchy changes since the last export) of data. For a full extract, records will be solely retrieved from the main organizational hierarchy tables. For a delta extract, the action type and entity ID will be retrieved from the organizational hierarchy export staging table and the attributes of the entities will be retrieved from their corresponding main entity tables.

Restart/Recovery

N/A

I/O Specification

Integration Type Extract from Merchandising
File Name orghierarchy_[Date]_[full/delta]_[#ofLines].dat
Integration Contract IntCon000203

Design Assumptions

N/A

Partner Address Publication API (BDI_PartnerAddr_Fnd_PF_From_RMS_JOB)

This section describes the Partner Address Publication BDI.

Functional Area

Foundation

Business Overview

BDI (Bulk Data Integration) is an integration layer that facilitates the bulk transfer of Code Head information from Merchandising to other Oracle Retail Applications. On this particular integration stream, the data flow is from Merchandising to BDI, and then BDI to downstream applications. To accomplish this data transfer, BDI will be calling a Merchandising-owned API that will pull data from Merchandising and deliver these to the BDI integration layer. This API will be in the form of a PLSQL function inside a PLSQL package.

Package Impact

Filename: **bdifoundationb.pls.pls**

```
BDI_FOUNDATION_SQL.PARTNER_ADDR_UP(
    O_error_message IN OUT RTK_ERRORS.RTK_TEXT%TYPE,
    O_control_id    IN OUT NUMBER,
    I_job_context   IN     VARCHAR2)
```

This function begins by calling a BDI function that signals the start of the interface process. The BDI function will update the internal BDI control tables to track the progress of the API.

A DML insert statement is then executed to populate the BDI outbound table that resides in the BDI_RMS_INT_SCHEMA schema. This outbound table is loaded with records from the Merchandising Partner Address table.

After the insert, another call to a BDI function is performed to signify the successful loading of records. This will update the internal BDI control tables.

A database commit is issued, and the control Id is returned by the API.

Data Definition XML

The BDI interface staging tables are generated based on the XML schema definition

Data Flow	Description	XML Schema Definition (XSD)
Partner Address	Partner Address upload to BDI	PartnerAddr_Fnd_BdiInterfaceModule.xml

Table Impact

TABLE	SELECT	INSERT	UPDATE	DELETE
PARTNER_ADDR_OUT	No	Yes	No	No
ADDR	Yes	No	No	No
V_ADD_TYPE_TL	Yes	No	No	No
ADD_TYPE_MODULE	Yes	No	No	No
STATE	Yes	No	No	No
COUNTRY	Yes	No	No	No
PARTNER	Yes	No	No	No

Partner Org Unit Publication API (BDI_PartOrgUnit_Fnd_PF_From_RMS_JOB)

This section describes the Partner Org Unit Publication BDI.

Functional Area

Foundation

Business Overview

BDI (Bulk Data Integration) is an integration layer that facilitates the bulk transfer of Code Head information from Merchandising to other Oracle Retail Applications. On this particular

integration stream, the data flow is from Merchandising to BDI, and then BDI to downstream applications. To accomplish this data transfer, BDI will be calling a Merchandising-owned API that will pull data from Merchandising and deliver these to the BDI integration layer. This API will be in the form of a PLSQL function inside a PLSQL package.

Package Impact

Filename: **bdifoundationb.pls.pls**

```
BDI_FOUNDATION_SQL.PARTNER_ORG_UNIT_UP(
    O_error_message IN OUT RTK_ERRORS.RTK_TEXT%TYPE,
    O_control_id    IN OUT NUMBER,
    I_job_context   IN     VARCHAR2)
```

This function begins by calling a BDI function that signals the start of the interface process. The BDI function will update the internal BDI control tables to track the progress of the API.

A DML insert statement is then executed to populate the BDI outbound table that resides in the BDI_RMS_INT_SCHEMA schema. This outbound table is loaded with records from the Merchandising Partner Org Unit table.

After the insert, another call to a BDI function is performed to signify the successful loading of records. This will update the internal BDI control tables.

A database commit is issued, and the control Id is returned by the API.

Data Definition XML

The BDI interface staging tables are generated based on the XML schema definition

Data Flow	Description	XML Schema Definition (XSD)
Partner Org Unit	Partner Org Unit upload to BDI	PartnerOrgUnit_Fnd_BdiInterfaceModule.xml

Table Impact

TABLE	SELECT	INSERT	UPDATE	DELETE
PARTNER_ORG_UNIT_OUT	No	Yes	No	No
PARTNER_ORG_UNIT	Yes	No	No	No

Partner Publication API (BDI_Partner_Fnd_PF_From_RMS_JOB)

This section describes the Partner Publication BDI.

Functional Area

Foundation

Business Overview

BDI (Bulk Data Integration) is an integration layer that facilitates the bulk transfer of Code Head information from Merchandising to other Oracle Retail Applications. On this particular integration stream, the data flow is from Merchandising to BDI, and then BDI to downstream applications. To accomplish this data transfer, BDI will be calling a Merchandising-owned API that will pull data from Merchandising and deliver these to the BDI integration layer. This API will be in the form of a PLSQL function inside a PLSQL package.

Package Impact

Filename: **bdifoundationb.pls.pls**

```
BDI_FOUNDATION_SQL.PARTNER_UP(O_error_message IN OUT RTK_ERRORS.RTK_TEXT%TYPE,
                               O_control_id    IN OUT NUMBER,
                               I_job_context   IN    VARCHAR2)
```

This function begins by calling a BDI function that signals the start of the interface process. The BDI function will update the internal BDI control tables to track the progress of the API. A DML insert statement is then executed to populate the BDI outbound table that resides in the BDI_RMS_INT_SCHEMA schema. This outbound table is loaded with records from the Merchandising Partner table. After the insert, another call to a BDI function is performed to signify the successful loading of records. This will update the internal BDI control tables. A database commit is issued, and the control Id is returned by the API.

Data Definition XML

The BDI interface staging tables are generated based on the XML schema definition

Data Flow	Description	XML Schema Definition (XSD)
Partner	Partner upload to BDI	Partner_Fnd_BdiInterfaceModule.xml

Table Impact

TABLE	SELECT	INSERT	UPDATE	DELETE
PARTNER_OUT	No	Yes	No	No
PARTNER	Yes	No	No	No

Store Address Publication API (BDI_StoreAddr_Fnd_PF_From_RMS_JOB)

This section describes the Store Address Publication BDI.

Functional Area

Foundation

Business Overview

BDI (Bulk Data Integration) is an integration layer that facilitates the bulk transfer of Store Address information from Merchandising to other Oracle Retail Applications. On this particular integration stream, the data flow is from Merchandising to BDI, and then BDI to

downstream applications. To accomplish this data transfer, BDI will be calling a Merchandising-owned API that will pull data from Merchandising and deliver these to the BDI integration layer. This API will be in the form of a PLSQL function inside a PLSQL package.

Package Impact

This section describes the package impact.

Bulk Interface Module

Filename: **bdiorgb.pls**

```
BDI_ORG_SQL.STORE_ADDR_UP(O_error_message IN OUT VARCHAR2,
                          O_control_id    IN OUT NUMBER,
                          I_job_context   IN     VARCHAR2)
```

This function begins by calling a BDI function that signals the start of the interface process. The BDI function will update the internal BDI control tables to track the progress of the API.

A DML insert statement is then executed to populate the BDI outbound table that resides in the BDI_RMS_INT_SCHEMA schema. This outbound table is loaded with records from the Merchandising Store Address table.

After the insert, another call to a BDI function is performed to signify the successful loading of records. This will update the internal BDI control tables.

A database commit is issued, and the control Id is returned by the API.

Data Definition XML

The BDI interface staging tables are generated based on the XML schema definition

Data Flow	Description	XML Schema Definition (XSD)
Store Addr	Store Address upload to BDI	StoreAddr_Fnd_BdiInterfaceModule.xml

Table Impact

TABLE	SELECT	INSERT	UPDATE	DELETE
STORE_ADDR_OUT	No	Yes	No	No
V_ADD_TYPE_TL	Yes	No	No	No
ADDR	Yes	No	No	No
STORE	Yes	No	No	No
STATE	Yes	No	No	No
COUNTRY	Yes	No	No	No
ADD_TYPE_MODULE	Yes	No	No	No

Store Hours Publication API (BDI_StoreHours_Fnd_PF_From_RMS_JOB)

This section describe the Store Hours Publication BDI.

Function Area

Foundation

Design Overview

BDI (Bulk Data Integration) is an integration layer that facilitates the bulk transfer of Store information from Merchandising to other Oracle Retail Applications. On this particular integration stream, the data flow is from Merchandising to BDI, and then BDI to downstream applications. To accomplish this data transfer, BDI will be calling a Merchandising-owned API that will pull data from Merchandising and deliver these to the BDI integration layer. This API will be in the form of a PLSQL function inside a PLSQL package.

Package Impact

The following packages are impacted by the Store Hours Publication BDI:

Bulk Interface Module

In the Build Interface Module:

Filename: bdiorgb.pls

```
BDI_ORG_SQL.STORE_HOURS_UP(O_error_message IN OUT VARCHAR2,
                           O_control_id    IN OUT NUMBER,
                           I_job_context   IN      VARCHAR2)
```

This function begins by calling a BDI function that signals the start of the interface process. The BDI function will update the internal BDI control tables to track the progress of the API.

A DML insert statement is then executed to populate the BDI outbound table that resides in the BDI_RMS_INT_SCHEMA schema. This outbound table is loaded with records from the Merchandising Store table.

After the insert, another call to a BDI function is performed to signify the successful loading of records. This will update the internal BDI control tables.

A database commit is issued, and the control Id is returned by the API.

Data Definition XML

The BDI interface staging tables are generated based on the XML schema definition.

Data Flow	Description	XML Schema Definition (XSD)
Store	Store upload to BDI	StoreHours_Fnd_BdiInterfaceModule.xml

Tables

TABLE	SELECT	INSERT	UPDATE	DELETE
STORE_HOURS_OUT	No	Yes	No	No
STORE_HOURS	Yes	No	No	No

Store Publication API (BDI_Store_Fnd_PF_From_RMS_JOB)

This section describes the Store Publication BDI.

Functional Area

Foundation

Business Overview

BDI (Bulk Data Integration) is an integration layer that facilitates the bulk transfer of Store information from Merchandising to other Oracle Retail Applications. On this particular integration stream, the data flow is from Merchandising to BDI, and then BDI to downstream applications. To accomplish this data transfer, BDI will be calling a Merchandising-owned API that will pull data from Merchandising and deliver these to the BDI integration layer. This API will be in the form of a PLSQL function inside a PLSQL package.

Package Impact

This section describes the package impact.

Bulk Interface Module

Filename: bdiorgb.pls

```
BDI_ORG_SQL.STORE_UP(O_error_message IN OUT VARCHAR2,
                    O_control_id     IN OUT NUMBER,
                    I_job_context    IN     VARCHAR2)
```

This function begins by calling a BDI function that signals the start of the interface process. The BDI function will update the internal BDI control tables to track the progress of the API.

A DML insert statement is then executed to populate the BDI outbound table that resides in the BDI_RMS_INT_SCHEMA schema. This outbound table is loaded with records from the Merchandising Item Location table.

After the insert, another call to a BDI function is performed to signify the successful loading of records. This will update the internal BDI control tables.

A database commit is issued, and the control Id is returned by the API.

Data Definition XML

The BDI interface staging tables are generated based on the XML schema definition.

Data Flow	Description	XML Schema Definition (XSD)
Store	Store upload to BDI	Store_Fnd_BdiInterfaceModule.xml

Table Impact

TABLE	SELECT	INSERT	UPDATE	DELETE
STORE_OUT	No	Yes	No	No
CODE_DETAIL	Yes	No	No	No
CHANNELS	Yes	No	No	No
STORE_FORMAT	Yes	No	No	No
LANG	Yes	No	No	No
VAT_REGION	Yes	No	No	No
TSFZONE	Yes	No	No	No

Stores Export (export_stores.ksh)

Module Name	export_stores.ksh
Description	Extraction of store data.
Functional Area	Foundation
Module Type	Integration
Module Technology	Ksh
Catalog ID	RMS263
Wrapper Script	rmswrap_shell_out.ksh

Schedule

Oracle Retail Merchandising Batch Schedule

Design Overview

This batch job will extract new, updated and deleted Merchandising store information into two flat files - one for store and one for store addresses. Data to be extracted will be pulled from the store export staging, store and address tables. The mode (full vs. delta) will be an input parameter for this batch. The mode will allow a full extract (all store records in Merchandising) as well as delta processing (all store changes since the last export) of data. For a full extract, records will be solely retrieved from the store table for store information and address table for store addresses. For a delta extract, the action type, store ID and address will be retrieved from the store export staging table and the details of the store will be retrieved from both the store and address tables.

Restart/Recovery

N/A

I/O Specification

Integration Type	Extract from Merchandising
File Name	store_[Date]_[full/delta]_[#ofLines].datstoreaddr_[Date]_[full/delta]_[#ofLines].dat

Integration Contract IntCon000204
IntCon000205

Design Assumptions

N/A

Supplier Address Publication API (BDI_SupplierAddr_Fnd_PF_From_RMS_JOB)

This section describes the Supplier Address Publication BDI.

Functional Area

Foundation

Business Overview

BDI (Bulk Data Integration) is an integration layer that facilitates the bulk transfer of Supplier Address positions from Merchandising to other Oracle Retail Applications. On this particular integration stream, the data flow is from Merchandising to BDI, and then BDI to downstream applications. To accomplish this data transfer, BDI will be calling a Merchandising-owned API that will pull data from Merchandising and deliver these to the BDI integration layer. This API will be in the form of a PLSQL function inside a PLSQL package.

Package Impact

Filename: **bdifoundations/b.pls**

```
BDI_FOUNDATION_SQL.SUPPLIER_ADDR_UP(O_error_message IN OUT VARCHAR2,  
                                     O_control_id     IN OUT NUMBER,  
                                     I_job_context    IN     VARCHAR2)
```

This function begins by calling a BDI function that signals the start of the interface process. The BDI function will update the internal BDI control tables to track the progress of the API.

A DML insert statement is then executed to populate the BDI outbound table that resides in the BDI_RMS_INT_SCHEMA schema. This outbound table is loaded with records from the Merchandising Supplier Address tables.

After the insert, another call to a BDI function is performed to signify the successful loading of records. This will update the internal BDI control tables.

A database commit is issued, and the control Id is returned by the API.

Data Definition XML

The BDI interface staging tables are generated based on the XML schema definition

Data Flow	Description	XML Schema Definition (XSD)
Supplier Address	Supplier Address upload to BDI	SupplierAddr_Fnd_BdiInterfaceModule.xml

Table Impact

TABLE	SELECT	INSERT	UPDATE	DELETE
SUPPLIER_ADDR_OUT	No	Yes	No	No
ADDR	Yes	No	No	No
V_ADD_TYPE_TL	Yes	No	No	No
STATE	Yes	No	No	No
COUNTRY	Yes	No	No	No
SUPS	Yes	No	No	No
ADD_TYPE_MODULE	Yes	No	No	No

Sups Publication API (BDI_Supplier_Fnd_PF_From_RMS_JOB)

This section describes the Sups Publication BDI.

Functional Area

Foundation

Business Overview

BDI (Bulk Data Integration) is an integration layer that facilitates the bulk transfer of Code Head information from Merchandising to other Oracle Retail Applications. On this particular integration stream, the data flow is from Merchandising to BDI, and then BDI to downstream applications. To accomplish this data transfer, BDI will be calling a Merchandising-owned API that will pull data from Merchandising and deliver these to the BDI integration layer. This API will be in the form of a PLSQL function inside a PLSQL package.

Package Impact

Filename: **bdifoundationb.pls.pls**

```
BDI_FOUNDATION_SQL.SUPS_UP(O_error_message IN OUT RTK_ERRORS.RTK_TEXT%TYPE,
                           O_control_id    IN OUT NUMBER,
                           I_job_context   IN    VARCHAR2)
```

This function begins by calling a BDI function that signals the start of the interface process. The BDI function will update the internal BDI control tables to track the progress of the API.

A DML insert statement is then executed to populate the BDI outbound table that resides in the BDI_RMS_INT_SCHEMA schema. This outbound table is loaded with records from the Merchandising Sups table.

After the insert, another call to a BDI function is performed to signify the successful loading of records. This will update the internal BDI control tables.

A database commit is issued, and the control Id is returned by the API.

Data Definition XML

The BDI interface staging tables are generated based on the XML schema definition

Data Flow	Description	XML Schema Definition (XSD)
Supplier	Supplier upload to BDI	Supplier_Fnd_BdiInterfaceModule.xml

Table Impact

TABLE	SELECT	INSERT	UPDATE	DELETE
SUPS_OUT	No	Yes	No	No
SUPS	Yes	No	No	No

Tax Download - Brazil (taxdnlD)

Module Name	taxdnlD
Description	Tax Download to 3rd Party POS in Global Tax [GTAX] Implementations
Functional Area	Integration - 3rd Party POS
Module Type	Integration
Module Technology	ProC
Catalog ID	RMS124
Wrapper Script	N/A

Schedule

Oracle Retail Merchandising Batch Schedule

Design Overview

This batch program downloads the tax information to 3rd Party POS systems when the Merchandising default tax type is GTAX. This program only needs to be run if the client uses Merchandising Global Tax functionality.

Restart/Recovery

The logical unit of work for this module is defined by item, ref_item and store combination. This batch program uses table-based restart/recovery. The commit happens in the database when the commit max counter is reached.

Integration Contract

Integration Type	Download from Merchandising
File Name	Determined by runtime parameter
Integration Contract	IntCon000020

Output File Layout

Table 6-3 Output File Layout

Record Name	Field Name	Field Type	Default Value	Description
FHEAD	File Type Record Descriptor	Char(5)	FHEAD	Identifies file record type
	File Line Sequence	Number(10)	N/A	Line number of the current file
	File Type Definition	Char(4)	TAXD	Identifies file as 'Tax Details'
	File Create Date	Char(14)	create date	Vdate in 'YYYYMMDDHHM ISS'format
FDETL	FDETL	Char(5)	FDETL	FDETL
	File Line Sequence	Number(10)	N/A	Line number of the current file
	STORE	Char(10)	N/A	Store number
	ITEM	Char(25)	N/A	Item
	item_number_type	Char(6)	S - Store W - Warehouse	Item number type
	format_id	Char(1)	N/A	Format id
	prefix	Char(2)	N/A	Prefix
	ref_item	Char(25)	N/A	Reference Item
	ref_item_number_type	Char(6)	N/A	Reference item number type
	ref_format_id	Char(1)	N/A	Ref format id
	ref_prefix	Char(2)	N/A	Ref no. prefix
	taxable indicator	Char(1)	N/A	Taxable indicator
	class_vat_ind	Char(1)	N/A	Class vat indicator
FTAXD	FTAXD	Char(5)	FTAXD	FTAXD
	File Line Sequence	Number(10)	N/A	Line number of the current file
	tax_code	Char(10)	N/A	Tax code
	tax_rate	Char(20)	N/A	Tax rate
	calculation_basis	Char(1)	N/A	Calculation basis
	tax_amount	Char(20)	N/A	Tax amount
	effective_from	Char(8)	N/A	Effective from
	time	Char(6)	N/A	Time
	status	Char(1)	N/A	Status

Table 6-3 (Cont.) Output File Layout

Record Name	Field Name	Field Type	Default Value	Description
FTAIL	File Type Record Descriptor	Char(5)	FTAIL	Identifies file record type
	File Line Sequence	Number(10)	N/A	Line number of the current file
	rec_counter	Number(10)	N/A	Record counter

Design Assumptions

N/A

Ticket Download (tcktdnld)

Module Name	tcktdnld.pc
Description	Download of Data to be Printed on Tickets
Functional Area	Foundation Data
Module Type	Integration
Module Technology	PROC
Catalog ID	RMS59
Wrapper Script	rmswrap_out.ksh

Schedule

Oracle Retail Merchandising Batch Schedule

Design Overview

This program creates an output file containing the information to be printed on a ticket or label for a particular item/location. This program is driven by the requests for tickets generated from Merchandising and Pricing. The details of what should be printed on each ticket are defined in Merchandising on the ticket type details table.

Restart/Recovery

N/A

I/O Specification

Integration Type	Download from Merchandising
File Name	Determined by runtime parameters
Integration Contract	IntCon000107

Output File Layout

Table 6-4 tcktdnld.pc - Output File Layout

Record Name	Field Name	Field Type	Default Value	Description
FHEAD	File Type Record Descriptor	Char(5)	FHEAD	Identifies file record type
	File Line Sequence	Number(10)	N/A	Line number of the current file
	File Type Definition	Char(4)	TCKT	Identifies file as 'Print Ticket Requests'
	File Create Date	Char(14)	N/A	The date on which the file was created in 'YYYYMMDDHHMISS' format
THEAD	File Type Record Descriptor	Char(5)	THEAD	Identifies file record type
	File Line Sequence	Number(10)	N/A	Line number of the current file
	ITEM	Char(25)	N/A	ID number of the transaction level item for which the ticket applies.
	Ticket Type	Char(4)	N/A	ID which indicates the ticket type to be printed
	Location Type	Char(1)	N/A	Identifies the type of location for which tickets will be printed. Valid values are store (S) and warehouse (W).
	Location	Char(10)	N/A	The ID of the store or warehouse for which tickets will be printed
	Quantity	Number(12,4)	N/A	The quantity of tickets to be printed; which includes 4 implied decimal places
TCOMP	File Type Record Descriptor	Char(5)	TCOMP	Identifies file record type
	File Line Sequence	Number(10)	N/A	Line number of the current file
	ITEM	Char(25)	N/A	ID number of the item which is only populated if the item in THEAD is a pack item
	Quantity	Number(12,4)	N/A	Quantity of the component item as a part of the pack; includes 4 implied decimal places
TDETL	File Type Record Descriptor	Char(5)	TDETL	Identifies file record type
	File Line Sequence	Number(10)	N/A	Line number of the current file

Table 6-4 (Cont.) tcktdnld.pc - Output File Layout

Record Name	Field Name	Field Type	Default Value	Description
	Detail Sequence Number	Number(10)	N/A	Sequential number assigned to the detail records
	Ticket Item	Char(4)	N/A	ID indicating the detail to be printed on the ticket. If the attribute is a UDA, then this will contain the ID of the UDA. Otherwise, it is the code associated with the attribute in Merchandising (such as, CLSS = class)
	Attribute Description	Char(120)	N/A	Description of the attribute – either the UDA description or the Merchandising description for the attribute
	Value	Char(250)	N/A	Detail to be printed on the ticket (for example: Item number, Department Number, Item description)
	Supplement	Char(120)	N/A	Supplemental description to the Value (for example: Department Name)
TTAIL	File Type Record Descriptor	Char(5)	TTAIL	Identifies file record type
	File Line Sequence	Number(10)	N/A	Line number of the current file
	Transaction Detail Line Count	Number(6)	sum of detail lines	Sum of the detail lines within a transaction
FTAIL	File Type Record Descriptor	Char(5)	FTAIL	Identifies file record type
	File Line Sequence	Number(10)	N/A	Line number of the current file

Design Assumptions

N/A

UDA Publication API (BDI_Uda_Fnd_PF_From_RMS_JOB)

This section describes the UDA Publication BDI.

Functional Area

Foundation

Business Overview

BDI (Bulk Data Integration) is an integration layer that facilitates the bulk transfer of Code Head information from Merchandising to other Oracle Retail Applications. On this particular integration stream, the data flow is from Merchandising to BDI, and then BDI to downstream applications. To accomplish this data transfer, BDI will be calling a Merchandising-owned API that will pull data from Merchandising and deliver these to the BDI integration layer. This API will be in the form of a PLSQL function inside a PLSQL package.

Package Impact

Filename: **bdifoundationb.pls**

```
BDI_FOUNDATION_SQL.UDA_UP(O_error_message IN OUT RTK_ERRORS.RTK_TEXT%TYPE,
                           O_control_id    IN OUT NUMBER,
                           I_job_context   IN    VARCHAR2)
```

This function begins by calling a BDI function that signals the start of the interface process. The BDI function will update the internal BDI control tables to track the progress of the API.

A DML insert statement is then executed to populate the BDI outbound table that resides in the BDI_RMS_INT_SCHEMA schema. This outbound table is loaded with records from the Merchandising UDA table.

After the insert, another call to a BDI function is performed to signify the successful loading of records. This will update the internal BDI control tables.

A database commit is issued, and the control Id is returned by the API.

Data Definition XML

The BDI interface staging tables are generated based on the XML schema definition

Data Flow	Description	XML Schema Definition (XSD)
UDA	UDA upload to BDI	Uda_Fnd_BdiInterfaceModule.xml

Table Impact

TABLE	SELECT	INSERT	UPDATE	DELETE
UDA_OUT	No	Yes	No	No
UDA	Yes	No	No	No

UDA Values Publication API (BDI_UdaValues_Fnd_PF_From_RMS_JOB)

This section describes the UDA Values Publication BDI.

Functional Area

Foundation

Business Overview

BDI (Bulk Data Integration) is an integration layer that facilitates the bulk transfer of Code Head information from Merchandising to other Oracle Retail Applications. On this particular integration stream, the data flow is from Merchandising to BDI, and then BDI to downstream applications. To accomplish this data transfer, BDI will be calling a Merchandising-owned API that will pull data from Merchandising and deliver these to the BDI integration layer. This API will be in the form of a PLSQL function inside a PLSQL package.

Package Impact

Filename: `bdifoundationb.pls.pls`

```
BDI_FOUNDATION_SQL.UDA_VALUES_UP(O_error_message IN OUT RTK_ERRORS.RTK_TEXT%TYPE,
                                O_control_id     IN OUT NUMBER,
                                I_job_context    IN     VARCHAR2)
```

This function begins by calling a BDI function that signals the start of the interface process. The BDI function will update the internal BDI control tables to track the progress of the API.

A DML insert statement is then executed to populate the BDI outbound table that resides in the BDI_RMS_INT_SCHEMA schema. This outbound table is loaded with records from the Merchandising UDA Values table.

After the insert, another call to a BDI function is performed to signify the successful loading of records. This will update the internal BDI control tables.

A database commit is issued, and the control Id is returned by the API.

Data Definition XML

The BDI interface staging tables are generated based on the XML schema definition

Data Flow	Description	XML Schema Definition (XSD)
UDA Values	UDA Values upload to BDI	UdaValues_Fnd_BdiInterfaceModule.xml

Table Impact

TABLE	SELECT	INSERT	UPDATE	DELETE
UDA_VALUES_OUT	No	Yes	No	No
UDA_VALUES	Yes	No	No	No

UOM Class Publication API (BDI_UomClass_Fnd_PF_From_RMS_JOB)

This section describes the UOM Class Publication BDI.

Functional Area

Cross Pillar

Business Overview

BDI (Bulk Data Integration) is an integration layer that facilitates the bulk transfer of Uom Class information from Merchandising to other Oracle Retail Applications. On this particular integration stream, the data flow is from Merchandising to BDI, and then BDI to downstream applications. To accomplish this data transfer, BDI will be calling a Merchandising-owned API that will pull data from Merchandising and deliver these to the BDI integration layer. This API will be in the form of a PLSQL function inside a PLSQL package.

Package Impact

Filename: `bdicrosspillarb.pls`

```
BDI_CROSS_PILLAR_SQL.UOM_CLASS_UP (
    O_error_message IN OUT RTK_ERRORS.RTK_TEXT%TYPE,
    O_control_id    IN OUT NUMBER,
    I_job_context   IN     VARCHAR2)
```

This function begins by calling a BDI function that signals the start of the interface process. The BDI function will update the internal BDI control tables to track the progress of the API.

A DML insert statement is then executed to populate the BDI outbound table that resides in the `BDI_RMS_INT_SCHEMA` schema. This outbound table is loaded with records from the Merchandising `UOM_CLASS` table.

After the insert, another call to a BDI function is performed to signify the successful loading of records. This will update the internal BDI control tables.

A database commit is issued, and the control Id is returned by the API.

Data Definition XML

The BDI interface staging tables are generated based on the XML schema definition

Data Flow	Description	XML Schema Definition (XSD)
Uom Class	Uom Class upload to BDI	UomClass_Fnd_BdiInterfaceModule.xml

Table Impact

TABLE	SELECT	INSERT	UPDATE	DELETE
UOM_CLASS_OUT	No	Yes	No	No
UOM_CLASS	Yes	No	No	No

UOM Conversion Publication API (BDI_UomConversion_Fnd_PF_From_RMS_JOB)

This section describes the UOM Conversion BDI.

Functional Area

Cross Pillar

Business Overview

BDI (Bulk Data Integration) is an integration layer that facilitates the bulk transfer of Uom Conversion information from Merchandising to other Oracle Retail Applications. On this particular integration stream, the data flow is from Merchandising to BDI, and then BDI to downstream applications. To accomplish this data transfer, BDI will be calling a Merchandising-owned API that will pull data from Merchandising and deliver these to the BDI integration layer. This API will be in the form of a PLSQL function inside a PLSQL package.

Package Impact

Filename: **bdicrosspillarb.pls**

```
BDI_CROSS_PILLAR_SQL.UOM_CONVERSION_UP (
    O_error_message IN OUT RTK_ERRORS.RTK_TEXT%TYPE,
    O_control_id    IN OUT NUMBER,
    I_job_context   IN     VARCHAR2)
```

This function begins by calling a BDI function that signals the start of the interface process. The BDI function will update the internal BDI control tables to track the progress of the API.

A DML insert statement is then executed to populate the BDI outbound table that resides in the BDI_RMS_INT_SCHEMA schema. This outbound table is loaded with records from the Merchandising UOM_CONVERSION table.

After the insert, another call to a BDI function is performed to signify the successful loading of records. This will update the internal BDI control tables.

A database commit is issued, and the control Id is returned by the API.

Data Definition XML

The BDI interface staging tables are generated based on the XML schema definition

Data Flow	Description	XML Schema Definition (XSD)
Uom Conversion	Uom Conversion upload to BDI	UomConversion_Fnd_BdiInterfaceModule.xml

Table Impact

TABLE	SELECT	INSERT	UPDATE	DELETE
UOM_CONVERSION_OUT	No	Yes	No	No
UOM_CONVERSION	Yes	No	No	No

VAT Codes, Regions, and Rates Export (export_vat.ksh)

Module Name	export_vat.ksh
Description	Extraction of vat data
Functional Area	Foundation

Module Type	Integration
Module Technology	Ksh
Catalog ID	RMS264
Wrapper Script	rmswrap_shell_out.ksh

Schedule

Oracle Retail Merchandising Batch Schedule

Design Overview

This batch job will extract new, updated and deleted Merchandising VAT information into a flat file. Data to be extracted will be pulled off from the VAT export staging, VAT region, VAT codes, and VAT code rates tables.

The mode (full vs. delta) will be an input parameter for this batch. The mode will allow a full extract (all vat region/vat code/vat code rate combination records in RMS) as well as delta processing (all VAT record changes in the time frame passed in the program) of data.

In either of the mode exempt vat region won't get fetched in case of SVAT tax type.

For a full extract, records will be retrieved from the VAT region, VAT code, and VAT code rates tables. For a delta extract, the action type, vat region, vat code and active date will be retrieved from the VAT export staging table and the attributes will be retrieved from the main table.

Restart/Recovery

N/A

I/O Specification

Integration Type	Extract from Merchandising
File Name	vat_date_[full/delta]_[#ofLines].dat
Integration Contract	IntCon000215

Design Assumptions

N/A

VAT Publication API (BDI_Vat_Fnd_PF_From_RMS_JOB)

This section describes the VAT Publication BDI.

Functional Area

Foundation

Design Overview

BDI (Bulk Data Integration) is an integration layer that facilitates the bulk transfer of Vat information from RMS to other Oracle Retail Applications. On this particular integration stream, the data flow is from RMS to BDI, and then BDI to downstream applications. To accomplish this data transfer, BDI will be calling an RMS owned API that will pull data from RMS and deliver these to the BDI integration layer. This API will be in the form of a PLSQL function inside a PLSQL package.

Package Impact

Bulk Interface Module

Filename: bdifoundationb.pls

```
BDI_FOUNDATION_SQL.FUNCTION VAT_UP
    (O_error_message IN OUT RTK_ERRORS.RTK_TEXT%TYPE,
    O_control_id     IN OUT  NUMBER,
    I_job_context    IN     VARCHAR2)
```

This function begins by calling a BDI function that signals the start of the interface process. The BDI function will update the internal BDI control tables to track the progress of the API.

A DML insert statement is then executed to populate the BDI outbound table that resides in the BDI_RMS_INT_SCHEMA schema. This outbound table is loaded with records from the VAT_CODES, VAT_CODE_RATES and VAT_REGION tables from RMS.

After the insert, another call to a BDI function is performed to signify the successful loading of records. This will update the internal BDI control tables.

A database commit is issued, and the control Id is returned by the API.

Data Definition XML

The BDI interface staging tables are generated based on the XML schema definition.

Data Flow	Description	XML Schema Definition (XSD)
Vat	Vat upload to BDI	Vat_Fnd_BdiInterfaceModule.xml

Table Impact

TABLE	SELECT	INSERT	UPDATE	DELETE
VAT_OUT	No	Yes	No	No
VAT_CODES	Yes	No	No	No
VAT_CODE_RATES	Yes	No	No	No
VAT_REGION	Yes	No	No	No

Warehouse (BDI_Wh_Fnd_PF_From_RMS_JOB)

This section describes Warehouse Publication BDI.

Functional Area

Foundation

Business Overview

BDI (Bulk Data Integration) is an integration layer that facilitates the bulk transfer of Warehouse information from Merchandising to other Oracle Retail Applications. On this particular integration stream, the data flow is from Merchandising to BDI, and then BDI to downstream applications. To accomplish this data transfer, BDI will be calling a Merchandising-owned API that will pull data from Merchandising and deliver these to the BDI integration layer. This API will be in the form of a PLSQL function inside a PLSQL package.

Package Impact

This section describes the package impact.

Bulk Interface Module

Filename: bdiorgb.pls

```
BDI_ORG_SQL.WH_UP(O_error_message IN OUT VARCHAR2,
                  O_control_id     IN OUT NUMBER,
                  I_job_context     IN     VARCHAR2)
```

This function begins by calling a BDI function that signals the start of the interface process. The BDI function will update the internal BDI control tables to track the progress of the API.

A DML insert statement is then executed to populate the BDI outbound table that resides in the BDI_RMS_INT_SCHEMA schema. This outbound table is loaded with records from the Merchandising Warehouse tables.

After the insert, another call to a BDI function is performed to signify the successful loading of records. This will update the internal BDI control tables.

A database commit is issued, and the control Id is returned by the API.

Data Definition XML

The BDI interface staging tables are generated based on the XML schema definition.

Data Flow	Description	XML Schema Definition (XSD)
Warehouse	Warehouse upload to BDI	Wh_Fnd_BdiInterfaceModule.xml

Table Impact

TABLE	SELECT	INSERT	UPDATE	DELETE
WH_OUT	No	Yes	No	No

TABLE	SELECT	INSERT	UPDATE	DELETE
WH	Yes	No	No	No

Warehouse Address Publication API (BDI_WhAddr_Fnd_PF_From_RMS_JOB)

This section describes Warehouse Address Publication BDI.

Functional Area

Foundation

Business Overview

BDI (Bulk Data Integration) is an integration layer that facilitates the bulk transfer of Warehouse Address information from Merchandising to other Oracle Retail Applications. On this particular integration stream, the data flow is from Merchandising to BDI, and then BDI to downstream applications. To accomplish this data transfer, BDI will be calling a Merchandising-owned API that will pull data from Merchandising and deliver these to the BDI integration layer. This API will be in the form of a PLSQL function inside a PLSQL package.

Package Impact

This section describes the package impact.

Bulk Interface Module

Filename: **bdiorgb.pls**

```
BDI_ORG_SQL.WH_ADDR_UP(O_error_message IN OUT VARCHAR2,
                       O_control_id    IN OUT NUMBER,
                       I_job_context   IN     VARCHAR2)
```

This function begins by calling a BDI function that signals the start of the interface process. The BDI function will update the internal BDI control tables to track the progress of the API.

A DML insert statement is then executed to populate the BDI outbound table that resides in the BDI_RMS_INT_SCHEMA schema. This outbound table is loaded with records from the Merchandising Warehouse Address tables.

After the insert, another call to a BDI function is performed to signify the successful loading of records. This will update the internal BDI control tables.

A database commit is issued, and the control Id is returned by the API.

Data Definition XML

The BDI interface staging tables are generated based on the XML schema definition.

Data Flow	Description	XML Schema Definition (XSD)
Warehouse Address	Warehouse Address upload to BDI	WhAddr_Fnd_BdiInterfaceModule.xml

Table Impact

TABLE	SELECT	INSERT	UPDATE	DELETE
WH_ADDR_OUT	No	Yes	No	No
ADDR	Yes	No	No	No

Items

Merchandising publishes item data for many other solution areas, including stores, warehouses, omni-channel, and so on.

- [Item Image Publication API \(BDI_ItemImage_Fnd_PF_From_RMS_JOB\)](#)
- [Item Location Export \(export_itemloc.ksh\)](#)
- [Item Location Publication API \(BDI_ItemLoc_Fnd_PF_From_RMS_JOB\)](#)
- [Item Master Export \(export_itemmaster.ksh\)](#)
- [Item Master Publication API \(BDI_ItemHdr_Fnd_PF_From_RMS_JOB\)](#)
- [Item Supplier Country Dimensions Publication API \(BDI_ItSupCtryDim_Fnd_PF_From_RMS_JOB\)](#)
- [Item Supplier Country Publication API \(BDI_ItSupCtry_Fnd_PF_From_RMS_JOB\)](#)
- [Item Supplier Manufacturing Country Publication API \(BDI_ItSupManCtry_Fnd_PF_From_RMS_JOB\)](#)
- [Item Supplier Publication API \(BDI_ItemSupp_Fnd_PF_From_RMS_JOB\)](#)
- [Item Supplier UOM Publication API \(BDI_ItemSuppUom_Fnd_PF_From_RMS_JOB\)](#)
- [Item VAT Rates Export \(export_itemvat.ksh\)](#)
- [Pack Item Publication API \(BDI_PckitemBrkout_Fnd_PF_From_RMS_JOB\)](#)
- [POS Configuration Data to 3rd Party POS \(poscdnld\)](#)
- [Price History Publication API \(BDI_PriceHist_Fnd_PF_From_RMS_JOB\)](#)
- [Related Items Export \(export_relitem.ksh\)](#)
- [Related Item Publication API \(BDI_RelatedItem_Fnd_PF_From_RMS_JOB\)](#)
- [UDA Item Date Publication API \(BDI_UdaltemDate_Fnd_PF_From_RMS_JOB\)](#)
- [UDA Item FF Publication API \(BDI_UdaltemFF_Fnd_PF_From_RMS_JOB\)](#)
- [UDA Item LOV Publication API \(BDI_UdaltemLov_Fnd_From_RMS_JOB\)](#)

Item Image Publication API (BDI_ItemImage_Fnd_PF_From_RMS_JOB)

This section describes the Item Image Publication BDI.

Functional Area

Item

Business Overview

BDI (Bulk Data Integration) is an integration layer that facilitates the bulk transfer of Item Image information from Merchandising to other Oracle Retail Applications. On this particular integration stream, the data flow is from Merchandising to BDI, and then BDI to downstream applications. To accomplish this data transfer, BDI will be calling a Merchandising-owned API that will pull data from Merchandising and deliver these to the BDI integration layer. This API will be in the form of a PLSQL function inside a PLSQL package.

Package Impact

Filename: bdiitemb.pls

```
BDI_ITEM_SQL.ITEM_IMAGE_UP (O_error_message IN OUT VARCHAR2,
                             O_control_id    IN OUT NUMBER,
                             I_job_context   IN     VARCHAR2)
```

This function begins by calling a BDI function that signals the start of the interface process. The BDI function will update the internal BDI control tables to track the progress of the API.

A DML insert statement is then executed to populate the BDI outbound table that resides in the BDI_RMS_INT_SCHEMA schema. This outbound table is loaded with records from the Merchandising ITEM_IMAGE table.

After the insert, another call to a BDI function is performed to signify the successful loading of records. This will update the internal BDI control tables.

A database commit is issued, and the control Id is returned by the API.

Data Definition XML

The BDI interface staging tables are generated based on the XML schema definition

Data Flow	Description	XML Schema Definition (XSD)
Item Image	Item Image upload to BDI	ItemImage_Fnd_BdiInterfaceModule.xml

Table Impact

TABLE	SELECT	INSERT	UPDATE	DELETE
ITEM_IMAGE_OUT	No	Yes	No	No
ITEM_MASTER	Yes	No	No	No
ITEM_IMAGE	Yes	No	No	No

Item Location Export (export_itemloc.ksh)

Module Name	export_itemloc.ksh
Description	Extraction of item location data.
Functional Area	Foundation

Module Type	Integration
Module Technology	ksh
Catalog ID	RMS257
Wrapper Script	rmswrap_shell_out.ksh

Schedule

Oracle Retail Merchandising Batch Schedule

Design Overview

This batch job extracts new, updated and deleted Merchandising item-location information into a flat file.

- This batch supports both a full and delta export of item-location data.
- A threading indicator parameter should be passed. Passing 'Y' means a thread number (1-20) will be passed in. Passing 'N' means no thread number will be passed in and the program will use a default thread number.
- An optional location parameter may be passed in for either modes. If this value is passed in, the batch will create a flat file for the location passed in. If it is not passed in, the batch will create flat files for all locations.
- This creates separate files per location (Store, Warehouse or External Finisher).
- This exports delta item header information for each applicable store location.
- This will export data only for approved, sellable items.
- This will export attributes from item/location and item/location traits.
- This should also include the item parent as its own record in the extract.
- The flat files that will be created will now be pipe delimited.

Restart/Recovery

N/A

I/O Specification

Integration Type	Extract from Merchandising
File Name	itemloc_[#date]_[#loc_type]_[#location]_[full/delta]_[#ofLines].dat itemhdr_[#date]_[#store_id]_delta_[#ofLines].dat
Integration Contract	IntCon000209 IntCon000208

Design Assumptions

N/A

Item Location Publication API (BDI_ItemLoc_Fnd_PF_From_RMS_JOB)

This section describes the Item Location Publication BDI.

Functional Area

Foundation

Business Overview

BDI (Bulk Data Integration) is an integration layer that facilitates the bulk transfer of Item Location information from Merchandising to other Oracle Retail Applications. On this particular integration stream, the data flow is from Merchandising to BDI, and then BDI to downstream applications. To accomplish this data transfer, BDI will be calling a Merchandising-owned API that will pull data from Merchandising and deliver these to the BDI integration layer. This API will be in the form of a PLSQL function inside a PLSQL package.

Package Impact

This section describes the package impact.

Bulk Interface Module

Filename: bdiitemb.pls

```
BDI_ITEM_SQL.ITEM_LOC_UP(O_error_message IN OUT VARCHAR2,
                        O_control_id     IN OUT NUMBER,
                        I_job_context    IN   VARCHAR2)
```

This function begins by calling a BDI function that signals the start of the interface process. The BDI function will update the internal BDI control tables to track the progress of the API.

A DML insert statement is then executed to populate the BDI outbound table that resides in the BDI_RMS_INT_SCHEMA schema. This outbound table is loaded with records from the Merchandising Item Location table.

After the insert, another call to a BDI function is performed to signify the successful loading of records. This will update the internal BDI control tables.

A database commit is issued, and the control Id is returned by the API.

Data Definition XML

The BDI interface staging tables are generated based on the XML schema definition.

Data Flow	Description	XML Schema Definition (XSD)
Item Location	Item Location upload to BDI	ItemLoc_Fnd_BdiInterfaceModule.xml

Table Impact

TABLE	SELECT	INSERT	UPDATE	DELETE
ITEM_LOC_OUT	No	Yes	No	No
ITEM_LOC	Yes	No	No	No
ITEM_LOC_TRAITS	Yes	No	No	No

TABLE	SELECT	INSERT	UPDATE	DELETE
STORE	Yes	No	No	No
WH	Yes	No	No	No
PARTNER	Yes	No	No	No
ITEM_MASTER	Yes	No	No	No

Item Master Export (export_itemmaster.ksh)

Module Name	export_itemmaster.ksh
Description	Extraction of item data
Functional Area	Foundation
Module Type	Integration
Module Technology	ksh
Catalog ID	RMS258
Wrapper Script	rmswrap_shell_out.ksh

Schedule

Oracle Retail Merchandising Batch Schedule

Design Overview

This batch job will extract new, updated and deleted Merchandising item master information into a flat file.

Data to be extracted will be pulled off from the item export information, item export staging and the main item tables.

- The mode (full vs. delta) will be an input parameter for this new batch. The mode will allow a full extract (all approved, sellable items in Merchandising) as well as delta processing (all approved, sellable item changes in the main item table since the last export) of data.
- A threading indicator parameter should be passed. Passing 'Y' means a thread number (1-20) will be passed in. Passing 'N' means no thread number will be passed in and the program will use a default thread number.
- In full mode, normal operation will produce both a corporate level file and files for all stores. An optional input parameter will also allow the program to produce a location level file for a specified store.
- In delta mode, the only option is to produce corporate level files. Item header files at the store level will be created in the export_itemloc.ksh for delta mode.
- The store specific file will also include UPC items. To determine which UPC Items to include, the store where the UPC's parent and/or grandparent item is ranged should be taken into consideration.
- The flat files that will be created will now be pipe delimited.

Restart/Recovery

N/A

I/O Specification

Integration Type	Extract from Merchandising
File Name	itemhdr_[#date]_corp_[full/delta]_[#ofLines].dat itemhdr_[#date]_[location]_full_[#ofLines].dat
Integration Contract	IntCon000208

Design Assumptions

N/A

Item Master Publication API (BDI_ItemHdr_Fnd_PF_From_RMS_JOB)

This section describes the Item Master Publication BDI.

Functional Area

Foundation

Business Overview

BDI (Bulk Data Integration) is an integration layer that facilitates the bulk transfer of Item Master information from Merchandising to other Oracle Retail Applications. On this particular integration stream, the data flow is from Merchandising to BDI, and then BDI to downstream applications. To accomplish this data transfer, BDI calls a Merchandising-owned API that will pull data from Merchandising and deliver these to the BDI integration layer. This API is in the form of a PLSQL function inside a PLSQL package.

Package Impact

This section describes the package impact.

Bulk Interface Module

Filename: bdiitemb.pls

```
BDI_ITEM_SQL.ITEM_MASTER_UP(O_error_message IN OUT VARCHAR2,
                             O_control_id    IN OUT NUMBER,
                             I_job_context   IN     VARCHAR2)
```

This function begins by calling a BDI function that signals the start of the interface process. The BDI function will update the internal BDI control tables to track the progress of the API.

A DML insert statement is then executed to populate the BDI outbound table that resides in the BDI_RMS_INT_SCHEMA schema. This outbound table is loaded with records from the Merchandising Item Master table.

After the insert, another call to a BDI function is performed to signify the successful loading of records. This will update the internal BDI control tables.

A database commit is issued, and the control Id is returned by the API.

Data Definition XML

Data Flow	Description	XML Schema Definition (XSD)
Item Master	Item Master upload to BDI	ItemHdr_Fnd_BdiInterfaceModule.xml

Table Impact

TABLE	SELECT	INSERT	UPDATE	DELETE
ITEM_HDR_OUT	No	Yes	No	No
ITEM_MASTER	Yes	No	No	No
CLASS	Yes	No	No	No
SUBCLASS	Yes	No	No	No
DIFF_GROUP_HEAD	Yes	No	No	No
DIFF_IDS	Yes	No	No	No
SYSTEM_OPTIONS	Yes	No	No	No

Item Supplier Country Dimensions Publication API (BDI_ItSupCtryDim_Fnd_PF_From_RMS_JOB)

This section describes the Item Supplier Country Dim Publication BDI.

Functional Area

Item

Business Overview

BDI (Bulk Data Integration) is an integration layer that facilitates the bulk transfer of Item Supplier Country Dim information from Merchandising to other Oracle Retail Applications. On this particular integration stream, the data flow is from Merchandising to BDI, and then BDI to downstream applications. To accomplish this data transfer, BDI will be calling a Merchandising-owned API that will pull data from Merchandising and deliver these to the BDI integration layer. This API will be in the form of a PLSQL function inside a PLSQL package.

Package Impact

Filename: bdiitemb.pls

```
BDI_ITEM_SQL.ITEM_SUP_CTY_DIM_UP (O_error_message IN OUT VARCHAR2,
                                   O_control_id     IN OUT NUMBER,
                                   I_job_context    IN     VARCHAR2)
```

This function begins by calling a BDI function that signals the start of the interface process. The BDI function will update the internal BDI control tables to track the progress of the API.

A DML insert statement is then executed to populate the BDI outbound table that resides in the BDI_RMS_INT_SCHEMA schema. This outbound table is loaded with records from the Merchandising ITEM_SUPP_COUNTRY_DIM table.

After the insert, another call to a BDI function is performed to signify the successful loading of records. This will update the internal BDI control tables.

A database commit is issued, and the control Id is returned by the API.

Data Definition XML

The BDI interface staging tables are generated based on the XML schema definition

Data Flow	Description	XML Schema Definition (XSD)
Item Supplier Country Dim	Item supplier country Dim upload to BDI	ItSupCtryDim_Fnd_BdiInterfaceModule.xml

Table Impact

TABLE	SELECT	INSERT	UPDATE	DELETE
ITEM_SUP_CTY_DIM_OUT	No	Yes	No	No
ITEM_MASTER	Yes	No	No	No
ITEM_SUPP_COUNTRY_DIM	Yes	No	No	No

Item Supplier Country Publication API (BDI_ItSupCtry_Fnd_PF_From_RMS_JOB)

This section describes the Item Supplier Country Publication BDI.

Functional Area

Item

Business Overview

BDI (Bulk Data Integration) is an integration layer that facilitates the bulk transfer of Item supplier country information from Merchandising to other Oracle Retail Applications. On this particular integration stream, the data flow is from Merchandising to BDI, and then BDI to downstream applications. To accomplish this data transfer, BDI will be calling a Merchandising-owned API that will pull data from Merchandising and deliver these to the BDI integration layer. This API will be in the form of a PLSQL function inside a PLSQL package.

Package Impact

Filename: Filename: bdiitemb.pls

```
BDI_ITEM_SQL.ITEM_SUPP_COUNTRY_UP (O_error_message IN OUT VARCHAR2,
O_control_id IN OUT NUMBER,
I_job_context IN VARCHAR2)
```

This function begins by calling a BDI function that signals the start of the interface process. The BDI function will update the internal BDI control tables to track the progress of the API.

A DML insert statement is then executed to populate the BDI outbound table that resides in the BDI_RMS_INT_SCHEMA schema. This outbound table is loaded with records from the Merchandising ITEM_SUPP_COUNTRY table.

After the insert, another call to a BDI function is performed to signify the successful loading of records. This will update the internal BDI control tables.

A database commit is issued, and the control Id is returned by the API.

Data Definition XML

The BDI interface staging tables are generated based on the XML schema definition

Data Flow	Description	XML Schema Definition (XSD)
Item Supplier Country	Item supplier country upload to BDI	ItSupCtry_Fnd_BdiInterfaceModule.xml

Table Impact

TABLE	SELECT	INSERT	UPDATE	DELETE
ITEM_SUPP_COUNTRY_OUT	No	Yes	No	No
ITEM_MASTER	Yes	No	No	No
ITEM_SUPP_COUNTRY	Yes	No	No	No

Item Supplier Manufacturing Country Publication API (BDI_ItSupManCtry_Fnd_PF_From_RMS_JOB)

This section describes the Item Supplier Manufacturing Country Publication BDI.

Functional Area

Item

Business Overview

BDI (Bulk Data Integration) is an integration layer that facilitates the bulk transfer of Item Supplier Manufacturing Country information from Merchandising to other Oracle Retail Applications. On this particular integration stream, the data flow is from Merchandising to BDI, and then BDI to downstream applications. To accomplish this data transfer, BDI will be calling a Merchandising-owned API that will pull data from Merchandising and deliver these to the BDI integration layer. This API will be in the form of a PLSQL function inside a PLSQL package.

Package Impact

Filename: bdiitemb.pls


```
BDI_ITEM_SQL.ITEM_SUP_MAN_CTY_UP (O_error_message IN OUT VARCHAR2,
                                O_control_id     IN OUT NUMBER,
                                I_job_context    IN     VARCHAR2)
```

This function begins by calling a BDI function that signals the start of the interface process. The BDI function will update the internal BDI control tables to track the progress of the API.

A DML insert statement is then executed to populate the BDI outbound table that resides in the BDI_RMS_INT_SCHEMA schema. This outbound table is loaded with records from the Merchandising ITEM_SUPP_MANU_COUNTRY table.

After the insert, another call to a BDI function is performed to signify the successful loading of records. This will update the internal BDI control tables.

A database commit is issued, and the control Id is returned by the API.

Data Definition XML

The BDI interface staging tables are generated based on the XML schema definition

Data Flow	Description	XML Schema Definition (XSD)
Item Supplier Manufacturing Country	Item supplier Manufacturing Country upload to BDI	ItSupManCtry_Fnd_BdiInterfaceM odule.xml

Table Impact

TABLE	SELECT	INSERT	UPDATE	DELETE
ITEM_SUP_MAN_CTY_OUT	No	Yes	No	No
ITEM_MASTER	Yes	No	No	No
ITEM_SUPP_MANU_COUNTRY	Yes	No	No	No

Item Supplier Publication API (BDI_ItemSupp_Fnd_PF_From_RMS_JOB)

This section describes the Item Supplier Publication BDI.

Functional Area

Item

Business Overview

BDI (Bulk Data Integration) is an integration layer that facilitates the bulk transfer of Item supplier information from Merchandising to other Oracle Retail Applications. On this particular integration stream, the data flow is from Merchandising to BDI, and then BDI to downstream applications. To accomplish this data transfer, BDI will be calling a Merchandising-owned API that will pull data from Merchandising and deliver these to the BDI integration layer. This API will be in the form of a PLSQL function inside a PLSQL package.

Package Impact

Filename: bdiitemb.pls

```

BDI_ITEM_SQL.ITEM_SUPPLIER_UP(O_error_message IN OUT VARCHAR2,
                              O_control_id    IN OUT NUMBER,
                              I_job_context   IN     VARCHAR2)

```

This function begins by calling a BDI function that signals the start of the interface process. The BDI function will update the internal BDI control tables to track the progress of the API.

A DML insert statement is then executed to populate the BDI outbound table that resides in the BDI_RMS_INT_SCHEMA schema. This outbound table is loaded with records from the Merchandising ITEM_SUPPLIER table.

After the insert, another call to a BDI function is performed to signify the successful loading of records. This will update the internal BDI control tables.

A database commit is issued, and the control Id is returned by the API.

Data Definition XML

The BDI interface staging tables are generated based on the XML schema definition

Data Flow	Description	XML Schema Definition (XSD)
Item supplier	Item Supplier upload to BDI	ItemSupplier_Fnd_BdiInterfaceModule.xml

Table Impact

TABLE	SELECT	INSERT	UPDATE	DELETE
ITEM_SUPPLIER_OUT	No	Yes	No	No
ITEM_MASTER	Yes	No	No	No
ITEM_SUPPLIER	Yes	No	No	No

Item Supplier UOM Publication API (BDI_ItemSuppUom_Fnd_PF_From_RMS_JOB)

This section describes the Item Supplier UOM Publication BDI.

Functional Area

Item

Business Overview

BDI (Bulk Data Integration) is an integration layer that facilitates the bulk transfer of Item supplier UOM information from Merchandising to other Oracle Retail Applications. On this particular integration stream, the data flow is from Merchandising to BDI, and then BDI to downstream applications. To accomplish this data transfer, BDI will be calling a Merchandising-owned API that will pull data from Merchandising and deliver these to the BDI integration layer. This API will be in the form of a PLSQL function inside a PLSQL package.

Package Impact

Filename: bdiitemb.pls

```
BDI_ITEM_SQL.ITEM_SUPP_UOM_UP(O_error_message IN OUT VARCHAR2,
                              O_control_id    IN OUT NUMBER,
                              I_job_context   IN     VARCHAR2)
```

This function begins by calling a BDI function that signals the start of the interface process. The BDI function will update the internal BDI control tables to track the progress of the API.

A DML insert statement is then executed to populate the BDI outbound table that resides in the BDI_RMS_INT_SCHEMA schema. This outbound table is loaded with records from the Merchandising ITEM_SUPP_UOM table.

After the insert, another call to a BDI function is performed to signify the successful loading of records. This will update the internal BDI control tables.

A database commit is issued, and the control Id is returned by the API.

Data Definition XML

The BDI interface staging tables are generated based on the XML schema definition

Data Flow	Description	XML Schema Definition (XSD)
Item supplier UOM	Item Supplier UOM upload to BDI	ItemSuppUom_Fnd_BdiInterfaceModule.xml

Table Impact

TABLE	SELECT	INSERT	UPDATE	DELETE
ITEM_SUPP_UOM_OUT	No	Yes	No	No
ITEM_MASTER	Yes	No	No	No
ITEM_SUPP_UOM	Yes	No	No	No

Item VAT Rates Export (export_itemvat.ksh)

Module Name	export_itemvat.ksh
Description	Extraction of vat item data.
Functional Area	Foundation
Module Type	Integration
Module Technology	ksh
Catalog ID	RMS259
Wrapper Script	rmswrap_shell_out.ksh

Schedule

Oracle Retail Merchandising Batch Schedule

Design Overview

This batch job will extract new, updated and deleted Merchandising item VAT information into a flat file.

- This batch supports both a full and delta export of item VAT data.
- A threading indicator parameter should be passed. Passing 'Y' means a thread number (1-20) will be passed in. Passing 'N' means no thread number will be passed in and the program will use a default thread number.
- In full mode, normal operation will produce both a corporate level file and files for all stores. An optional input parameter will also allow the program to produce a location level file for a specified store.
- In full mode for store specific file if store belong to such a vat region, which is exempt (In case of tax type SVAT), then files for that store won't get generated.
- In delta mode, this will produce both corporate level files and files for all stores the modified items are ranged to and the vat region the store is associated with.
- In delta mode for store specific file if store belong to such a vat region, which is exempt, then files for that store won't get generated.
- This will export data only for approved, sellable items.
- This will export item VAT information from the item export staging and item tables.
- This should also include the item parent as its own record in the extract.
- The flat files that will be created will now be pipe delimited.

Restart/Recovery

N/A

I/O Specification

Integration Type	Extract from Merchandising
File Name	vatitem_[#date]_corp_[full/delta]_[#ofLines].dat vatitem_[#date]_[location]_[full/delta]_[#ofLines].dat
Integration Contract	IntCon000214

Design Assumptions

N/A

Pack Item Publication API (BDI_PckitemBrkout_Fnd_PF_From_RMS_JOB)

This section describes the Pack Item Publication BDI.

Functional Area

Item

Business Overview

BDI (Bulk Data Integration) is an integration layer that facilitates the bulk transfer of Pack Item information from Merchandising to other Oracle Retail Applications. On this particular integration stream, the data flow is from Merchandising to BDI, and then BDI to downstream applications. To accomplish this data transfer, BDI will be calling a Merchandising-owned API

that will pull data from Merchandising and deliver these to the BDI integration layer. This API will be in the form of a PLSQL function inside a PLSQL package.

Package Impact

Filename: bdiitemb.pls

```
BDI_ITEM_SQL.PACK_ITEM_UP(O_error_message IN OUT VARCHAR2,
                           O_control_id   IN OUT NUMBER,
                           I_job_context  IN      VARCHAR2)
```

This function begins by calling a BDI function that signals the start of the interface process. The BDI function will update the internal BDI control tables to track the progress of the API.

A DML insert statement is then executed to populate the BDI outbound table that resides in the BDI_RMS_INT_SCHEMA schema. This outbound table is loaded with records from the Merchandising PACKITEM table.

After the insert, another call to a BDI function is performed to signify the successful loading of records. This will update the internal BDI control tables.

A database commit is issued, and the control Id is returned by the API.

Data Definition XML

The BDI interface staging tables are generated based on the XML schema definition

Data Flow	Description	XML Schema Definition (XSD)
Pack Item	Pack Item upload to BDI	PackItem_Fnd_BdiInterfaceModule.xml

Table Impact

TABLE	SELECT	INSERT	UPDATE	DELETE
PACK_ITEM_OUT	No	Yes	No	No
ITEM_MASTER	Yes	No	No	No
PACKITEM	Yes	No	No	No

POS Configuration Data to 3rd Party POS (poscdnld)

Module Name	poscdnld.pc
Description	Download of POS Configuration Data to 3rd Party POS
Functional Area	Integration - 3rd Party POS
Module Type	Integration
Module Technology	ProC
Catalog ID	RMS69
Wrapper Script	rmswrap_out.ksh

Schedule

Oracle Retail Merchandising Batch Schedule

Design Overview

This program downloads POS configuration information from Merchandising to a flat file. This file can be used to load POS and back-office systems. This program (and its related prepost function) should only be run if Merchandising is used to master:

- Coupon definitions and relationships to items
- Restrictions on product sales, including but not limited to minimum age of purchaser, time/days when product cannot be sold, tenders that cannot be used to purchase the product, and so on.

Restart/Recovery

The logic unit of work is pos configuration type and pos configuration ID. The commit_max_ctr field should be set to prevent excessive rollback space usage, and to reduce the overhead of file I/O. The recommended commit counter setting is 1000 records (subject to change based on implementation).

I/O Specification

Integration Type	Download from Merchandising
File Name	Determined by runtime parameter.
Integration Contract	IntCon000063

Output File Layout

Table 6-5 Output File Layout

Record Name	Field Name	Field Type	Default Value	Description
FHEAD	Record Type	Char(5)	'FHEAD'	Record Identifier
	Line id	Number(10)	0000000001	Sequential Line Identifier
	File Name	Char(4)	'POSC'	File Identifier
	File Date	Char(14)	N/A	Date the file was created in 'YYYYMMDD HHMMSS' format
TCOUP	Record Type	Char(5)	TCOUP	Record Identifier
	Line id	Number(10)	N/A	Sequential Line Identifier
	Coupon id	Number(6)	N/A	N/A
	Coupon Desc	Char(250)	N/A	N/A
	Currency Code	Char2(3)	N/A	N/A

Table 6-5 (Cont.) Output File Layout

Record Name	Field Name	Field Type	Default Value	Description
	Max Discount Amount	Number(20,4)	N/A	N/A
	Amount	Number(20,4)	N/A	N/A
	Percent Ind	Char(1)	'N' - Amount 'Y' - Percentage	N/A
	Profit Center	Char(6)	N/A	N/A
	Tax Class	Char(6)	N/A	N/A
	Export Code	Char(6)	N/A	N/A
	Effective Date	Char(14)	N/A	Indicates the first day the coupon can be used in 'YYYYMMDD HHMMSS' format
	Expiration Date	Char(14)	N/A	Indicates the day the coupon becomes invalid in 'YYYYMMDD HHMMSS' format
	Prompted Ind	Char(1)	'Y', 'N'	This indicator identifies if the cashier should be prompted to ask for a Coupon.
	Display Ind	Char(1)	'Y', 'N'	This indicator specifies whether the coupon is displayed in the list of valid coupons on the register.
	Status	Char(1)	'A','C','D'	Indicates if the coupon configuration is new, has been changed, or being deleted.
	Vendor	Number(10)	N/A	N/A

Table 6-5 (Cont.) Output File Layout

Record Name	Field Name	Field Type	Default Value	Description
TPRES	Vendor Type	Char(6)	'AG' - Agent 'AP' - Applicant 'BK' - Bank 'BR' - Broker 'CN' - Coconsignee 'CO' - Consolidator 'FA' - Factory 'FF' - Freight Forwarder 'IM' - Importer 'SU' - Supplier	N/A
	Promotion	Number(10)	N/A	N/A
	Coupon Barcode	Char(20)	N/A	N/A
	Coupon Max Qty	Number(6)	N/A	N/A
	Record Type	Char(5)	TPRES	Record Identifier
	Line id	Number(10)	N/A	Sequential Line Identifier
	POS Product Restriction id	Number(6)	N/A	N/A
	POS Product Restriction Desc	Char(120)	N/A	N/A

Table 6-5 (Cont.) Output File Layout

Record Name	Field Name	Field Type	Default Value	Description
	POS Product Restriction Type	Char(6)	'PPRT' include: 'STMP' - Food Stamp 'MNAG' - Minimum Age 'CNDP' - Container Deposit 'CNVL' - Container Redemption Value 'DTDR' - Day/Time/Date Restriction 'TENT' - Tender Type 'NDSC' - Non-Discountable 'RTRN' - Returnable 'QLMT' - Quantity Limit	N/A
	Effective Date	Char(14)	N/A	Date the product restriction is first effective in 'YYYYMMDD HHMMSS' format
	Currency Code	Char(3)	N/A	N/A
	Product Restriction Amount	Number(20,4)	N/A	N/A
	Age Minimum	Number(2)	N/A	N/A
	Date Restriction	Char(14)	N/A	Date on which a specified product restriction is applied in 'YYYYMMDD HHMMSS' format
	Before Time Restriction	Char(6)	N/A	N/A
	After Time Restriction	Char(6)	N/A	N/A
	Day Restriction	Char(6)	N/A	N/A
	Max Qty Amount	Number(12,4)	N/A	N/A

Table 6-5 (Cont.) Output File Layout

Record Name	Field Name	Field Type	Default Value	Description
	Tender Type Group	Char(6)	'CASH' - Cash, 'CHECK' - Check, 'CCARD' - Credit, 'COUPON' - Coupon, 'LOTTRY' - Lottery, 'FSTAMP' - Food Stamp, 'DCARD' - Debit Card, 'MORDER' - Money Order 'VOUCH' - Voucher 'ERR' - Error, 'SOCASS' - Social Assistance, 'TERM' - Termination Record, 'DRIVEO' - Drive Off, 'EBS' - Electronic Benefits (Food Stamps)	N/A
	Status	Char(1)	'A','C','D'	Indicates if the product restriction configuration is new, has been changed, or being deleted.
TSTOR	Record Type	Char(5)	'TSTOR'	N/A
	Line id	Number(10)	N/A	N/A
	Store	Number(10)	N/A	N/A
	Status	Char(1)	'A' - Add 'D' - Delete 'C' - Change	N/A
TITEM	Record Type	Char(5)	TITEM	Record Identifier
	Line id	Number(10)	N/A	Sequential Line Identifier

Table 6-5 (Cont.) Output File Layout

Record Name	Field Name	Field Type	Default Value	Description
	Item	Char(25)	N/A	Left-Justified Item Identifier
	Status	Char(1)	'A' - Add 'D' - Delete 'C' - Change	Indicates the item's status at the POS. Overlays of items as a result of a change to the merch criteria will have a 'C' status.
FTAIL	Record Type	Char(5)	FTAIL	Marks end of file
	Line id	Number(10)	N/A	Total number of lines in file
	Number of transactions	Number(10)	N/A	Number of transactions in file

Design Assumptions

N/A

Price History Publication API (BDI_PriceHist_Fnd_PF_From_RMS_JOB)

This section describes the Price History Publication BDI.

Functional Area

Foundation

Business Overview

BDI (Bulk Data Integration) is an integration layer that facilitates the bulk transfer of Price History positions from Merchandising to other Oracle Retail Applications. On this particular integration stream, the data flow is from Merchandising to BDI, and then BDI to downstream applications. To accomplish this data transfer, BDI will be calling a Merchandising-owned API that will pull data from Merchandising and deliver these to the BDI integration layer. This API will be in the form of a PLSQL function inside a PLSQL package.

Package Impact

Filename: **bdifoundations/b.pls**

```
BDI_FOUNDATION_SQL.PRICE_HIST_UP(O_error_message IN OUT VARCHAR2,
                                O_control_id     IN OUT NUMBER,
                                I_job_context    IN     VARCHAR2)
```

This function begins by calling a BDI function that signals the start of the interface process. The BDI function will update the internal BDI control tables to track the progress of the API.

A DML insert statement is then executed to populate the BDI outbound table that resides in the BDI_RMS_INT_SCHEMA schema. This outbound table is loaded with records from the Merchandising Price History tables.

After the insert, another call to a BDI function is performed to signify the successful loading of records. This will update the internal BDI control tables.

A database commit is issued, and the control Id is returned by the API.

Data Definition XML

The BDI interface staging tables are generated based on the XML schema definition

Data Flow	Description	XML Schema Definition (XSD)
Price History	Price History upload to BDI	PriceHist_Fnd_BdiInterfaceModule.xml

Table Impact

TABLE	SELECT	INSERT	UPDATE	DELETE
PRICE_HIST_OUT	No	Yes	No	No
PRICE_HIST	Yes	No	No	No

Related Items Export (export_relitem.ksh)

Module Name	export_relitem.ksh
Description	Extraction of related item data
Functional Area	Foundation
Module Type	Integration
Module Technology	ksh
Catalog ID	RMS262
Wrapper Script	rmswrap_shell_out.ksh

Schedule

Oracle Retail Merchandising Batch Schedule

Design Overview

This batch job will extract new, updated and deleted Merchandising related items information into a flat file.

- This batch will support both a full and delta export of related item data.
- A threading indicator parameter should be passed. Passing 'Y' means a thread number (1-20) will be passed in. Passing 'N' means no thread number will be passed in and the program will use a default thread number.

- In full mode, normal operation will produce both a corporate level files and files for all stores. An optional input parameter will also allow the program to produce location level files for a specified store.
- In delta mode, this will produce both corporate level files and files for all stores the modified data are ranged to.
- This will export data only for approved, sellable items.
- This will export item related item information from the related item export staging and related item tables.
- Two types of flat files will be created for this extract - one for the related item header information and one for the related item detail information.
- When creating the location level files, ensure that both items (the main item and related item) are ranged in the location.
- The flat files that will be created will now be pipe delimited.

Restart/Recovery

N/A

I/O Specification

Integration Type	Extract from Merchandising
File Name	relitemhead_date_corp_[full/delta]_[#ofLines].dat relitemhead_date_[Location]_[full/delta]_[#ofLines].dat relitemdet_date_corp_[full/delta]_[#ofLines].dat relitemdet_date_[Location]_[full/delta]_[#ofLines].dat
Integration Contract	IntCon000210 IntCon000211

Design Assumptions

N/A

Related Item Publication API (BDI_RelatedItem_Fnd_PF_From_RMS_JOB)

This section describes the Related Item Publication BDI.

Functional Area

Foundation

Business Overview

BDI (Bulk Data Integration) is an integration layer that facilitates the bulk transfer of Related Items from Merchandising to other Oracle Retail Applications. On this particular integration stream, the data flow is from Merchandising to BDI, and then BDI to downstream applications. To accomplish this data transfer, BDI will be calling a Merchandising-owned API that will pull data from Merchandising and deliver these to the BDI integration layer. This API will be in the form of a PLSQL function inside a PLSQL package.

Package Impact

Filename: bdiitemb.pls

```
BDI_ITEM_SQL.REL_ITEM_UP(O_error_message IN OUT VARCHAR2,
                        O_control_id     IN OUT NUMBER,
                        I_job_context     IN     VARCHAR2)
```

This function begins by calling a BDI function that signals the start of the interface process. The BDI function will update the internal BDI control tables to track the progress of the API.

A DML insert statement is then executed to populate the BDI outbound tables that reside in the BDI_RMS_INT_SCHEMA schema. These outbound tables are loaded with records from the Merchandising Related Item head and detail tables.

After the insert, another call to a BDI function is performed to signify the successful loading of records. This will update the internal BDI control tables.

A database commit is issued, and the control Id is returned by the API.

Data Definition XML

The BDI interface staging tables are generated based on the XML schema definition

Data Flow	Description	XML Schema Definition (XSD)
Related Item	Related Item upload to BDI	RelatedItem_Fnd_BdiInterfaceModule.xml

Table Impact

TABLE	SELECT	INSERT	UPDATE	DELETE
RELATED_ITEM_OUT	No	Yes	No	No
RELATED_ITEM_DTL_OUT	No	Yes	No	No
RELATED_ITEM_HEAD	Yes	No	No	No
RELATED_ITEM_DETAIL	Yes	No	No	No
ITEM_MASTER	Yes	No	No	No

UDA Item Date Publication API (BDI_UdaltemDate_Fnd_PF_From_RMS_JOB)

This section describes the UDA Item Date Publication BDI.

Functional Area

Foundation

Business Overview

BDI (Bulk Data Integration) is an integration layer that facilitates the bulk transfer of Code Head information from Merchandising to other Oracle Retail Applications. On this particular integration stream, the data flow is from Merchandising to BDI, and then BDI to downstream applications. To accomplish this data transfer, BDI will be calling a Merchandising-owned API

that will pull data from Merchandising and deliver these to the BDI integration layer. This API will be in the form of a PLSQL function inside a PLSQL package.

Package Impact

Filename: bdiitemb.pls

```
BDI_ITEM_SQL.UDA_ITEM_DATE_UP(O_error_message IN OUT RTK_ERRORS.RTK_TEXT%TYPE,
                              O_control_id    IN OUT NUMBER,
                              I_job_context   IN    VARCHAR2)
```

This function begins by calling a BDI function that signals the start of the interface process. The BDI function will update the internal BDI control tables to track the progress of the API.

A DML insert statement is then executed to populate the BDI outbound table that resides in the BDI_RMS_INT_SCHEMA schema. This outbound table is loaded with records from the Merchandising UDA Item Date table.

After the insert, another call to a BDI function is performed to signify the successful loading of records. This will update the internal BDI control tables.

A database commit is issued, and the control Id is returned by the API.

Data Definition XML

The BDI interface staging tables are generated based on the XML schema definition

Data Flow	Description	XML Schema Definition (XSD)
UDA ITEM DATE	UDA Item Date upload to BDI	UdaltemDate_Fnd_BdiInterfaceModule.xml

Table Impact

TABLE	SELECT	INSERT	UPDATE	DELETE
UDA_ITEM_DATE_OUT	No	Yes	No	No
UDA_ITEM_DATE	Yes	No	No	No

UDA Item FF Publication API (BDI_UdaltemFF_Fnd_PF_From_RMS_JOB)

This section describes the UDA Item FF Publication BDI.

Functional Area

Foundation

Business Overview

BDI (Bulk Data Integration) is an integration layer that facilitates the bulk transfer of Code Head information from Merchandising to other Oracle Retail Applications. On this particular integration stream, the data flow is from Merchandising to BDI, and then BDI to downstream applications. To accomplish this data transfer, BDI will be calling a Merchandising-owned API that will pull data from Merchandising and deliver these to

the BDI integration layer. This API will be in the form of a PLSQL function inside a PLSQL package.

Package Impact

Filename: bdiitemb.pls

```
BDI_ITEM_SQL.UDA_ITEM_FF_UP(O_error_message IN OUT RTK_ERRORS.RTK_TEXT%TYPE,
                             O_control_id   IN OUT NUMBER,
                             I_job_context  IN   VARCHAR2)
```

This function begins by calling a BDI function that signals the start of the interface process. The BDI function will update the internal BDI control tables to track the progress of the API.

A DML insert statement is then executed to populate the BDI outbound table that resides in the BDI_RMS_INT_SCHEMA schema. This outbound table is loaded with records from the Merchandising UDA Item FF table.

After the insert, another call to a BDI function is performed to signify the successful loading of records. This will update the internal BDI control tables.

A database commit is issued, and the control Id is returned by the API.

Data Definition XML

The BDI interface staging tables are generated based on the XML schema definition

Data Flow	Description	XML Schema Definition (XSD)
UDA ITEM FF	UDA Item FF upload to BDI	UdaltemFF_Fnd_BdiInterfaceModule.xml

Table Impact

TABLE	SELECT	INSERT	UPDATE	DELETE
UDA_ITEM_FF_OUT	No	Yes	No	No
UDA_ITEM_FF	Yes	No	No	No

UDA Item LOV Publication API (BDI_UdaltemLov_Fnd_From_RMS_JOB)

This section describes the UDA Item LOV Publication BDI.

Functional Area

Foundation

Business Overview

BDI (Bulk Data Integration) is an integration layer that facilitates the bulk transfer of Code Head information from Merchandising to other Oracle Retail Applications. On this particular integration stream, the data flow is from Merchandising to BDI, and then BDI to downstream applications. To accomplish this data transfer, BDI will be calling a Merchandising-owned API that will pull data from Merchandising and deliver these to the BDI integration layer. This API will be in the form of a PLSQL function inside a PLSQL package.

Package Impact

Filename: bdiitemb.pls

```
BDI_ITEM_SQL.UDA_ITEM_LOV_UP(O_error_message IN OUT RTK_ERRORS.RTK_TEXT%TYPE,
                             O_control_id    IN OUT NUMBER,
                             I_job_context   IN     VARCHAR2)
```

This function begins by calling a BDI function that signals the start of the interface process. The BDI function will update the internal BDI control tables to track the progress of the API.

A DML insert statement is then executed to populate the BDI outbound table that resides in the BDI_RMS_INT_SCHEMA schema. This outbound table is loaded with records from the Merchandising UDA Item LOV table.

After the insert, another call to a BDI function is performed to signify the successful loading of records. This will update the internal BDI control tables. A database commit is issued, and the control Id is returned by the API.

Data Definition XML

The BDI interface staging tables are generated based on the XML schema definition

Data Flow	Description	XML Schema Definition (XSD)
UDA ITEM LOV	UDA Item LOV upload to BDI	UdaltemLov_Fnd_BdiInterfaceModule.xml

Table Impact

TABLE	SELECT	INSERT	UPDATE	DELETE
UDA_ITEM_LOV_OUT	No	Yes	No	No
UDA_ITEM_LOV	Yes	No	No	No

Financials

Merchandising stages General Ledger (GL) data for subsequent upload into a financial system. A set of batch processes gather and organize the data before using it to populate the staging table, STG_FIF_GL_DATA.

For more information about how data moves from these staging tables to the General Ledger of a financial application and other integration between Merchandising and financial applications, see *Oracle Retail Financial Integration for Oracle Retail Merchandise Operations Management* and *Oracle E-Business Suite Financials Implementation Guide*.

The following scheduled outbound integrations are included in this functional area:

- [Daily or Weekly Download of Stock Ledger Data \(stlgnld\)](#)
- [Finance General Ledger to RFI \(BDI_RFI_FinGenLdgr_Tx_PF_From_RMS_JOB\)](#)
- [Fixed Deal Income \(dealfinc\)](#)
- [Franchise Billing Extract \(wfbillex.ksh\)](#)

- [Item/Location Daily Stock Ledger Transactions \(fifglDn1\)](#)
- [Monthly Stock Ledger Transactions \(fifglDn3\)](#)
- [Open to Buy Download Stock Ledger \(otbdlsal\)](#)
- [Rolled Up Daily Stock Ledger Transactions \(fifglDn2\)](#)
- [Stage G/L Extracts \(gl_extract.ksh\)](#)
- [Tran Data Publication \(BDI_TrانData_Tx_PF_From_RMS_EOW_JOB\)](#)

Daily or Weekly Download of Stock Ledger Data (stlgdnlD)

Module Name	stlgdnlD.pc
Description	Weekly or Historical Download of Stock Ledger Data
Functional Area	Stock Ledger
Module Type	Integration
Module Technology	ProC
Catalog ID	RMS17
Wrapper Script	batch_stlgdnlD.ksh

Schedule

Oracle Retail Merchandising Batch Schedule

Design Overview

This program extracts stock ledger data at the item level. The program can extract data for a historic period or for the most current complete week. The program accepts an input file that determines whether the extract is a historic extract or a weekly extract.

This program is often used in integration with RPAS applications.

Restart/Recovery

The logical unit of work for this program is set at item, location type, location and date. Threading is done by dept using the v_restart_dept view to thread properly.

The changes will be posted when the commit_max_ctr value is reached. The commit_max_ctr field should be set to prevent excessive rollback space usage, and to reduce the overhead of file I/O. The value of the counter is subject to change based on implementation.

I/O Specification

Integration Type	Download from Merchandising
File Name	The input filename is a runtime parameter. The output filename is hardcoded to stklDgr%d.dat where %d is substituted with the domain id. Each run of the program can produce multiple output files, one for each department. Additional input parameters are defined in the input file
Integratin Contract	IntCon000034 (output file)

Input File Layout

Table 6-6 Input File Layout

Field Name	Field Type	Default Value	Description
Task Indicator	Char(1)	N/A	Task Indicator. Valid values are 'H' - historical, 'W' - weekly
From Date	Char(8)	N/A	From Date in 'YYYYMMDD' format
To Date	Char(8)	N/A	To Date in 'YYYYMMDD' format

Output File Layout

Table 6-7 Output File Layout

Field Name	Field Type	Default Value	Description
Item	Char(25)	N/A	Item number
Location Type	Char(1)	N/A	Location Type Valid values are 'S','W'
Location	Number(20)	N/A	Location Number
Eow_date	Char(8)	N/A	End of Week date in 'YYYYMMDD' format
Update_Ind	Char(1)	N/A	Update Indicator Valid values are 'I' and 'U'
Regular_sales_retail	Number(25,4)	N/A	Regular sales value (retail)
Regular_sales_cost	Number(25,4)	N/A	Regular sales value (cost)
Regular_sales_units	Number(17,4)	N/A	Regular sales value (units)
Promo_sales_retail	Number(25,4)	N/A	Promo sales value (retail)
Promo_sales_cost	Number(25,4)	N/A	Promo sales value (cost)
Promo_sales_units	Number(17,4)	N/A	Promo sales value (units)
Clear_sales_retail	Number(25,4)	N/A	Clearance sales value (retail)
Clear_sales_cost	Number(25,4)	N/A	Clearance sales value (cost)

Table 6-7 (Cont.) Output File Layout

Field Name	Field Type	Default Value	Description
Clear_sales_units	Number(17,4)	N/A	Clearance sales value (units)
Sales_retail_excluding_vat	Number(25,4)	N/A	Sales value excluding vat (retail)
Custom_returns_retail	Number(25,4)	N/A	Custom returns value (retail)
Custom_returns_cost	Number(25,4)	N/A	Custom returns value (cost)
Custom_returns_units	Number(17,4)	N/A	Custom returns value (units)
Rtv_retail	Number(25,4)	N/A	Return to Vendor value (retail)
Rtv_cost	Number(25,4)	N/A	Return to Vendor value (cost)
Rtv_units	Number(17,4)	N/A	Return to Vendor value (units)
Reclass_in_retail	Number(25,4)	N/A	Reclass In value (retail)
Reclass_in_cost	Number(25,4)	N/A	Reclass In value (cost)
Reclass_in_units	Number(17,4)	N/A	Reclass In value (units)
Reclass_out_retail	Number(25,4)	N/A	Reclass Out value (retail)
Reclass_out_cost	Number(25,4)	N/A	Reclass Out value (cost)
Reclass_out_units	Number(17,4)	N/A	Reclass Out value (units)
Perm_markdown_value	Number(25,4)	N/A	Permanent markdown value (retail)
Prom_markdown_value	Number(25,4)	N/A	Promotion markdown value (retail)
Clear_markdown_value	Number(25,4)	N/A	Clearance markdown value (retail)
Markdown_cancel_value	Number(25,4)	N/A	Markdown cancel value
Markup_value	Number(25,4)	N/A	Markup value
Markup_cancel_value	Number(25,4)	N/A	Markup cancel value
Stock_adj_retail	Number(25,4)	N/A	Stock adjustment value (retail)

Table 6-7 (Cont.) Output File Layout

Field Name	Field Type	Default Value	Description
Stock_adj_cost	Number(25,4)	N/A	Stock adjustment value (cost)
Stock_adj_units	Number(17,4)	N/A	Stock adjustment value (units)
Received_retail	Number(25,4)	N/A	Received value (retail)
Received_cost	Number(25,4)	N/A	Received value (cost)
Received_units	Number(17,4)	N/A	Received value (units)
Tsf_in_retail	Number(25,4)	N/A	Transfer In value (retail)
Tsf_in_cost	Number(25,4)	N/A	Transfer In value (cost)
Tsf_in_units	Number(17,4)	N/A	Transfer In value (units)
Tsf_out_retail	Number(25,4)	N/A	Transfer Out value (retail)
Tsf_out_cost	Number(25,4)	N/A	Transfer Out value (cost)
Tsf_out_units	Number(17,4)	N/A	Transfer Out value (units)
Freight_cost	Number(25,4)	N/A	Freight cost
Employee_disc_retail	Number(25,4)	N/A	Employee disc (retail)
Cost_variance	Number(25,4)	N/A	Cost variance
Wkroom_other_cost_sales	Number(25,4)	N/A	Wkroom other sales (cost)
Cash_disc_retail	Number(25,4)	N/A	Cash disc (retail)
Freight_claim_retail	Number(25,4)	N/A	Freight Claim (retail)
Freight_claim_cost	Number(25,4)	N/A	Freight Claim (cost)
Freight_claim_units	Number(25,4)	N/A	Freight Claim (Units)
Stock_adj_cogs_retail	Number(25,4)	N/A	Stock Adjust COGS (retail)
Stock_adj_cogs_cost	Number(25,4)	N/A	Stock Adjust COGS (cost)
Stock_adj_cogs_units	Number(25,4)	N/A	Stock Adjust COGS (Units)
Intercompany_in_retail	Number(25,4)	N/A	Intercompany In value (retail)
Intercompany_in_cost	Number(25,4)	N/A	Intercompany In value (cost)

Table 6-7 (Cont.) Output File Layout

Field Name	Field Type	Default Value	Description
Intercompany_in_units	Number(25,4)	N/A	Intercompany In value (units)
Intercompany_out_retail	Number(25,4)	N/A	Intercompany Out value (retail)
Intercompany_out_cost	Number(25,4)	N/A	Intercompany Out value (cost)
Intercompany_out_units	Number(25,4)	N/A	Intercompany Out value (units)
Intercompany_markup	Number(25,4)	N/A	Intercompany Markup
Intercompany_markup_units	Number(25,4)	N/A	Intercompany Markup (units)
Intercompany_markdown	Number(25,4)	N/A	Intercompany Markdown
Intercompany_markdown_units	Number(25,4)	N/A	Intercompany Markdown (units)
Wo_activity_upd_inv	Number(25,4)	N/A	Work Order Activity - Update Inventory (cost)
Wo_activity_upd_inv_units	Number(25,4)	N/A	Work Order Activity - Update Inventory (units)
Wo_activity_post_fin	Number(25,4)	N/A	Work Order Activity - Post to Financials (retail)
Wo_activity_post_fin_units	Number(25,4)	N/A	Work Order Activity - Post to Financials (units)

Design Assumptions

N/A

Finance General Ledger to RFI (BDI_RFI_FinGenLdgr_Tx_PF_From_RMS_JOB)

Module Name	BDI_RFI_FinGenLdgr_Tx_PF_From_RMS_JOB
Description	Extracts staged data from Merchandising and Sales Audit to the General Ledger
Functional Area	Finance
Module Type	Integration
Module Technology	BDI Job
Catalog ID	N/A
Runtime Parameters	FinGenLdgr_Tx_ProcessFlow_From_RMS FinGenLdgr_Tx_Extractor

Design Overview

This API extracts staged data from Merchandising and Sales Audit and transfers it to the General Ledger inbound staging tables. To accomplish this data transfer, BDI will call a Merchandising-owned API that will pull data from Merchandising and deliver these to the BDI integration layer.

This integration is applicable when using Oracle Retail Financial Integration (RFI) to supported financial solutions. For more information on RFI, see the *ORFI Implementation Guide*, as part of the documentation for Oracle Retail Integration Cloud Service.

Scheduling Constraints

Schedule Information	Description
Processing Cycle	End of Day
Frequency	Daily
Scheduling Consideration	N/A
Pre-Processing	N/A
Post-Processing	N/A
Threading Scheme	N/A

Data Definition XML

The BDI interface staging tables are generated based on the XML schema definition.

Data Flow	Description	XML Schema Definition (XML)
Finance	General Ledger upload to BDI	FinGenLdgr_Tx_BdiInterfaceModule.xml

Fixed Deal Income (dealfinc)

Module Name	dealfinc.pc
Description	Calculation & Interface of Fixed Deal Income for General Ledger
Functional Area	Integration - General Ledger
Module Type	Integration
Module Technology	ProC
Catalog ID	RMS65
Wrapper Script	rmswrap_multi.ksh

Schedule

Oracle Retail Merchandising Batch Schedule

Design Overview

This module writes to the STG_FIF_GL_DATA financial staging table to perform stock ledger processing for fixed deals. It splits deal income over all dept/class/subclass locations on the deal. This prorated income is written to the general ledger under a suitable cost center mapping.

Restart/Recovery

The logical unit of work for this program is a DEAL_ID. The database commit takes place when number of deal records processed is equal to the commit max counter in the restart control table.

I/O Specification

Integration Type	Download from Merchandising
File Name	N/A
Integration Contract	IntCon000019 STG_FIF_GL_DATA table

Design Assumptions

N/A

Franchise Billing Extract (wfbillex.ksh)

Module Name	wfbillex.ksh
Description	Franchise Billing Extract
Functional Area	Franchise Management
Module Type	Integration
Module Technology	ksh
Catalog ID	RMS155
Wrapper Script	rmswrap_shell.ksh

Schedule

Oracle Retail Merchandising Batch Schedule

Design Overview

The purpose of this shell script module is to fetch all billing information for Franchise sale and return transactions and write these to an output file for integration with an external financial application that manages billing. A file is generated for each customer location (store)/day.

The format of the generated file is based on a run time parameter:

- If no parameter is passed or if the value 1 is used, the previously existing format (refer Output File Layout Format 1) will be generated.

- If the value 2 (or greater than 2) is passed, the new file format (refer Output File Layout Format 2) will be generated.

Restart/Recovery

The logical unit of work for this module is defined as the customer location (store). Only one commit will be done for a customer location that has been completely processed. The WFBX formatted output file will be created with a temporary name and renamed just before a customer location commit. In case of failure, all work done will be rolled back.

I/O Specification

Integration Type Download from Merchandising
File Name WFBX_<store>_<SYSDATE>
Integration Contract IntCon000110

Output File Layout

Table 6-8 Output File Layout Format 1

Record Name	Field Name	Field Type	Default Value	Description
FHEAD	Record descriptor	Char(5)	FHEAD	Identifies the file record type
	File Line Id	Char(10)		Sequential file line number
	File type definition	Char(4)	WFBX	Identifies the file type
	File Create Date	Char(14)		File Create Date in YYYYMMDDH HMMSS format
THEAD	Record descriptor	Char(5)	THEAD	Identifies the file record type
	File Line Id	Char(10)		Sequential file line number
	Customer Location	Number(10)		Franchise store number
	Customer Order Reference Number	Char(20)		Reference number provided by the franchise customer
	Franchise Order Number	Number(10)		Franchise Order Number
	Transaction Type	Char(6)		SALES or RETURN

Table 6-8 (Cont.) Output File Layout Format 1

Record Name	Field Name	Field Type	Default Value	Description
	RMA Number	Number(10)		Return Merchandise Authorization Number for the return
	Order Return Date	Number(8)		Order return date for Return transaction type or Order date for Sale transaction type in YYYYMMDD format
	Shipment Date	Number(8)		Date on which the item was shipped to the franchise location or returned to the retailer
TDETL	Record descriptor	Char(5)	TDETL	Identifies the file record type
	File Line Id	Char(10)		Sequential file line number
	Item	Char(25)		Item sequence number
	Department	Number(4)		Department number of the item
	Class	Number(4)		Class number of the item
	Subclass	Char(4)		Subclass number of the item
	Order Return Quantity	Number(12)		Return quantity with 4 implied decimal places
	Order Return Quantity UOM	Char(4)		Return quantity unit of measure
	Order Return Cost	Number(20)		Return cost for Return transaction type or Customer cost for Sale transaction type. For both it is the per-unit cost

Table 6-8 (Cont.) Output File Layout Format 1

Record Name	Field Name	Field Type	Default Value	Description
	Freight Cost	Number(20)		Freight associated to the franchise order
	Return Restocking Fee	Number(20)		Unit restocking fee charged for received items
	VAT Code	Char(6)		VAT code for the item
	VAT Rate	Number(20)		VAT rate associated to the VAT code for the item
	Other Order Charges	Number(20)		Other charges for the item
TTAIL	Record descriptor	Char(5)	TTAIL	Identifies the file record type
	File Line Id	Char(10)		Sequential file line number
	Tran Record Counter	Number(6)		Number of TDETL records in this transaction set
FTAIL	Record descriptor	Char(5)	FTAIL	Identifies the file record type
	File Line Id	Number(10)		Sequential file line number
	File Record counter	Number(10)		Number of records/ transactions processed in current file (only records between head & tail)

Table 6-9 Output File Layout Format 2

Record Name	Field Name	Field Type	Default Value	Description
FHEAD	Record descriptor	Char(5)	FHEAD	Identifies the file record type
	File Line Id	Char(10)		Sequential file line number
	File type definition	Char(4)	WFBX	Identifies the file type

Table 6-9 (Cont.) Output File Layout Format 2

Record Name	Field Name	Field Type	Default Value	Description
THEAD	File Create Date	Char(14)		File Create Date in YYYYMMDDH HMMSS format
	Version No.	Char(2)	02	Identifies the file format version
	Record descriptor	Char(5)	THEAD	Identifies the file record type
	File Line Id	Char(10)		Sequential file line number
	Customer Location	Number(10)		Franchise store number
	Customer Order Reference Number	Char(20)		Reference number provided by the franchise customer
	Franchise Order Number	Number(10)		Franchise Order Number
	Transaction Type	Char(6)		SALES or RETURN
	RMA Number	Number(10)		Return Merchandise Authorization Number for the return
	Order Return Date	Number(8)		Order return date for Return transaction type or Order date for Sale transaction type in YYYYMMDD format
TDETL	Shipment Date	Number(8)		Date on which the item was shipped to the franchise location or returned to the retailer
	Record descriptor	Char(5)	TDETL	Identifies the file record type
	File Line Id	Char(10)		Sequential file line number
	Item	Char(25)		Item sequence number

Table 6-9 (Cont.) Output File Layout Format 2

Record Name	Field Name	Field Type	Default Value	Description
	Department	Number(4)		Department number of the item
	Class	Number(4)		Class number of the item
	Subclass	Char(4)		Subclass number of the item
	Order Return Quantity	Number(12)		Return quantity with 4 implied decimal places
	Order Return Quantity UOM	Char(4)		Return quantity unit of measure
	Order Return Cost	Number(20)		Return cost for Return transaction type or Customer cost for Sale transaction type. For both it is the per-unit cost
	Freight Cost	Number(20)		Freight associated to the franchise order
	Return Restocking Fee	Number(20)		Unit restocking fee charged for received items
	VAT Code	Char(6)		VAT code for the item
	VAT Rate	Number(20)		VAT rate associated to the VAT code for the item
	Other Order Charges	Number(20)		Other charges for the item
	Uom Type	Char(4)		Uom Type fetched from GTS tax engine to be considered when tax is in value

Table 6-9 (Cont.) Output File Layout Format 2

Record Name	Field Name	Field Type	Default Value	Description
	Uom Value	Number(20)		UOM value fetched from GTS tax engine to be considered when tax is in value. Having 10 places of decimal digits.
	Uom Tax Value Per Unit	Number(20)		UOM tax value per unit fetched from GTS tax engine to be considered when tax is in value. Having 10 places of decimal digits.
TTAIL	Record descriptor	Char(5)	TTAIL	Identifies the file record type
	File Line Id	Char(10)		Sequential file line number
	Tran Record Counter	Number(6)		Number of TDETL records in this transaction set
FTAIL	Record descriptor	Char(5)	FTAIL	Identifies the file record type
	File Line Id	Number(10)		Sequential file line number
	File Record counter	Number(10)		Number of records/ transactions processed in current file (only records between head & tail)

Design Assumptions

N/A

Item/Location Daily Stock Ledger Transactions (fifgldn1)

Module Name	fifgldn1.pc
Description	Interface to General Ledger of Item/Loc Level Transactions
Functional Area	General Ledger

Module Type	Integration
Module Technology	ProC
Catalog ID	RMS66
Wrapper Script	rmswrap_multi.ksh

Schedule

Oracle Retail Merchandising Batch Schedule

Design Overview

This program extracts the detailed stock ledger information for certain transaction types on a daily basis in order to bridge the information to an interfaced financial application. The program reads from the IF_TRAN_DATA table for each transaction type/amount type and posts it to the Oracle Retail General Ledger staging table at the SKU detail level.

If transactions exist for which GL cross mappings do not exist, these will be logged in TRAN_DATA_ERRORS and a notification will be sent to the Finance Analyst user indicating that unmapped transactions exist. The TRAN_DATA_ERRORS table is available through the Data Access Schema, enabling users to view the errors and create the missing mappings. The figlgn1 module will attempt to reprocess records from TRAN_DATA_ERRORS if mappings have been created. During the month-end processing run, if unmapped transactions still exist they will be posted into the clearing accounts as outlined below:

1. The cost value of the transaction will be posted into the Credit Clearance account [Credit Clearance Segment 1-10] associated with the location's set of books.
2. The cost value of the transaction will be posted into the Debit Clearance account [Debit Clearance Segment 1-10] associated with the location's set of books.
3. The retail value of the transaction will be posted into the Credit Clearance account [Credit Clearance Segment 1-10] associated with the location's set of books.
4. The retail value of the transaction will be posted into the Debit Clearance account [Debit Clearance Segment 1-10] associated with the location's set of books.

The figlgn1 module will fail during month-end processing if unmapped transactions exist and clearance accounts haven't been defined.

Restart/Recovery

The logical unit of work is department/class/subclass. The batch is multithreaded using the restart department view.

I/O Specification

Integration Type	Download from Merchandising
File Name	N/A
Integration Contract	IntCon000019 STG_FIF_GL_DATA table

Design Assumptions

N/A

Monthly Stock Ledger Transactions (fifgldn3)

Module Name	fifgldn3.pc
Description	General Ledger Interface 3
Functional Area	Interface to General Ledger of Month Level Information
Module Type	Integration
Module Technology	ProC
Catalog ID	RMS68
Wrapper Script	rmswrap_multi.ksh

Schedule

Oracle Retail Merchandising Batch Schedule

Design Overview

This program summarizes stock ledger data from the monthly stock ledger table based on the level of information required and writes it to the financial general ledger staging table. The transactions extracted are determined by the CODE_TYPE 'GLRT' (general ledger rolled transactions). Written information is then sent to the financial application. Stock ledger information may be rolled-up at department, class or subclass level. The level at which information is rolled-up to is determined by the system parameter GL_ROLLUP.

The fifgldn3 module is scheduled to run monthly and operates off both the TRAN_DATA and MONTH_DATA tables. If the module encounters an unmapped transaction, depending on whether the source of the unmapped transaction is TRAN_DATA or MONTH_DATA, the transaction will be logged into TRAN_DATA_ERRORS or MONTH_DATA_ERRORS tables, respectively. If clearing accounts have been defined, these unmapped transactions will be posted into the clearing accounts as outlined below:

1. The cost value of the transaction will be posted into the Credit Clearance account [Credit Clearance Segment 1-10] associated with the location's set of books.
2. The cost value of the transaction will be posted into the Debit Clearance account [Debit Clearance Segment 1-10] associated with the location's set of books.
3. The retail value of the transaction will be posted into the Credit Clearance account [Credit Clearance Segment 1-10] associated with the location's set of books.
4. The retail value of the transaction will be posted into the Debit Clearance account [Debit Clearance Segment 1-10] associated with the location's set of books.

A notification will be sent to the Finance Analyst user indicating that unmapped transactions exist and whether the postings to the clearance accounts have occurred. The fifgldn3 module will fail during month-end processing if unmapped transactions exist and clearance accounts haven't been defined.

Restart/Recovery

The logical unit of work is dependent on the level of rollup defined in the system options table. It can be department (department rollup), department/class (class rollup) or department/class/subclass (subclass rollup). The batch is multithreaded using the restart all locations view.

I/O Specification

Integration Type	Download from Merchandising
File Name	N/A
Integration Contract	IntCon000019 STG_FIF_GL_DATA table

Design Assumptions

N/A

Open to Buy Download Stock Ledger (otbdlsal)

Module Name	otbdlsal.pc
Description	Open To Buy Download Stock Ledger
Functional Area	OTB - Stock Ledger to Planning System Interface
Module Type	Integration
Module Technology	ProC
Catalog ID	RMS16
Wrapper Script	Rmswrap_out.ksh

Schedule

Oracle Retail Merchandising Batch Schedule

Design Overview

This module will sum stock ledger data from the DAILY_DATA table and opening stock information from the WEEK_DATA table across the current week, grouping by department, class, subclass, location and date, and export the data to a flat file for use by an outside planning system.

Restart/Recovery

The logical unit of work for the OTBDLSAL module is department, class, subclass and location. The commit_max_ctr field should be set to prevent excessive rollback space usage, and to reduce the overhead of the file I/O. The recommended commit counter setting is 10000 records. Each time the record counter equals the maximum recommended commit number, an application image array record will be written to the restart_start_array for restart/recovery if a fatal error occurs.

Locking Strategy

N/A

Security Considerations

N/A

Performance Considerations

N/A

I/O Specification

Integration Type	Download from Merchandising
File Name	Determined by runtime parameter
Integration Contract	OTB - Stock Ledger to Planning System Interface IntCon00030

Output File Format

Table 6-10 File Layout

Record Name	Field Name	Field Type	Default Value	Description
FHEAD	File Type Record Descriptor	Char(5)	FHEAD	Identifies file record type
	File Line Sequence Number	Number(10)	0000000001	Keeps track of the record's position in the file by line number
	File Type Definition	Char(4)	STKE	Identifies file as Stock Ledger Export
	File Create Date	Char(14)	vdate	Date file was written by batch program in YYYYMMDD format. Remaining six characters are blank.
FDETL	File Type Record Descriptor	Char(5)	FDETL	Identifies file record type
	File Line Sequence Number	Number(10)	line number in file	Keeps track of the record's position in the file by line number

Table 6-10 (Cont.) File Layout

Record Name	Field Name	Field Type	Default Value	Description
	Transaction Set Control Number	Number(14)	sequence number	Used to force unique file check
	Department	Number(4)	N/A	The ID number of a department
	Class	Number(4)	N/A	The ID number of a class within the department given
	Subclass	Number(4)	N/A	The ID number of a subclass within the class given
	Loc_type	Char(1)	N/A	The type of the location from which stock ledger data was collected
	Location	Number(10)	N/A	The location from which stock ledger data was collected
	Half No.	Number(5)	N/A	The half number for this stock ledger data
	Month No.	Number(2)	N/A	The month number in the half for this stock ledger data
	Week No.	Number(2)	N/A	The week number in the month for this stock ledger data
	Open Stock Retail	Number(20,4)	N/A	The retail opening stock from the week_data table *10000 (implied 4 decimal places) for this stock ledger period
	Open Stock Cost	Number(20,4)	N/A	The cost opening stock from the week_data table *10000 (implied 4 decimal places) for this stock ledger period

Table 6-10 (Cont.) File Layout

Record Name	Field Name	Field Type	Default Value	Description
	Stock Adjustments Retail	Number(20,4)	N/A	The retail stock adjustments summed from the DAILY_DATA table *10000 (implied 4 decimal places) for this stock ledger period
	Stock Adjustments Cost	Number(20,4)	N/A	The cost stock adjustments summed from the DAILY_DATA table *10000 (implied 4 decimal places) for this stock ledger period
	Purchases Retail	Number(20,4)	N/A	The retail purchases summed from the DAILY_DATA table *10000 (implied 4 decimal places) for this stock ledger period
	Purchases Cost	Number(20,4)	N/A	The cost purchases summed from the DAILY_DATA table *10000 (implied 4 decimal places) for this stock ledger period
	RTV Retail	Number(20,4)	N/A	The retail return to vendor amount summed from the DAILY_DATA table *10000 (implied 4 decimal places) for this stock ledger period

Table 6-10 (Cont.) File Layout

Record Name	Field Name	Field Type	Default Value	Description
	RTV Cost	Number(20,4)	N/A	The cost return to vendor amount summed from the DAILY_DATA table *10000 (implied 4 decimal places) for this stock ledger period
	Freight Cost	Number(20,4)	N/A	The freight cost summed from the DAILY_DATA table *10000 (implied 4 decimal places) for this stock ledger period
	Net Sales Retail	Number(20,4)	N/A	The retail net sales summed from the DAILY_DATA table *10000 (implied 4 decimal places) for this stock ledger period
	Net Sales Cost	Number(20,4)	N/A	The cost net sales summed from the DAILY_DATA table *10000 (implied 4 decimal places) for this stock ledger period
	Returns Retail	Number(20,4)	N/A	The retail returns amount summed from the DAILY_DATA table *10000 (implied 4 decimal places) for this stock ledger period

Table 6-10 (Cont.) File Layout

Record Name	Field Name	Field Type	Default Value	Description
	Returns Cost	Number(20,4)	N/A	The cost returns amount summed from the DAILY_DATA table *10000 (implied 4 decimal places) for this stock ledger period
	Promotional Markdowns Retail	Number(20,4)	N/A	The retail promotional markdowns summed from the DAILY_DATA table *10000 (implied 4 decimal places) for this stock ledger period
	Markdown Cancellations Retail	Number(20,4)	N/A	The retail markdown cancellations summed from the DAILY_DATA table *10000 (implied 4 decimal places) for this stock ledger period
	Employee Discount Retail	Number(20,4)	N/A	The retail employee discounts amount summed from the DAILY_DATA table *10000 (implied 4 decimal places) for this stock ledger period
	Workroom Amount	Number(20,4)	N/A	The workroom amount summed from the DAILY_DATA table *10000 (implied 4 decimal places) for this stock ledger period

Table 6-10 (Cont.) File Layout

Record Name	Field Name	Field Type	Default Value	Description
	Cash Discount Amount	Number(20,4)	N/A	The cash discounts amount summed from the DAILY_DATA table *10000 (implied 4 decimal places) for this stock ledger period
	Sales Units	Number(12,4)	N/A	The sales units summed from the DAILY_DATA table *10000 (implied 4 decimal places) for this stock ledger period
	Markups Retail	Number(20,4)	N/A	The retail markups summed from the DAILY_DATA table *10000 (implied 4 decimal places) for this stock ledger period
	Markup Cancellations Retail	Number(20,4)	N/A	The retail markup cancellations summed from the DAILY_DATA table *10000 (implied 4 decimal places) for this stock ledger period
	Clearance Markdowns Retail	Number(20,4)	N/A	The retail clearance markdowns summed from the DAILY_DATA table *10000 (implied 4 decimal places) for this stock ledger period

Table 6-10 (Cont.) File Layout

Record Name	Field Name	Field Type	Default Value	Description
	Permanent Markdowns Retail	Number(20,4)	N/A	The retail permanent markdowns summed from the DAILY_DATA table *10000 (implied 4 decimal places) for this stock ledger period
	Freight Claim Retail	Number(20,4)	N/A	The retail freight claim summed from the DAILY_DATA table *10000 (implied 4 decimal places) for this stock ledger period
	Freight Claim Cost	Number(20,4)	N/A	The cost freight claim summed from the DAILY_DATA table *10000 (implied 4 decimal places) for this stock ledger period
	Stock Adjust Cost of Goods Sold (COGS) Retail	Number(20,4)	N/A	The retail stock adjust COGS summed from the DAILY_DATA table *10000 (implied 4 decimal places) for this stock ledger period
	Stock Adjust Cost of Goods Sold (COGS) Cost	Number(20,4)	N/A	The cost stock adjust COGS summed from the DAILY_DATA table *10000 (implied 4 decimal places) for this stock ledger period

Table 6-10 (Cont.) File Layout

Record Name	Field Name	Field Type	Default Value	Description
	Inter-company In Retail	Number(20,4)	N/A	The Inter-company In retail summed from the DAILY_DATA table *10000 (implied 4 decimal places) for this stock ledger period
	Inter-company In Cost	Number(20,4)	N/A	The Inter-company In cost summed from the DAILY_DATA table *10000 (implied 4 decimal places) for this stock ledger period
	Inter-company Out Retail	Number(20,4)	N/A	The Inter-company Out Retail summed from the DAILY_DATA table *10000 (implied 4 decimal places) for this stock ledger period
	Inter-company Out Cost	Number(20,4)	N/A	The Inter-company Out Cost summed from the DAILY_DATA table *10000 (implied 4 decimal places) for this stock ledger period
	Inter-company Markup	Number(20,4)	N/A	The Inter-company Markup summed from the DAILY_DATA table *10000 (implied 4 decimal places) for this stock ledger period

Table 6-10 (Cont.) File Layout

Record Name	Field Name	Field Type	Default Value	Description
	Inter-company Markdown	Number(20,4)	N/A	The Inter-company Markdown summed from the DAILY_DATA table *10000 (implied 4 decimal places) for this stock ledger period
	Work Order Activity Update Inventory	Number(20,4)	N/A	The Work Order Activity Update Inventory summed from the DAILY_DATA table *10000 (implied 4 decimal places) for this stock ledger period
	Work Order Activity Post Finishing	Number(20,4)	N/A	The Work Order Activity Post Finishing summed from the DAILY_DATA table *10000 (implied 4 decimal places) for this stock ledger period
FTAIL	File Type Record Descriptor	Char(5)	FTAIL	Identifies file record type
	File Line Sequence Number	Number(10)	N/A	Keeps track of the record's position in the file by line number
	Control Number File Line Count	Number(10)	N/A	Total number of all transaction lines, not including file header and trailer

Design Assumptions

N/A

Rolled Up Daily Stock Ledger Transactions (fifgldn2)

Module Name fifgldn2.pc

Description	Interface to General Ledger of Rolled Up Transactions
Functional Area	Integration - General Ledger
Module Type	Integration
Module Technology	ProC
Catalog ID	RMS67
Wrapper Script	rmswrap_multi.ksh

Schedule

Oracle Retail Merchandising Batch Schedule

Design Overview

This program summarizes stock ledger data from the transaction staging table (IF_TRAN_DATA) based on the level of information required and writes it to the financial general ledger staging table. The transactions extracted are determined by the CODE_TYPE 'GLRT' (General Ledger Rolled Transactions). The written information can then be extracted by the financial applications. Stock ledger information may be rolled-up at department, class or subclass level. The level at which information is rolled-up to is determined by the system parameter GL_ROLLUP.

If transactions exist for which GL cross mappings do not exist, these will be logged in TRAN_DATA_ERRORS and a notification will be sent to the Finance Analyst user indicating that unmapped transactions exist. The TRAN_DATA_ERRORS table is available through the Data Access Schema, enabling users to view the errors and create the missing mappings. The figlnd2 module will attempt to reprocess records from TRAN_DATA_ERRORS if mappings have been created. During the month-end processing run, if unmapped transactions still exist they will be posted into the clearing accounts as outlined below:

1. The cost value of the transaction will be posted into the Credit Clearance account [Credit Clearance Segment 1-10] associated with the location's set of books.
2. The cost value of the transaction will be posted into the Debit Clearance account [Debit Clearance Segment 1-10] associated with the location's set of books.
3. The retail value of the transaction will be posted into the Credit Clearance account [Credit Clearance Segment 1-10] associated with the location's set of books.
4. The retail value of the transaction will be posted into the Debit Clearance account [Debit Clearance Segment 1-10] associated with the location's set of books.

The figlnd2 module will fail during month-end processing if unmapped transactions exist and clearance accounts haven't been defined.

Restart/Recovery

The logical unit of work is dependent on the level of rollup defined in the system options table. It can be department (department rollup), department/class (class rollup) or department/class/subclass (subclass rollup). The batch is multithreaded using the restart department view.

I/O Specification

Integration Type	Download from Merchandising
File Name	N/A
Integration Contract	IntCon000019 STG_FIF_GL_DATA table

Design Assumptions

N/A

Stage G/L Extracts (gl_extract.ksh)

Module Name	gl_extract.ksh
Description	Extraction of General Ledger transaction data from Merchandising and Sales Audit to be interfaced to third party GL/Financial system
Functional Area	Integration to General Ledger
Module Type	Integration
Module Technology	ksh
Catalog ID	RMS495
Wrapper Script	rmswrap_shell_out.ksh

Schedule

Oracle Retail Merchandising Batch Schedule

Design Overview

This batch job will extract general ledger transaction data from Sales Audit and Merchandising into a file. Data to be extracted will be pulled off from the STG_FIF_GL_DATA table. Once the data is extracted into the file batch will purge the data from the table.

Restart/Recovery

N/A

I/O Specification

Integration Type	Extract from Merchandising
File Name	GL_EXTRACT_[#date].dat
Integration Contract	Na

Output File Layout

The output file is comma delimited with the following fields:

Record Name	Field Name
All records have the same structure	SET_OF_BOOKS_ID
	ACCOUNTING_DATE
	CURRENCY_CODE
	STATUS
	DATE_CREATED
	CREATED_BY
	ACTUAL_FLAG
	USER_JE_CATEGORY_NAME
	USER_JE_SOURCE_NAME
	CURRENCY_CONVERSION_DATE
	CURRENCY_CONVERSION_TYPE
	ACCT_SEGMENT1
	ACCT_SEGMENT2
	ACCT_SEGMENT3
	ACCT_SEGMENT4
	ACCT_SEGMENT5
	ACCT_SEGMENT6
	ACCT_SEGMENT7
	ACCT_SEGMENT8
	ACCT_SEGMENT9
	ACCT_SEGMENT10
	ENTERED_DR_AMOUNT
	ENTERED_CR_AMOUNT
	TRANSACTION_DATE
	REFERENCE1
	REFERENCE2
	REFERENCE3
	REFERENCE4
	REFERENCE5
	ATTRIBUTE1
	ATTRIBUTE2
ATTRIBUTE3	
ATTRIBUTE4	
ATTRIBUTE5	
ATTRIBUTE6	
PERIOD_NAME	
CODE_COMBINATION_ID	
PGM_NAME	
ACCT_SEGMENT11	

Record Name	Field Name
	ACCT_SEGMENT12
	ACCT_SEGMENT13
	ACCT_SEGMENT14
	ACCT_SEGMENT15
	ACCT_SEGMENT16
	ACCT_SEGMENT17
	ACCT_SEGMENT18
	ACCT_SEGMENT19
	ACCT_SEGMENT20
	REFERENCE_TRACE_ID
	PRIM_CURRENCY_CODE
	PRIM_ENTERED_DR_AMOUNT
	PRIM_ENTERED_CR_AMOUNT
	FIN_GL_SEQ_ID
	PROCESSED_FLAG

Design Assumptions

N/A

Tran Data Publication (BDI_TranData_Tx_PF_From_RMS_EOW_JOB)

This section describes the Tran Data Publication BDI.

Functional Area

Transactional Data

Business Overview

BDI (Bulk Data Integration) is an integration layer that facilitates the bulk transfer of transactional data from Merchandising to other Oracle Retail Applications. On this particular integration stream, the data flow is from Merchandising to BDI, and then BDI to downstream applications. To accomplish this data transfer, BDI will be calling a Merchandising-owned API that will pull data from Merchandising and deliver these to the BDI integration layer. This API will be in the form of a PLSQL function inside a PLSQL package.

Package Impact

Filename: bdimfpb.pls

```
BDI_MFP_SQL.TRAN_DATA_UP(O_error_message IN OUT
RTK_ERRORS.RTK_TEXT%TYPE, O_control_id IN OUT
NUMBER, I_job_context IN VARCHAR2)
```

This function begins by calling a BDI function that signals the start of the interface process. The BDI function will update the internal BDI control tables to track the progress of the API.

A DML insert statement is then executed to populate the BDI outbound table that resides in the BDI_RMS_INT_SCHEMA schema. This outbound table is loaded with records from the Merchandising transaction tables/views.

After the insert, another call to a BDI function is performed to signify the successful loading of records. This will update the internal BDI control tables.

A database commit is issued, and the control Id is returned by the API.

Data Definition XML

The BDI interface staging tables are generated based on the XML schema definition

Data Flow	Description	XML Schema Definition (XSD)
Tran Data	Tran Data upload to BDI	TranData_Tx_BdiInterfaceModule.xml

Table Impact

TABLE	SELECT	INSERT	UPDATE	DELETE
TRAN_DATA_OUT	No	Yes	No	No
V_BDI_MFP_TRAN_DATA	Yes	No	No	No

Ordering and Inventory

Merchandising publishes purchase order and inventory-related for many other solution areas, including purchase orders, import details, invoices, available inventory, and other inventory related data. This section has been broken down into subsections for:

- [Purchasing](#)
- [Import Management](#)
- [Invoices](#)
- [Inventory](#)

Purchasing

Merchandising has scheduled integration for the following purchasing related data:

- [Download Contracts to Suppliers \(edidlcon\)](#)
- [Download Current & Future OTB by Subclass \(otbdnid\)](#)
- [Download Purchase Orders to Suppliers \(edidlord\)](#)
- [Download Summary of Outstanding Orders on OTB by Subclass \(otbdlord\)](#)
- [On Order Publication API \(BDI_OnOrder_Tx_PF_From_RMS_EOW_JOB\)](#)
- [Replenishment Item Location Publication API \(BDI_ReplItemLoc_Fnd_PF_From_RMS_JOB\)](#)

Download Contracts to Suppliers (edidlcon)

Module Name edidlcon.pc

Description	Download Contracts to Suppliers
Functional Area	Contracts
Module Type	Integration
Module Technology	ProC
Catalog ID	RMS45
Wrapper Script	rmswrap_out.ksh

Schedule

Oracle Retail Merchandising Batch Schedule

Design Overview

Contracts are defined in an Merchandising UI that writes to series of contracts database tables. This program is used to send this contract information to vendors. Only approved contracts that are flagged as EDI contracts are processed by this batch program. The output file of this program contains all records for the supplier contract data which are in approved status.

Restart/Recovery

The logical unit of work for this program is set at the contract number. This program processes one contract number at a time.

I/O Specification

Integration Type	Upload to Merchandising
File Name	Determined by runtime parameter
Integration Contract	IntCon000011

Output File Layout

Table 6-11 edidlcon.pc- File Layout

Record Name	Field Name	Field Type	Default Value	Description
FHEAD	File head descriptor	Char(5)	FHEAD	Describes file line type
	Line Number	Number(10)	000000001	Sequential file line number
	Gentran ID	Char(4)	'DNCN'	Identifies which translation Gentran uses
	Current date	Char(14)	N/A	Indicates the date that the file was created in YYYYMMDDH H24MISS format

Table 6-11 (Cont.) edidlcon.pc- File Layout

Record Name	Field Name	Field Type	Default Value	Description
THEAD	File head descriptor	Char(5)	THEAD	Describes file line type
	Line Number	Number(10)	N/A	Sequential file line number
	Transaction Number	Number(10)	N/A	Sequential transaction number
	Supplier	Number(10)	N/A	Indicates the supplier associated with the contract
	Contract Number	Number(6)	N/A	Indicates the Merchandising contract number
	Contract type	Char(1)	N/A	Type of contract. Valid types are A, B, C or D
	Department	Number(4)	N/A	Indicates the Merchandising department ID for which the contract applies
	Currency code	Char(3)	N/A	Indicates the currency code for the contract
	Total contract cost	Number(20)	N/A	Contains the total cost of the contract; includes 4 implied decimal places
TDETL	File record descriptor	Char(5)	TDETL	Describes file line type
	Line Number	Number(10)	N/A	Sequential file line number
	Transaction number	Number(10)	N/A	Sequential transaction number

Table 6-11 (Cont.) edidlcon.pc- File Layout

Record Name	Field Name	Field Type	Default Value	Description
	Item Number Type	Char(6)	N/A	Indicates the type of item number is represented in the file. This corresponds to the item number type defined for items on ITEM_MASTER
	Item Number	Char(25)	N/A	Contains the unique ID for the item on the contract
	Ref Item Number Type	Char(6)	N/A	Indicates the item number type for the reference number corresponding to the item number
	Ref Item Number	Char(25)	N/A	Contains the unique ID for the reference number for the item
	Diff1	Char(120)	N/A	Contains the description of Diff1 for the item
	Diff2	Char(120)	N/A	Contains the description of Diff2 for the item
	Diff3	Char(120)	N/A	Contains the description of Diff3 for the item
	Diff4	Char(120)	N/A	Contains the description of Diff4 for the item
	VPN	Char(30)	N/A	Vendor Product Number for the item

Table 6-11 (Cont.) edidlcon.pc- File Layout

Record Name	Field Name	Field Type	Default Value	Description
	Unit cost	Number(20)		Contains the cost of the item on the contract with 4 implied decimal places
	Ready Date	Char(14)		Date on which the items are to be provided by supplier. This field contains only values for contract types of 'A' or 'B'
	Ready Quantity	Number(20)		Quantity contracted with supplier with 4 implied decimal points. This field contains only values for contract types of 'A' or 'B'
	Location Type	Char(2)		Indicates the type of location on the contract - either 'ST' (store) or 'WH' (warehouse). This field contains only values for contract types of 'A' or 'B'
	Location number	Number(10)		Contains a location on the contract. This field contains only values for contract types of 'A' or 'B'
TTAIL	File Record descriptor	Char(5)	TTAIL	Describes file line type
	Line Number	Number(10)	N/A	Sequential file line number
	Transaction number	Number(10)	N/A	Sequential transaction number
FTAIL	File record descriptor	Char(5)	FTAIL	Marks the end of file
	Line number	Number(10)	N/A	Sequential file line number

Table 6-11 (Cont.) edidlcon.pc- File Layout

Record Name	Field Name	Field Type	Default Value	Description
	Number of lines	Number(10)	N/A	Number of lines in file not counting FHEAD and FTAIL

Design Assumptions

- This module should only be run if contracting is turned on in the system.

Download Current & Future OTB by Subclass (otbdnld)

Module Name	otbdnld.pc
Description	Download Current & Future OTB by Subclass
Functional Area	Open To Buy
Module Type	Integration
Module Technology	ProC
Catalog ID	RMS130
Wrapper Script	rmswrap_out.ksh

Schedule

Oracle Retail Merchandising Batch Schedule

Design Overview

This batch program will extract current and future Open to Buy data from the OTB table in Merchandising and export it to a flat file for use by an external planning system. All records with an end of week date greater than or equal to today will be sent.

Restart/Recovery

The logical unit of work for the OTBDNLD module is department, class, subclass, and end-of-week date, with a recommended commit counter setting of 10,000. Each time the record counter equals the maximum recommended commit number, an application image array record will be written to the restart_start_array for restart/recovery if a fatal error occurs.

I/O Specification

Integration Type	Download from Merchandising
File Name	Determined by runtime parameter
Integration Contract	IntCon000031

Output File Layout

Table 6-12 otbdnld.pc - Output File

Record Name	Field Name	Field Type	Default Value	Description
FHEAD	File Type Record Descriptor	Char (5)	FHEAD	Identifies file record type
	File Line Sequence Number	Number (10)	N/A	Keeps track of the record's position in the file by line number
	File Type Definition	Char (4)	N/A	Identifies file as 'OTB Export'
	File Create Date	Char(14)	N/A	Date the file was created in YYYYMMDD format. Remaining 6 characters are blank
FDETL	File record descriptor	Char(5)	FDETL	Identifies file record type
	File Line Sequence Number	Number (10)		Keeps track of the record's position in the file by line number
	Transaction Set Control Number	Number(14)		Used to force unique file check
	Department	Number(4)		The ID number of a department
	Class	Number(4)		The ID number of a class within the department given
	Subclass	Number(4)		The ID number of a subclass within the class given
	EOW Date	Date		The end of week date for the budgeted period. Format is 'YYYYMMDD HHMMSS'

Table 6-12 (Cont.) otbdnld.pc - Output File

Record Name	Field Name	Field Type	Default Value	Description
	Week number	Number(2)		The week number in the month for the budgeted period
	Month number	Number(2)		The month number in the half for the budgeted period
	Half number	Number(5)		The half number for the budgeted period
	Cancel Amount	Number(20)		The total amount cancelled from orders of all order type for the budgeted period; value includes 4 implied decimal places
	N Approved Amount	Number(20)		The amount of approved non-basic (order type N/B) orders for the budgeted period; value includes 4 implied decimal places
	N Receipts Amount	Number(20)		The amount of non-basic (order type N/B) orders due in the budgeted period that have been received; value includes 4 implied decimal places

Table 6-12 (Cont.) otbdnld.pc - Output File

Record Name	Field Name	Field Type	Default Value	Description
	B Approved Amount	Number(20)		The amount of approved buyer-replenished basic (order type BRB) orders for the budgeted period; value includes 4 implied decimal places
	B Receipts Amount	Number(20)		The amount of buyer-replenished basic (order type BRB) orders due in the budgeted period that have been received; value includes 4 implied decimal places
	A Approved Amount	Number(20)		The amount of approved auto-replenished basic (order type ARB) orders for the budgeted period; value includes 4 implied decimal places
	A Receipts Amount	Number(20)		The amount of auto-replenished basic (order type ARB) orders due in the budgeted period that have been received; value includes 4 implied decimal places

Table 6-12 (Cont.) otbdnld.pc - Output File

Record Name	Field Name	Field Type	Default Value	Description
FTAIL	File record descriptor	Char (5)	FTAIL	Identifies file record type
	File Line Sequence Number	Number (10)		Keeps track of the record's position in the file by line number
	Number of lines	Number (10)		Total number of all transaction lines, not including file header and trailer

Design Assumptions

N/A

Download Purchase Orders to Suppliers (edidlord)

Module Name	edidlord.pc
Description	Download of Purchase Order from Merchandising to Suppliers
Functional Area	Purchase Order
Module Type	Integration
Module Technology	ProC
Catalog ID	RMS46
Wrapper Script	rmswrap_multi_out.ksh

Schedule

Oracle Retail Merchandising Batch Schedule

Design Overview

Orders created within the Oracle Retail system are written to a flat file if they are approved and marked as EDI orders. This module is used to write new and changed purchase order data to a flat file in the Oracle Retail standard format. The translation to EDI format is expected to take place via a 3rd party translation utility. The order revision tables and allocation revision tables are also used to ensure that the latest changes are being sent and to allow both original and modified values to be sent. These revision tables are populated during the online ordering process and the batch replenishment process whenever an order has been approved, and constitutes a history of all revisions to the order.

The program sums up all quantities to the physical warehouse level from the virtual warehouse level for an order, before writing it into the output file.

If shipments are to be pre-marked by the supplier for cross docking, then along with the order information: allocation, location and quantities are also sent.

If the backhaul type is specified as "Calculated", then the backhaul allowances will be calculated.

If the order contains pack items; hierarchical pack information is sent (this may include outer packs, inner packs, and fashion styles with associated pack templates as well as component item information).

If the order is a Drop Ship Customer Order (location is a non-stockholding store), the customer billing and delivery information will be written to the flat file.

Restart/Recovery

The logical unit of work for this program is set at the supplier level. Threading is performed by the supplier using the v_restart_supplier view.

Restart ability is implied because the program updates ordhead.edi_sent_ind as records and are written out. The commit_max_ctr field should be set to prevent excessive rollback space usage, and to reduce the overhead of the file I/O. The recommended commit counter setting is 10000 records.

I/O Specification

Integration Type	Download from Merchandising
File Name	Determined by runtime parameter
Integration Contract	IntCon000012

Output File Layout

Table 6-13 File Layout

Record Name	Field Name	Field Type	Default Value	Description
FHEAD	Record descriptor	Char(5)	FHEAD	File head marker
	Line id	Number(10)	0000000001	Unique line id
	Translator id	Char(5)	DLORD	Identifies transaction type
	File create date	Char(14)	N/A	Vdate in YYYYMMDDHH24MISS format
TORDR	Record descriptor	Char(5)	TORDR	Order header information
	Line id	Number(10)	N/A	Unique file line id
	Transaction id	Number(10)	N/A	Unique transaction id
	Order change type	Char(2)	N/A	'CH' (changed) or 'NW' (new)
	Order number	Number(12)	N/A	Internal Oracle Retail order no
	Supplier	Number(10)	N/A	Internal Oracle Retail supplier id

Table 6-13 (Cont.) File Layout

Record Name	Field Name	Field Type	Default Value	Description
	Vendor order id	Char(15)	N/A	External vendor_order_no (if available)
	Order written date	Char(14)	N/A	Order created date in YYYYMMDDHH24MISS format
	Original order approval date	Char(14)	N/A	Original order approval date in YYYYMMDDHH24MISS format
	Old Currency Code	Char(3)	N/A	Old order currency_code (ISO standard)
	New Currency Code	Char(3)	N/A	Changed order currency_code (ISO standard)
	Old Shipment Method of Payment	Char(2)	N/A	Old ship_pay_method
	New Shipment Method of Payment	Char(2)	N/A	Changed ship_pay_method
	Old Transportation Responsibility	Char(2)	N/A	Old fob_trans_res
	Old Transportation Responsibility Description	Char(250)	N/A	Old fob_trans_res_desc
	New Transportation Responsibility	Char(2)	N/A	Changed fob_trans_res
	New Trans. Resp. Description	Char(250)	N/A	New fob_trans_res_desc
	Old Title Passage Location	Char(2)	N/A	Old fob_title_pass
	New Title Passage Location	Char(2)	N/A	Changed fob_title_pass
	Old Title Passage Description	Char(250)	N/A	Old fob_title_pass_desc
	New Title Passage Description	Char(250)	N/A	Changed fob_title_pass_desc
	Old not before date	Char(14)	N/A	Old not_before_date in YYYYMMDDHH24MISS format

Table 6-13 (Cont.) File Layout

Record Name	Field Name	Field Type	Default Value	Description
	New not before date	Char(14)	N/A	Changed not_before_date in YYYYMMDDHH24MISS format
	Old not after date	Char(14)	N/A	Old not_after_date in YYYYMMDDHH24MISS format
	New not after date	Char(14)	N/A	Changed not_after_date in YYYYMMDDHH24MISS format
	Old Purchase type	Char(6)	N/A	Old Purchase type
	New Purchase type	Char(6)	N/A	New Purchase type
	Backhaul allowance	Char(20)	N/A	Backhaul allowance
	Old terms description	Char(240)	N/A	Old terms description from terms table
	New terms description	Char(240)	N/A	New terms description from terms table
	Old pickup date	Char(14)	N/A	Old pickup date YYYYMMDDHH24MISS
	New pickup date	Char(14)	N/A	New pickup date YYYYMMDDHH24MISS
	Old ship method	Char(6)	N/A	Old ship method
	New ship method	Char(6)	N/A	New ship method
	Old comment description	Char(2000)	N/A	Old comment description
	New comment description	Char(2000)	N/A	New comment description
	Supplier DUNS number	Char(9)	N/A	Supplier DUNS number
	Supplier DUNS location	Char(4)	N/A	Supplier DUNS location
	Customer order number	Char(48)	N/A	Master customer order number from the Order Management System
TITEM	File record descriptor	Char(5)	TITEM	Item info
	Line id	Number(10)	N/A	Unique line id
	Transaction id	Number(10)	N/A	Unique transaction id
	Item Number Type	Char(6)	N/A	Item_number_type

Table 6-13 (Cont.) File Layout

Record Name	Field Name	Field Type	Default Value	Description
	Item	Char(25)	N/A	Item (For a pack item, this will be the pack number)
	Old Ref Item Number type	Char(6)	N/A	Item_number_type for old ref_item
	Old Ref Item	Char(25)	N/A	Old Ref_Item
	New Ref Item Number type	Char(6)	N/A	Item_number_type for new ref_item
	New Ref Item	Char(25)	N/A	Changed Ref_Item
	Vendor catalog number	Char(30)	N/A	Supplier_item (VPN)
	Free Form Description	Char(250)	N/A	Item_desc
	Supplier Diff 1	Char(120)	N/A	Supplier's diff 1
	Supplier Diff 2	Char(120)	N/A	Supplier's diff 2
	Supplier Diff 3	Char(120)	N/A	Supplier's diff 3
	Supplier Diff 4	Char(120)	N/A	Supplier's diff 4
	Pack Size	Number(12)	N/A	Supplier defined pack size * 10000 (4 implied decimal places)
	Item line no	Number(10)	N/A	Indicates the detail item line number.
TPACK	File record descriptor	Char(5)	TPACK	Pack component info
	Line id	Number(10)	N/A	Unique line id
	Transaction id	Number(10)	N/A	Unique transaction id
	Pack id	Char(25)	N/A	Packitem_breakout.pack_no (same as item for the pack item)
	Inner pack id	Char(25)	N/A	Inner pack identification
	Pack Quantity	Number(12)	N/A	Packitem_breakout.pack_item_qty*10000 (4 implied decimal places)
	Component Pack Quantity	Number(12)	N/A	Packitem_breakout.comp_pack_qty*10000 (4 implied decimal places)
	Item Parent Part Quantity	Number(12)	N/A	Packitem_breakout.item_parent_pt_qty*10000 (4 implied decimal places)
	Item Quantity	Number(12)	N/A	Packitem_breakout.item_qty*10000 (4 implied decimal places)
	Item Number Type	Char(6)	N/A	Item number type

Table 6-13 (Cont.) File Layout

Record Name	Field Name	Field Type	Default Value	Description
	Item	Char(25)	N/A	Item
	Ref Item Number Type	Char(6)	N/A	Ref_item_number_type
	Ref Item	Char(25)	N/A	Ref_item
	VPN	Char(30)	N/A	Supplier item (vpn)
	Supplier Diff 1	Char(120)	N/A	Supplier's diff 1
	Supplier Diff 2	Char(120)	N/A	Supplier's diff 2
	Supplier Diff 3	Char(120)	N/A	Supplier's diff 3
	Supplier Diff 4	Char(120)	N/A	Supplier's diff 4
	Item Parent	Char(25)	N/A	Required when Pack Template is not NULL
	Pack template	Number(8)	N/A	Pack template associated w/ style (packitem_breakout.pack_tmpl_id)
	Template description	Char(250)	N/A	Description of pack template. sups_pack_tmpl_desc.supp_pack_desc
TSHIP	Record type	Char(5)	TSHIP	Describes the file record-shipment information
	Line id	Number(10)	N/A	Unique file line number
	Transaction id	Number(10)	N/A	Unique transaction number
	Location type	Char(2)	N/A	'ST' store or 'WH' warehouse
	Ship to location	Number(10)	N/A	Location value form ordloc (store or warehouse – For warehouse,if multichannel option is ON, physical warehouse value is taken from warehouse)
	Old unit cost	Number(20)	N/A	Old unit cost*10000 (4 implied decimal places)
	New unit cost	Number(20)	N/A	New unit cost*10000 (4 implied decimal places)
	Old quantity	Number(12)	N/A	Old qty_ordered *10000 or qty_allocated*10000 (4 implied decimal places)
	New quantity	Number(12)	N/A	Changed qty_ordered*10000 or qty_allocated*10000 (4 implied decimal places)
	Old outstanding quantity	Number(12)	N/A	Old (qty_ordered-qty_received)*10000 or (qty_allocated-qty transferred)*10000 for an allocation (4 implied decimal places)

Table 6-13 (Cont.) File Layout

Record Name	Field Name	Field Type	Default Value	Description
	New outstanding quantity	Number(12)	N/A	Changed qty_ordered-qty_received (4 implied decimal places)(or qty_allocated-qty_transferred, for an allocation)
	Cancel code	Char(1)	N/A	N/A
	Old cancelled quantity	Number(12)	N/A	Previous quantity cancelled (4 implied decimal places)
	New cancelled quantity	Number(12)	N/A	Changed quantity cancelled (4 implied decimal places)
	Quantity type flag	Char(1)	N/A	'S'hip to 'A'locate
	Store or warehouse indicator	Char(2)	N/A	'ST' (store) or 'WH' (warehouse)
	Old x-dock location	Number(10)	N/A	Alloc_detail location (store or wh)
	New x-dock location	Number(10)	N/A	Alloc_detail location (store or wh)
	Case length	Number(12)	N/A	Case length (4 implied decimal places)
	Case width	Number(12)	N/A	Case width (4 implied decimal places)
	Case height	Number(12)	N/A	Case height (4 implied decimal places)
	Case LWH unit of measure	Char(4)	N/A	Case LWH unit of measure
	Case weight	Number(12)	N/A	Case weight (4 implied decimal places)
	Case weight unit of measure	Char(4)	N/A	Case weight unit of measure
	Case liquid volume	Number(12)	N/A	Case liquid volume (4 implied decimal places)
	Case liquid volume unit of measure	Char(4)	N/A	Case liquid volume unit of measure
	Location DUNS number	Char(9)	N/A	Location DUNS number
	Location DUNS loc	Char(4)	N/A	Location DUNS loc
	Old unit cost init	Number(20)	N/A	Old unit cost init (4 implied decimal places)

Table 6-13 (Cont.) File Layout

Record Name	Field Name	Field Type	Default Value	Description
TCUST	New unit cost init	Number(20)	N/A	New unit cost init (4 implied decimal places)
	Item/loc discounts	Number(20)	N/A	Item/loc discounts (4 implied decimal places)
	Record type	Char(5)	TCUST	Describes the file record-customer order information
	Line id	Number(10)	N/A	Unique file line number
	Transaction id	Number(10)	N/A	Unique transaction number
	Delivery first name	Char(120)	N/A	First name for the delivery address on the order
	Delivery phonetic first name	Char(120)	N/A	Phonetic first name for the delivery address on the order
	Delivery last name	Char(120)	N/A	Last name for the delivery address on the order
	Delivery phonetic last name	Char(120)	N/A	Phonetic last name for the delivery address on the order
	Delivery preferred name	Char(120)	N/A	Preferred name for the delivery address on the order
	Delivery company name	Char(120)	N/A	Company name for the delivery address on the order
	Delivery address Line 1	Char(240)	N/A	First line of the delivery address of the customer
	Delivery address Line 2	Char(240)	N/A	Second line of the delivery address of the customer
	Delivery address Line 3	Char(240)	N/A	Third line of the delivery address of the customer
	Delivery county	Char(250)	N/A	County portion of the delivery address
	Delivery city	Char(120)	N/A	City portion of the delivery address
	Delivery state	Char(3)	N/A	State portion of the delivery address
Delivery country ID	Char(3)	N/A	Country portion of the delivery address	
Delivery post	Char(30)	N/A	Postal code portion of the delivery address	

Table 6-13 (Cont.) File Layout

Record Name	Field Name	Field Type	Default Value	Description
	Delivery jurisdiction	Char(10)	N/A	Jurisdiction code of the delivery country-state relationship
	Delivery phone	Char(20)	N/A	Phone number in the delivery information
	Billing first name	Char(120)	N/A	First name for the billing address on the order
	Billing phonetic first name	Char(120)	N/A	Phonetic first name for the billing address on the order
	Billing last name	Char(120)	N/A	Last name for the billing address on the order
	Billing phonetic last name	Char(120)	N/A	Phonetic last name for the billing address on the order
	Billing preferred name	Char(120)	N/A	Preferred name for the billing address on the order
	Billing company name	Char(120)	N/A	Company name for the billing address on the order
	Billing address Line 1	Char(240)	N/A	First line of the billing address of the customer
	Billing address Line 2	Char(240)	N/A	Second line of the billing address of the customer
	Billing address Line 3	Char(240)	N/A	Third line of the billing address of the customer
	Billing county	Char(250)	N/A	County portion of the billing address
	Billing city	Char(120)	N/A	City portion of the billing address
	Billing state	Char(3)	N/A	State portion of the billing address
	Billing country ID	Char(3)	N/A	Country portion of the billing address
	Billing post	Char(30)	N/A	Postal code portion of the billing address
	Billing jurisdiction	Char(10)	N/A	Jurisdiction code of the billing country-state relationship
	Billing phone	Char(20)	N/A	Phone number in the billing information
TTAIL	Record type	Char(5)	TTAIL	Describes file record – marks end of order
	Line id	Number(10)	N/A	Unique file line id
	Transaction id	Number(10)	N/A	Unique transaction id

Table 6-13 (Cont.) File Layout

Record Name	Field Name	Field Type	Default Value	Description
	#Lines in transaction	Number(10)	N/A	Number of lines in transaction
FTAIL	Record type	Char(5)	FTAIL	Describes file record – marks end of file
	Line id	Number(10)	N/A	Unique file line id
	#lines	Number(10)	N/A	Total number of transaction lines in file (not including FHEAD and FTAIL)

For a new order, the “old” fields should be blank. For a changed order, both old and new fields should hold values. If the value has changed, “old” values come from the revision tables for the latest revision before the current one (the last one sent), while new orders come from the ordering tables.

- FHEAD - REQUIRED: File identification, one line per file.
- TORDR - REQUIRED: Order level information, one line per order.
- TITEM - REQUIRED: Item description, multiple lines per order possible.
- TPACK - OPTIONAL: Pack contents, multiple lines per order possible. This line will be written only for pack items.
- TSHIP - REQUIRED: Ship to location and quantity, allocation location, multiple lines per item possible. Allocation information is optional on this line-will exist if premark_ind is 'Y'.
- TCUST - OPTIONAL: Customer order information, one line per order. This line will be written only for Drop Ship Customer Orders.
- TTAIL - REQUIRED: Order end, one line per order.
- FTAIL - REQUIRED: End of file marker, one line per file.

Design Assumptions

N/A

Download Summary of Outstanding Orders on OTB by Subclass (otbdlord)

Module Name	otbdlord.pc
Description	Download Summary of Outstanding Orders on OTB by Subclass
Functional Area	Open To Buy
Module Type	Integration
Module Technology	ProC
Catalog ID	RMS13
Wrapper Script	rmswrap_out.ksh

Schedule

Oracle Retail Merchandising Batch Schedule

Design Overview

This batch program runs at the end of the half to delete rows from the OTB table that are at least one half old. The current and previous half's OTB data is retained. The number of days that OTB records are retained by Merchandising is not configurable via a system parameter.

Restart/Recovery

The logical unit of work for the otbdlord module is department/class/subclass. The commit_max_ctr field should be set to prevent excessive rollback space usage, and to reduce the overhead of the file I/O. The recommended commit counter setting is 10000 records. Each time the record counter equals the maximum recommended commit number, an application image array record will be written to the restart_start_array for restart/recovery if a fatal error occurs.

I/O Specification

Integration Type	Download from Merchandising
File Name	Determined by runtime parameter
Integration Contract	IntCon000029

Output File Layout

Table 6-14 otbdlord.pc - Output File

Record Name	Field Name	Field Type	Default Value	Description
File Header	File Type Record Descriptor	Char(5)	FHEAD	Identifies file record type
	File Line Sequence Number	Number(10)	N/A	Keeps track of the record's position in the file by line number
	File Type Definition	Char(4)	OOEX	Identifies file as 'OTB Outstanding Order Export'
	File Create Date	Char(14)	N/A	Date the file was created in YYYYMMDD format. Remaining six characters are blank.
File Detail	File Type Record Descriptor	Char(5)	FDETL	Identifies file record type

Table 6-14 (Cont.) otbdlord.pc - Output File

Record Name	Field Name	Field Type	Default Value	Description
	File Line Sequence Number	Number(10)	N/A	Keeps track of the record's position in the file by line number
	Transaction Set Control Number	Number(14)	N/A	Sequence number used to force unique detail record check
	Department	Number(4)	N/A	The number of the department which contains the outstanding order quantity value
	Class	Number(4)	N/A	The number of the class which contains the outstanding order quantity value.
	Subclass	Number(4)	N/A	The number of the subclass which contains the outstanding order quantity value
	N Outstanding Amt	Number(20)	N/A	The amount of outstanding non-basic orders (order type N/B) for past periods; value includes 4 implied decimal places
	B Outstanding Amt	Number(20)	N/A	The amount of outstanding buyer-replenished basic (order type BRB) orders for past periods; value includes 4 implied decimal places

Table 6-14 (Cont.) otbdlord.pc - Output File

Record Name	Field Name	Field Type	Default Value	Description
	A Outstanding Amt	Number(20)	N/A	The amount of outstanding auto-replenished basic (order type ARB) orders for past periods; value includes 4 implied decimal places
File Trailer	File Type Record Descriptor	Char(5)	FTAIL	Identifies file record type
	File Line Sequence Number	Number(10)	N/A	Keeps track of the record's position in the file by line number
	Control Number File Line Count	Control Number File Line Count Number(10)	N/A	Total number of all transaction lines, not including file header and trailer

Design Assumptions

N/A

On Order Publication API (BDI_OnOrder_Tx_PF_From_RMS_EOW_JOB)

This section describes the On Order Publication BDI.

Functional Area

Inventory Tracking

Business Overview

BDI (Bulk Data Integration) is an integration layer that facilitates the bulk transfer of quantities On Order information from Merchandising to other Oracle Retail Applications. On this particular integration stream, the data flow is from Merchandising to BDI, and then BDI to downstream applications. To accomplish this data transfer, BDI will be calling a Merchandising-owned API that will pull data from Merchandising and deliver these to the BDI integration layer. This API will be in the form of a PLSQL function inside a PLSQL package.

Package Impact

Filename: bdimfpb.pls

```
BDI_MFP_SQL.ON_ORDER_UP(O_error_message IN OUT RTK_ERRORS.RTK_TEXT%TYPE,
                        O_control_id     IN OUT NUMBER,
                        I_job_context     IN    VARCHAR2)
```

This function begins by calling a BDI function that signals the start of the interface process. The BDI function will update the internal BDI control tables to track the progress of the API.

A DML insert statement is then executed to populate the BDI outbound table that resides in the BDI_RMS_INT_SCHEMA schema. This outbound table is loaded with records from the Merchandising Order tables/view.

After the insert, another call to a BDI function is performed to signify the successful loading of records. This will update the internal BDI control tables.

A database commit is issued, and the control Id is returned by the API.

Data Definition XML

The BDI interface staging tables are generated based on the XML schema definition

Data Flow	Description	XML Schema Definition (XSD)
On Order	On Order upload to BDI	OnOrder_Tx_BdiInterfaceModule.xml

Table Impact

TABLE	SELECT	INSERT	UPDATE	DELETE
ON_ORDER_OUT	No	Yes	No	No
V_BDI_MFP_ON_ORDER	Yes	No	No	No

Replenishment Item Location Publication API (BDI_ReplItemLoc_Fnd_PF_From_RMS_JOB)

This section describes the Replenishment Item Location Publication BDI.

Functional Area

Item

Business Overview

BDI (Bulk Data Integration) is an integration layer that facilitates the bulk transfer of Replenishment Item Location information from Merchandising to other Oracle Retail Applications. On this particular integration stream, the data flow is from Merchandising to BDI, and then BDI to downstream applications. To accomplish this data transfer, BDI will be calling a Merchandising-owned API that will pull data from Merchandising and deliver these to the BDI integration layer. This API will be in the form of a PLSQL function inside a PLSQL package.

Package Impact

Filename: bdiitemb.pls

```

BDI_ITEM_SQL.REPL_ITEM_LOC_UP(O_error_message IN OUT VARCHAR2,
                               O_control_id   IN OUT NUMBER,
                               I_job_context   IN      VARCHAR2)

```

This function begins by calling a BDI function that signals the start of the interface process. The BDI function will update the internal BDI control tables to track the progress of the API.

A DML insert statement is then executed to populate the BDI outbound table that resides in the BDI_RMS_INT_SCHEMA schema. This outbound table is loaded with records from the Merchandising REPL_ITEM_LOC table.

After the insert, another call to a BDI function is performed to signify the successful loading of records. This will update the internal BDI control tables.

A database commit is issued, and the control Id is returned by the API.

Data Definition XML

The BDI interface staging tables are generated based on the XML schema definition

Data Flow	Description	XML Schema Definition (XSD)
Replenishment Item Location	Replenishment Item Location upload to BDI	ReplItemLoc_Fnd_BdiInterfaceModule.xml

Table Impact

TABLE	SELECT	INSERT	UPDATE	DELETE
REPL_ITEM_LOC_OUT	No	Yes	No	No
ITEM_MASTER	Yes	No	No	No
REPL_ITEM_LOC	Yes	No	No	No

Import Management

When using the Import Management features in Merchandising, there are several outbound integration processes that are available for customs entry and letter of credit functions. These integrations are only available if you are using not using Simplified Import Management (based on your system options configurations).

For additional information about import management, including detailed flow diagrams, see the *RTM Overview* white paper in the Merchandising Documentation Library (Doc ID: 1585843.1).

- [Download of Customs Entry Transactions to Brokers \(cednld\)](#)
- [Letter of Credit Amendment Download \(lcmdnld\)](#)
 - [SWIFT File Conversion – Letter of Credit Amendment \(lcmt707\)](#)
- [Letter of Credit Application Download \(lcadnld\)](#)
 - [SWIFT File Conversion - Letter of Credit Application \(lcmt700\)](#)

Download of Customs Entry Transactions to Brokers (cednld)

Module Name	cednld.pc
Description	Download of Customs Entry Transactions to Brokers
Functional Area	Oracle Retail Trade Management
Module Type	Integration
Module Technology	ProC
Catalog ID	RMS53
Wrapper Script	batch_cednld.ksh

Schedule

Oracle Retail Merchandising Batch Schedule

Design Overview

This program is used to download custom entry information from the Merchandising database to brokers. Each night, this program reads all customs entry (CE) transactions that are in Sent status for a broker ID. These transactions are written to a flat file and the status is changed to Downloaded. One flat file is written per broker.

Restart/Recovery

The Logical Unit of Work for the program is a single row from the customs entry header table. Restart/Recovery will be used for init and commit.

Table based restart/recovery must be used. The commit max counter field should be set to prevent excessive rollback space usage, and to reduce the overhead of file I/O. The recommended commit counter setting is 1000 records (subject to change based on implementation).

I/O Specification

Integration Type	Download from Merchandising
File Name	Determined by runtime parameter
Integratin Contract	IntCon000050

Output File Layout

Table 6-15 Output File Layout

Record Name	Field Name	Field Type	Default Value	Description
File Header	File Type Descriptor	Char(5)	FHEAD	Identifies file record type
	File Line Identifier	Number(10)	Nine leading zeroes: 0000000001	ID of current line being processed by input file

Table 6-15 (Cont.) Output File Layout

Record Name	Field Name	Field Type	Default Value	Description
THEAD	File Type Definition	Char(4)	CEDN	Identifies file as 'Customs Entry download'
	File Create Date	Date	Create date	Vdate in YYYYMMDDH H24MISS format
	File Type Descriptor	Char(5)	THEAD	Identifies file record type
	File Line Identifier	Number(10)	Incremented internally	ID of current line being processed by input file
	CE ID	Number(10)	ce_head.ce_id	N/A
	Entry No	Char (15)	ce_head.entry_no	N/A
	Entry Date	Char(14)	ce_head.entry_date	YYYYMMDDH H24MISS format
	Entry Status	Char(6)	ce_head.entry_status	N/A
	Entry Type	Char(6)	ce_head.entry_type	N/A
	Entry Port	Char(5)	ce_head.entry_port	N/A
	Summary Date	Char(14)	ce_head.summary date	YYYYMMDDH H24MISS format
	Broker ID	Char(10)	ce_head.broker_id	N/A
	Broker Ref. ID	Char(18)	ce_head.broker_ref_id	N/A
	File Number	Char(18)	ce_head.file_no	N/A
	Importer ID	Char(10)	ce_head.importer_id	N/A
Import Country	Char(3)	ce_head.import_country_id	N/A	
Currency Code	Char(3)	ce_head.currency_code	N/A	

Table 6-15 (Cont.) Output File Layout

Record Name	Field Name	Field Type	Default Value	Description
	Exchange Rate	Number(20,10)	ce_head.exchange_rate*1000000000 (with 10 implied decimal places)	N/A
	Bond Number	Char(18)	ce_head.bond_no	N/A
	Bond Type	Char(6)	ce_head.bond_type	N/A
	Surety Code	Char(6)	ce_head.surety_code	N/A
	Consignee ID	Char(10)	ce_head.consignee_id	N/A
	Live Indicator	Char(1)	ce_head.live_ind	N/A
	Batch Number	Char(20)	ce_head.batch_no	N/A
	Entry Team	Char(3)	ce_head.entry_team	N/A
	Liquidation Amount	Number(20,4)	ce_head.liquidation_amt*10000 (4 implied decimal places)	N/A
	Liquidation Date	Date	ce_head.liquidation_date	YYYYMMDDH24MISS format
	Reliquidation Amount	Number(20,4)	ce_head.reliquidation_amt*10000 (4 implied decimal places)	N/A
	Reliquidation Date	Date	ce_head.reliquidation_date	YYYYMMDDH24MISS format
	Merchandise Loc	Char(40)	ce_head.merchandise_loc	N/A
	Location Code	Char(4)	ce_head.location_code	N/A
TSHIP	File Type Descriptor	Char(5)	TSHIP	Identifies file record type

Table 6-15 (Cont.) Output File Layout

Record Name	Field Name	Field Type	Default Value	Description
	File Line Identifier	Number(10)	Incremented internally	ID of current line being processed by input file
	Vessel ID	Char(20)	ce_shipment.vessel_id	N/A
	Voyage Flt ID	Char(10)	ce_shipment.voyage_ftl_id	N/A
	Estimated Departure Date	Date	ce_shipment.estimated_depart_date	YYYYMMDDH24MISS format
	Vessel SCAC Code	Char(6)	ce_shipment.vessel_scac_code	N/A
	Lading Port	Char(5)	ce_shipment.lading_port	N/A
	Discharge Port	Char(5)	ce_shipment.discharge_port	N/A
	Tran Mode ID	Char(6)	ce_shipment.tran_mode_id	N/A
	Export Date	Date	ce_shipment.export_date	YYYYMMDDH24MISS
	Import Date	Date	ce_shipment.import_date	YYYYMMDDH24MISS
	Arrival Date	Date	ce_shipment.arrival_date	YYYYMMDDH24MISS
	Export Country	Char(3)	ce_shipment.export_country_id	N/A
	Shipment Number	Number(10)	ce_shipment.shipment_no	N/A
TORDI	File Type Descriptor	Char(5)	TORDI	Identifies file record type
	File Line Identifier	Number(10)	Incremented internally	ID of current line being processed by input file
	Order Number	Number(8)	ce_ord_item.order_no	N/A
	Item	Char (25)	ce_ord_item.item	N/A

Table 6-15 (Cont.) Output File Layout

Record Name	Field Name	Field Type	Default Value	Description
	BL AWB ID	Char(30)	ce_ord_item.bl_awb_id	'MULTI' – means multiple airway bills (otherwise a single airway bill will be retrieved)
	Invoice ID	Char(30)	ce_ord_item.invoice_id	N/A
	Invoice Date	Date	ce_ord_item.invoice_date	YYYYMMDDH H24MISS format
	Invoice Amount	Number(20,4)	ce_ord_item.invoice_amt*10000 (4 implied decimal places)	N/A
	Currency Code	Char(3)	ce_ord_item.currency_code	N/A
	Exchange Rate	Number(20,10)	ce_ord_item.exchange_rate*1000000000 (10 implied decimal places)	N/A
	Manifest Item Quantity	Number(12,4)	ce_ord_item.manifest_qty*10000 (4 implied decimal places)	N/A
	Manifest Item Quantity UOM	Char(4)	ce_ord_item.manifest_qty_uom	N/A
	Carton Quantity	Number(12,4)	ce_ord_item.carton_qty*10000 (4 implied decimal places)	N/A
	Carton Quantity UOM	Char(4)	ce_ord_item.carton_qty_uom	N/A
	Gross Weight	Number(12,4)	ce_ord_item.gross_wt*1000 (4 implied decimal places)	N/A
	Gross Weight UOM	Char(4)	ce_ord_item.gross_wt_uom	N/A

Table 6-15 (Cont.) Output File Layout

Record Name	Field Name	Field Type	Default Value	Description
	Net Weight	Number(12,4)	ce_ord_item.net_wt*10000 (4 implied decimal places)	N/A
	Net Weight UOM	Char(4)	ce_ord_item.net_wt_uom	N/A
	Cubic	Number(12,4)	ce_ord_item.cubic*10000 (4 implied decimal places)	N/A
	Cubic UOM	Char(4)	ce_ord_item.cubic_uom	N/A
	Cleared Quantity	Number(12,4)	ce_ord_item.cleared_qty*10000 (4 implied decimal places)	N/A
	Cleared Quantity UOM	Char(4)	ce_ord_item.cleared_qty_uom	N/A
	In Transit Number	Char(15)	ce_ord_item.in_transit_no	N/A
	In Transit Date	Date	ce_ord_item.in_transit_date	YYYYMMDDH H24MISS format
	Rush Indicator	Char(1)	ce_ord_item.rush_ind	N/A
	Related Indicator	Char(1)	ce_ord_item.related_ind	N/A
	Tariff Treatment	Char(10)	ce_ord_item.tariff_treatment	N/A
	Ruling Number	Char(10)	ce_ord_item.ruling_no	N/A
	Do Number	Char(10)	ce_ord_item.do_no	N/A
	Do Date	Date	ce_ord_item.do_date	YYYYMMDDH H24MISS format
	Manufacture ID	Char(18)	sup_import_atr.mfg_id	N/A
TBLAW	File Type Descriptor	Char(5)	TBLAW	Identifies file record type

Table 6-15 (Cont.) Output File Layout

Record Name	Field Name	Field Type	Default Value	Description
TCONT	File Line Identifier	Number(10)	Incremented internally	ID of current line being processed by input file
	BL AWB ID	Char(30)	Transportation .bl_awb_id	N/A
	File Type Descriptor	Char(5)	TCONT	Identifies file record type
	File Line Identifier	Number(10)	Incremented internally	ID of current line being processed by input file
	Container ID	Char(20)	Transportation .container_id	N/A
TLICV	Container SCAC Code	Char(6)	Transportation .container_scac_code	N/A
	File Type Descriptor	Char(5)	TLICV	Identifies file record type
	File Line Identifier	Number(10)	Incremented internally	ID of current line being processed by input file
	License/Visa Type	Char(6)	ce_lic_visa.license_visa_type	N/A
	License/Visa ID	Char(30)	ce_lic_visa.license_visa_id	N/A
	License/Visa Quantity	Number(12,4)	ce_lic_visa.license_visa_qty*10000 (4 implied decimal places)	N/A
	License/Visa Quantity UOM	Char(4)	ce_lic_visa.license_visa_qty_uom	N/A
	Quota Category	Char (6)	ce_lic_visa.quota_category	N/A
	Net Weight	Number(12,4)	ce_lic_visa.net_weight*10000 (4 implied decimal places)	N/A
Net Weight UOM	Char(4)	ce_lic_visa.net_weight_uom	N/A	
Holder ID	Char(18)	ce_lic_visa.holder_id	N/A	

Table 6-15 (Cont.) Output File Layout

Record Name	Field Name	Field Type	Default Value	Description
TCHRG	File Type Descriptor	Char(5)	TCHRG	Identifies file record type
	File Line Identifier	Number(10)	Incremented internally	ID of current line being processed by input file
	Sequence Number	Number(6)	ce_charges.s eq_no	N/A
	Pack Item	Char(25)	ce_charges.p ack_item	N/A
	HTS	Char(10)	ce_charges.ht s	N/A
	Effect From Date	Date	ce_charges.ef fect_from	YYYYMMDDH H24MISS format
	Effect To Date	Char(14)	ce_charges.ef fect_to	YYYYMMDDH H24MISS format
	Component ID	Date	ce_charges.c omp_id	N/A
	Component Rate	Number(20,4)	ce_charges.c omp_rate*100 00 (4 implied decimal places)	N/A
	Per Count UOM	Char(3)	ce_charges.p er_count_uom	N/A
Component Value	Number(20,4)	ce_charges.c omp_value * 10000 (4 implied decimal places)	N/A	
TMDOC	File Type Descriptor	Char(5)	TMDOC	Identifies file record type
	File Line Identifier	Number(10)	Incremented internally	ID of current line being processed by input file
	Doc_id	Number(6)	Missing_doc.d oc_id	N/A
	Received_date	Date	Missing_doc.r eceived_date	YYYYMMDDH H24MISS format
FTAIL	File Type Descriptor	Char(5)	FTAIL	Identifies file record type

Table 6-15 (Cont.) Output File Layout

Record Name	Field Name	Field Type	Default Value	Description
	File Line Identifier	Number(10)	Incremented internally	ID of current line being processed by input file.
	File Record Counter	Number(10)	Determined internally	Number of records/ transactions processed in current file (only records between head & tail)

Design Assumptions

N/A

Letter of Credit Amendment Download (lcmdnld)

Module Name	lcmdnld.pc
Description	Letter of Credit Amendment Download
Functional Area	Oracle Retail Trade Management
Module Type	Integration
Module Technology	ProC
Catalog ID	RMS56
Wrapper Script	rmswrap_dnld_in.ksh

Schedule

Oracle Retail Merchandising Batch Schedule

Design Overview

lcmdnld.pc downloads amended letter of credit information to a bank, in the S.W.I.F.T. format.

Online user actions flag LCs for download by writing to the LC_DOWNLOAD table.

Restart/Recovery

Restart/recovery for this program is set up at the lc_ref_id level. The recommended commit counter setting is 1000 records (subject to change based on experimentation).

I/O Specification

Integration Type	Download from Merchandising
File Name	Determined by runtime parameter

Integratin Contract IntCon000053

Output File Layout

Table 6-16 File Layout

Record Name	Field Name	Field Type	Default Value	Description
File Header	File Type Record Descriptor	Char(5)	FHEAD	Identifies file record type
	File Line Sequence Number	Number(10)	Line number in file	Keeps track of the record's position in the file by line number
	File Type Definition	Char(4)	LCAM	Identifies file as 'Letter of Credit Amendment'
	File Create Date	Char(14)	Create date	Current date, formatted to 'YYYYMMDDHH24MISS'
Transaction Header	Filetype Record descriptor	Char(5)	THEAD	Identifies file record type
	File Line Sequence Number	Number (10)	Line number in file	Keeps track of the record's position in the file by line number
	Transaction Set Control Number	Number (10)	Sequence number	Used to force unique file check
	Issuing Bank	Char(10)	lc_head.issuimg_bank	Used to sort the LCs into individualized bank SWIFT formatted files (using another program) – bank where LC application is headed
	Issuing Bank Name	Char(240)	partner.partner_desc	The description from the partner table where partner_id = issuing_bank and partner_type = 'BK'
	Issuing Bank Address 1	Char(240)	addr.add_1	Mandatory line of address
	Issuing Bank Address 2	Char(240)	addr.add_2	Non-mandatory line of address (can be null)
	Issuing Bank Address 3	Char(240)	addr.add_3	Non-mandatory line of address (can be null)
	Issuing Bank City	Char(120)	addr.city	City bank located in
	Issuing Bank State	Char(3)	addr.state	State, if applicable, where bank located in
	Issuing Bank Post Code	Char(30)	addr.post	Post code, if applicable, where bank located in
	Issuing Bank Country	Char(3)	addr.country_id	Country bank located in

Table 6-16 (Cont.) File Layout

Record Name	Field Name	Field Type	Default Value	Description
	Letter of Credit	Number (8)	lc_detail.lc_ref_id	The LC_REF_ID off the LC_DETAIL table
	Bank Letter of Credit ID	Char(16)	lc_head.bank_lc_id	The BANK_LC_ID off the LC_HEAD table
	Currency Code	Char(3)	lc_head.currency_code	The CURRENCY_CODE off the LC_HEAD table
	Date of Issue/ Transfer of the Credit	Char(14)	lc_head.confirmed_date	Date the Issuing Bank thinks is the date of issue—when it was officially confirmed, formatted to 'YYYYMMDDHH24MISS'
	Current Amount of LC	Number (20,4)	N/A	This amount will be calculated in the get_current_amount() function and will be the net amount of the LC calculated only using amendments that have been downloaded. Normally, the net amount is calculated using amendments in the 'D'ownloaded status
	Beneficiary	Number (10)	lc.head.beneficiary	Party in favor of which the LC is being issued
	Beneficiary Name	Char(240)	supr.sup_name	Beneficiary (supplier) name from the SUPS table
	Beneficiary Address 1	Char(240)	addr.add_1	Mandatory line of address
	Beneficiary Address 2	Char(240)	addr.add_2	Non-mandatory line of address (can be null)
	Beneficiary Address 3	Char(240)	addr.add_3	Non-mandatory line of address (can be null)
	Beneficiary City	Char(120)	addr.city	City beneficiary located in
	Beneficiary State	Char(3)	addr.state	State, if applicable, where beneficiary located in
	Beneficiary Post Code	Char(30)	addr.post	Post code, if applicable, where beneficiary located in
	Beneficiary Country	Char(3)	addr.country_id	Country beneficiary located in
Transaction Detail	File Type Record Descriptor	Char(5)	TDETL	Identifies file record type
	File Line Sequence Number	Number (10)	line number in file	Keeps track of the record's position in the file by line number
	Transaction Set Control Number	Number (10)	sequence number	Used to force unique file check number

Table 6-16 (Cont.) File Layout

Record Name	Field Name	Field Type	Default Value	Description
	Amendment Number	Number (8)	lc_amendment s.amend_no	Holds the amendment number for the amendment
	Order_no	Number (8)	lc_amendment s.order_no	Order_no, if applicable, that is attached to the LC that is being amended
	Item	Char(25)	lc_amendment s.item	Item being amended, either a Style or Staple sku
	Value Being Amended	Char(6)	lc_amendment s.amended_value	LC Field being amended. Can be any of the following code_types: CODE CODE_DESC AI Add Item AO Add PO ARQD Add Reqd Doc. C Cost ED Expiration Date ESD Earliest Ship Date LSD Latest Ship Date NA Net Amount ND Negotiation Days OC Origin Country OQ Order Quantity PE Place of Expiry PRT Presentation Terms PSF Partial Ship Flag RI Remove Item RO Remove PO RRQD Remove Reqd Doc TFF Transferable Flag TSF Transshipment Flag
	Value Being Amended Description	Char(40)	code_detail.co de_desc	The Value Being Amended decoded (see the above list). Will possibly be used when printing to the SWIFT file MT 707 for clarity
	Original Value of Amended Field	Char(45)	lc_amendment s.original_value	Current value of field that is being amended
	New Value of Amended Field	Char (2000)	lc_amendment s.new_value	New value of the field that is being amended

Table 6-16 (Cont.) File Layout

Record Name	Field Name	Field Type	Default Value	Description
	Description of New Value	Char(40)	code_detail.co de_desc	The new value decoded (or fetched from a table, as in the origin_country case)– only applicable to the following amended values: place of expiry, title_pass_location, origin_country, presentation terms, purchase type
	Sign	Char(1)	N/A	If the effect is negative it will be “-“ if the effect is positive it will be “ “
	Effect	Number (20,4)	lc.amendment s.effect	Effect that amendment will have on LC if amendment to change qty or cost of a PO or amount of LC itself
	Date of Amendment	Char(14)	Lc_amendmen ts.accept_date	Date on which Issuing Bank (or issuing party, in this case the retailer) considers the credit as being amended, formatted to 'YYYYMMDD HH24MISS'
Transaction Text	File Type Record Descriptor	Char(5)	TTEXT	Identifies file record type
	File Line Sequence Number	Number (10)	line number in file	Keeps track of the record's position in the file by line number
	Transaction Set Control Number	Number (10)	sequence number	Used to force unique file check
	Amendment Text	Char (2000)	text description	A text description of the individual amendment (for each TDETL line of the output file) built by the package LC_AMEND_SQL. AMEND_TEXT.
Transaction Trailer	File Type Record Descriptor	Char (5)	TTAIL	Identifies File Record Type
	File Line Sequence Number	Number (10)	Line Number in file	ID of current line being created for output file
	Transaction set control number	Number (10)	Sequence number	Used to force unique file check
	Transaction detail line count	Number (10)	ID of current line being created for output file	Some of the detail lines within a transaction

Table 6-16 (Cont.) File Layout

Record Name	Field Name	Field Type	Default Value	Description
File Trailer	File Type Record Descriptor	Char(5)	FTAIL	Identifies file record type
	File Line Sequence Number	Number (10)	line number in file	Keeps track of the record's position in the file by line number
	Control Number File Line Count	Number (10)	total detail lines	Sum of all transaction lines, not including the file header and trailer

Letter of Credit Application Download (lcadnld)

Module Name	Lcadnld.pc
Description	Letter of Credit Application Download
Functional Area	Retail Trade Management
Module Type	Integration
Module Technology	ProC
Catalog ID	RMS57
Wrapper Script	rmswrap_dnld_in.ksh

Schedule

See Oracle Merchandising Batch Schedule.

Design Overview

Lcadnld sends letter of credit (LC) applications to partner banks. Online user actions flag LCs for download by writing to the LC_DOWNLOAD table.

Restart/Recovery

Restart/recovery for this program is set up at the lc_ref_id level. The recommended commit counter setting is 10000 records (subject to change based on experimentation).

I/O Specification

Integration Type	Download from Merchandising
File Name	Determined by runtime parameter
Integratin Contract	IntCon000052

Output File Layout

Table 6-17 File Layout

Record Name	Field Name	Field Type	Default Value	Description
File Header	File Type Record Descriptor	Char(5)	FHEAD	Identifies file record type
	File Line Identifier	Number(10)	line number in file	ID of current line being created for output file
	File Type Definition	Char(4)	LCAP	Identifies file as 'Letter of Credit Application'
	File Create Date	Char(14)	create date	Current date, formatted to 'YYYYMMDDH H24MISS'
File Detail	File Type Record Descriptor	Char(5)	THEAD	Identifies file record type
	File Line Sequence Number	Number(10)	line number in file	ID of current line being created for output file.
	Transaction Set Control Number	Number(10)	sequence number	Used to force unique file check
	Issuing Bank	Char(10)	lc_head.issuing_bank	Used to sort the LCs into individualized bank SWIFT formatted files (using another program) - bank where LC application is headed
	Issuing Bank Name	Char(240)	partner.partner_desc	The description from the partner table where partner_id = issuing_bank and partner_type = 'BK'
	Issuing Bank Address 1	Char(240)	addr.add_1	Mandatory line of address
	Issuing Bank Address 2	Char(240)	addr.add_2	Non-mandatory line of address (can be null)

Table 6-17 (Cont.) File Layout

Record Name	Field Name	Field Type	Default Value	Description
	Issuing Bank Address 3	Char(240)	addr.add_3	Non-mandatory line of address (can be null)
	Issuing Bank City	Char(120)	addr.city	City bank located in
	Issuing Bank State	Char(3)	addr.state	State, if applicable, where bank located in
	Issuing Bank Post Code	Char(30)	addr.post	Post code, if applicable, where bank located in
	Issuing Bank Country	Char(3)	addr.country_id	Country bank located in
	Advising Bank	Char(10)	lc_head.advising_bank	Used to sort the LCs into individualized bank SWIFT formatted files (using another program) - bank where LC application is headed
	Advising Bank Name	Char(240)	Partner.partner_desc	The description from the partner table where partner_id = advising_bank and partner_type = 'BK'
	Advising Bank Address 1	Char(240)	Addr.add_1	Mandatory line of address
	Advising Bank Address 2	Char(240)	Addr.add_2	Non-mandatory line of address (can be null)
	Advising Bank Address 3	Char(240)	Addr.add_3	Non-mandatory line of address (can be null)
	Advising Bank City	Char(120)	Addr.city	City bank located in
	Advising Bank State	Char(3)	Addr.state	State, if applicable, where bank located in

Table 6-17 (Cont.) File Layout

Record Name	Field Name	Field Type	Default Value	Description
	Advising Bank Post Code	Char(30)	Addr.post	Post code, if applicable, where bank located in
	Advising Bank Country	Char(3)	Addr.country_id	Country bank located in
	Letter of Credit	Number(8)	lc_head.lc_ref_id	The LC_REF_ID off the LC_HEAD table
	Form Type	Char(6)	lc_head.form_type	The level of detail that the LC will send to the issuing bank
	Form Type Description	Char(40)	code_detail.code_desc	Describes the form type: Long or Short
	Letter of Credit Type	Char(6)	lc_head.lc_type	Describes the form type: Long or Short
	Letter of Credit Type Description	Char(40)	code_detail.code_desc	Describes the LC type: Master, Normal, Revolving
	Form of Letter of Credit – I	Char(1)	sup_import_attr.revocable_ind	The REVOCABLE_IND from the SUP_IMPORT_ATTR table
	Form of Letter of Credit – II	Char(1)	lc_head.transferable_ind	Indicates if LC transferable
	Application Date	Char(14)	lc_head.application_date	Date the LC is created within Import Management/ Merchandising, formatted to 'YYYYMMDD HH24MISS'
	Expiration Date	Char(14)	lc_head.expiration_date	The date the LC expires, formatted to 'YYYYMMDD HH24MISS'
	Place of Expiry	Char(6)	lc_head.place_of_expiry	Code for the place the LC will expire

Table 6-17 (Cont.) File Layout

Record Name	Field Name	Field Type	Default Value	Description
	Place of Expiry Description	Char(40)	desc is retrieved through a decode	The description of the place the LC will expire
	Applicant	Char(10)	lc_head.applicant	Party on whose behalf the LC is being issued
	Applicant Name	Char(240)	partner.partner_desc	The description from the partner table where partner_id = applicant and partner_type = 'AP'
	Applicant Address 1	Char(240)	addr.add_1	Mandatory line of address
	Applicant Address 2	Char(240)	addr.add_2	Non-mandatory line of address (can be null)
	Applicant Address 3	Char(240)	addr.add_3	Non-mandatory line of address (can be null)
	Applicant City	Char(120)	addr.city	City applicant located in
	Applicant State	Char(3)	addr.state	State, if applicable, where applicant located in
	Applicant Post Code	Char(10)	addr.post	Post code, if applicable, where applicant located in
	Applicant Country	Char(3)	addr.country_id	Country applicant located in
	Beneficiary	Number(10)	lc.head.beneficiary	Party in favor of which the LC is being issued
	Beneficiary Name	Char(240)	supp.sup_name	Beneficiary (supplier) name from the SUPS table
	Beneficiary Address 1	Char(240)	addr.add_1	Mandatory line of address
	Beneficiary Address 2	Char(240)	addr.add_2	Non-mandatory line of address (can be null)

Table 6-17 (Cont.) File Layout

Record Name	Field Name	Field Type	Default Value	Description
	Beneficiary Address 3	Char(240)	addr.add_3	Non-mandatory line of address (can be null)
	Beneficiary City	Char(120)	addr.city	City beneficiary located in
	Beneficiary State	Char(3)	addr.state	State, if applicable, where beneficiary located in
	Beneficiary Post Code	Char(30)	addr.post	Post code, if applicable, where beneficiary located in
	Beneficiary Country	Char(3)	addr.country_id	Country beneficiary located in
	Currency Code	Char(3)	lc_head.currency_code	The country of origin for the orders on the LC
	Exchange Rate	Number (20,10)	lc_head.exchange_rate	Exchange_rate to convert LC currency to Merchandising currency
	Origin Country ID	Char(3)	lc_head.origin_country_id	Origin country of the orders associated with the LC
	Presentation Terms	Char(6)	lc_head.presentation_terms	Code for the terms of presentation
	Presentation Terms Description	Char(40)	desc is retrieved through a decode	Description of the terms of presentation
	Purchase Type	Char(6)	lc_head.purchase_type	Code for the purchase type
	Purchase Type Description	Char(40)	desc is retrieved through a decode	Description of the purchase type
	Advice Method	Char(6)	lc_head.advice_method	Code for the advice method

Table 6-17 (Cont.) File Layout

Record Name	Field Name	Field Type	Default Value	Description
	Advice Method Description	Char(40)	desc is retrieved through a decode	Description of the advice method (eg. Full Wire, Mail, and so on)
	Issuance	Char(6)	lc_head.issuance	Code for the issuance
	Issuance Description	Char(40)	desc is retrieved through a decode	Description of the issuance (for example Cable, Telex, and so on)
	Amount Type	Char(6)	lc_head.amount_type	If 'E'xact, then amount must be exact, if 'A'pproximate then amount can be within variance percent
	Amount Type Description	Char(40)	desc is retrieved through a decode	Description of amount_type
	Amount	Number (20,4)	lc_head.amount	The total amt of the Letter of Credit
	Variance Percent	Number (12,4)	lc_head.variance_pct	Allowed currency variance percent for the LC
	Specification	Char(6)	lc_head.specification	Code for any condition for the credit, such as, "maximum", and so on
	Specification Description	Char(40)	desc is retrieved through a decode	Description of condition for the credit, such as, "maximum", and so on
	Credit Available With	Char(10)	lc_head.credit_avail_with	Code for bank with which credit is available

Table 6-17 (Cont.) File Layout

Record Name	Field Name	Field Type	Default Value	Description
	Credit With Bank Name	Char(40)	partner.partner_desc	The description from the partner table where partner_id = credit_avail_with and partner_type = 'BK'
	Credit With Address 1	Char(240)	addr.add_1	Mandatory line of address
	Credit With Address 2	Char(240)	addr.add_2	Non-mandatory line of address (can be null)
	Credit With Address 3	Char(240)	addr.add_3	Non-mandatory line of address (can be null)
	Credit With City	Char(120)	addr.city	City creditor located in
	Credit With State	Char(3)	addr.state	State, if applicable, where creditor located in
	Credit With Post Code	Char(30)	addr.post	Post code, if applicable, where creditor located in
	Credit With Country	Char(3)	addr.country_id	Country creditor located in
	Drafts At	Char(6)	lc_head.drafts_at	Specifies the terms of the drafts to be drawn under the LC
	Drafts At Description	Char(40)	desc is retrieved through a decode	Description of the terms of the drafts to be drawn under the LC
	Drawee	Char(10)	lc_head.paying_bank	Identifies drawee of drafts to be drawn under LC (paying bank)

Table 6-17 (Cont.) File Layout

Record Name	Field Name	Field Type	Default Value	Description
	Drawee Name	Char(240)	partner.partner_desc	The description from the partner table where partner_id = paying_bank and partner_type = 'BK'
	Drawee Address 1	Char(240)	addr.add_1	Mandatory line of address
	Drawee Address 2	Char(240)	addr.add_2	Non-mandatory line of address (can be null)
	Drawee Address 3	Char(240)	addr.add_3	Non-mandatory line of address (can be null)
	Drawee City	Char(120)	addr.city	City bank located in
	Drawee State	Char(3)	addr.state	State, if applicable, where bank located in
	Drawee Post Code	Char(30)	addr.post	Post code, if applicable, where bank located in
	Drawee Country	Char(3)	addr.country_id	Country bank located in
	Negotiating Bank	Char(10)	lc_head.negotiating_bank	Identifies the negotiating bank
	Negotiating Bank Name	Char(240)	partner.partner_desc	The description from the partner table where partner_id = negotiating_bank and partner_type = 'BK'
	Negotiating Bank Address 1	Char(240)	addr.add_1	Mandatory line of address
	Negotiating Bank Address 2	Char(240)	addr.add_2	Non-mandatory line of address (can be null)
	Negotiating Bank Address 3	Char(240)	addr.add_3	Non-mandatory line of address (can be null)

Table 6-17 (Cont.) File Layout

Record Name	Field Name	Field Type	Default Value	Description
	Negotiating Bank City	Char(120)	addr.city	City bank located in
	Negotiating Bank State	Char(3)	addr.state	State, if applicable, where bank located in
	Negotiating Bank Post Code	Char(30)	addr.post	Post code, if applicable, where bank located in
	Negotiating Bank Country	Char(3)	addr.country_id	Country bank located in
	Confirming Bank	Char(10)	lc_head.confirming_bank	
	Confirming Bank Name	Char(240)	partner.partner_desc	Identifies the confirming bank
	Confirming Bank Address 1	Char(240)	addr.add_1	The description from the partner table where partner_id = confirming_bank and partner_type = 'BK'
	Confirming Bank Address 2	Char(240)	addr.add_2	Mandatory line of address
	Confirming Bank Address 3	Char(240)	addr.add_3	Non-mandatory line of address (can be null)
	Confirming Bank City	Char(120)	addr.city	Non-mandatory line of address (can be null)
	Confirming Bank State	Char(3)	addr.state	City bank located in
	Confirming Bank Post Code	Char(30)	addr.post	State, if applicable, where bank located in
	Confirming Bank Country	Char(3)	addr.country_id	Post code, if applicable, where bank located in
	Transferring Bank	Char(10)	lc_head.transferring_bank	Country bank located in
	Transferring Bank Name	Char(240)	partner.partner_desc	Identifies the transferring bank

Table 6-17 (Cont.) File Layout

Record Name	Field Name	Field Type	Default Value	Description
	Transferring Bank Address 1	Char(240)	addr.add_1	The description from the partner table where partner_id = transferring_bank and partner_type = 'BK'
	Transferring Bank Address 2	Char(240)	addr.add_2	Mandatory line of address
	Transferring Bank Address 3	Char(240)	addr.add_3	Non-mandatory line of address (can be null)
	Transferring Bank City	Char(120)	addr.city	Non-mandatory line of address (can be null)
	Transferring Bank State	Char(3)	addr.state	City bank located in
	Transferring Bank Post Code	Char(30)	addr.post	State, if applicable, where bank located in
	Transferring Bank Country	Char(3)	addr.country_id	Post code, if applicable, where bank located in
	Partial Shipment Indicator	Char(1)	lc_head.partial_ship_ind	Country bank located in
	Transshipment Indicator	Char(1)	lc_head.transshipment_ind	Indicates whether goods covered by LC can be partially shipped or not
	Fob Title Pass	Char(6)	lc_head.fob_title_pass	Indicates whether goods can be transferred to another vessel midway through the voyage
	Fob Title Pass Decode	Char(40)	desc is retrieved through a decode	Indicates where the title for goods is passed from the vendor to the purchaser

Table 6-17 (Cont.) File Layout

Record Name	Field Name	Field Type	Default Value	Description
	Fob Title Pass Description	Char(250)	lc_head.ob_title_pass_desc	Decode of where the title for goods is passed from the vendor to the purchaser
	Transportation to	Char(5)	lc_head.transportation_to	Describes the FOB_TITLE_PASS - could be city name and so on
	Transportation to description	Char(150)	outloc.outloc_desc	Transportation to location
	With Recourse Indicator	Char(1)	lc_head.with_recourse_ind	Description of transportation to location
	Latest Shipment Date	Char(14)	lc_head.latest_ship_date	Indicates conditional payment on the part of the bank as instructed by the buyer
	Earliest Shipment Date	Char(14)	lc_head.earliest_ship_date	Latest ship date for all Pos included in the LC, formatted to 'YYYYMMDD HH24MISS'
	Letter of Credit Negotiation Days	Number(3) replaces x in the string "DOCUMENTS TO BE PRESENTED WITHIN x DAYS AFTER ISSUANCE OF THE SHIPPING DOCUMENTS BUT WITHIN THE VALIDITY OF THIS CREDIT"	lc.head.lc_neg_days	The number of days to negotiate documents
	Bank's LC reference id	Number(8)	lc_head.bank_lc_id	Bank's LC ref id
	File Type Record Descriptor	Char(5)	THDCM	Identifies file record type

Table 6-17 (Cont.) File Layout

Record Name	Field Name	Field Type	Default Value	Description
	File Line Sequence Number	Number(10)	line number in file	ID of current line being created for output file
	Transaction Set Control Number	Number(10)	sequence number	Used to force unique file check
	Header Level Comments	Char(2000)	lc_head.comments	Holds any comments that you added to the Letter of Credit.
	File Type Record Descriptor	Char(5)	TDOCS	Identifies file record type
	File Line Sequence Number	Number(10)	line number in file	ID of current line being created for output file
	Transaction Set Control Number	Number(10)	sequence number	Used to force unique file check
	Swift Tag	Char(6)	doc.swift_tag	Identifies individual document types that can be associated with an LC
	Document ID	Number(6)	req_doc.doc_id	Uniquely identifies the individual documents associated with an LC
	Body Text	Char(2000)	req_doc.doc_text	Documents associated with a given LC Description of Goods and Services OR Documents Required OR Additional Conditions OR Narrative
	File Type Record Descriptor	Char(5)	TDETL	Identifies file record type

Table 6-17 (Cont.) File Layout

Record Name	Field Name	Field Type	Default Value	Description
	File Line Sequence Number	Number(10)	line number in file	ID of current line being created for output file
	Transaction Set Control Number	Number(10)	sequence number	Used to force unique file check
	Order Number	Number(8)	lc_detail.order_no	PO associated with the LC
	Item	Char(25)	lc_detail.item	Item on the PO - item is rolled up to the item_level of 1, if possible
	Cost	Number (20,4)	lc_detail.cost	If form_type = 'S'hort then cost is the total cost of the order; if the form_type = 'L'ong then the cost is the unit cost of the item
	Quantity	Number (12,4)	lc_detail.qty	Total qty of the item for the order on the LC
	Standard UOM	Char(4)	Item_master.standard_uom	Standard unit of measure of the quantity of the item for the order on the LC
	Earliest Ship Date	Char(14)	lc_detail.earliest_ship_date	The earliest date an order on the LC can be shipped, formatted to 'YYYYMMDDHH24MISS'
	Latest Ship Date	Char(14)	lc_detail.latest_ship_date	The latest date an order on the LC can be shipped, formatted to 'YYYYMMDDHH24MISS'
	item description	Char(250)	Item_master.description	Item's description
	File Type Record Descriptor	Char(5)	TMERC	Identifies file record type

Table 6-17 (Cont.) File Layout

Record Name	Field Name	Field Type	Default Value	Description
	File Line Sequence Number	Number(10)	line number in file	ID of current line being created for output file
	Transaction Set Control Number	Number(10)	sequence number	Used to force unique file check
	Merchandise Description	Char(2000)	lc_detail.merch_desc	Contains the merchandise description of the field.
	File Type Record Descriptor	Char(5)	TDTCM	Identifies file record type
	File Line Sequence Number	Number(10)	line number in file	ID of current line being created for output file
	Transaction Set Control Number	Number(10)	sequence number	Used to force unique file check
	Detail Level Comments	Char(2000)	lc_detail.comments	Holds any comments that you added to the Letter of Credit detail record.
File Trailer	File Type Record Descriptor	Char(5)	TTAIL	Identifies file record type
	File Line Sequence Number	Number(10)	line number in file	ID of current line being created for output file
	Transaction Set Control Number	Number(10)	sequence number	Used to force unique file check
	Transaction detail line count	Number(10)	ID of current line being created for output file	Sum of the detail lines within a transaction
File Trailer	File Type Record Descriptor	Char(5)	FTAIL	Identifies file record type
	File Line Identifier	Number(10)	Sequential number Created by program.	ID of current line being created for output file.

Table 6-17 (Cont.) File Layout

Record Name	Field Name	Field Type	Default Value	Description
	File Record Counter	Number(10)	N/A	Number of records/ transactions processed in current file (only records between head & tail)

SWIFT File Conversion – Letter of Credit Amendment (lcmt707)

Module Name	lcmt707
Description	SWIFT File Conversion – Letter of Credit Amendment
Functional Area	Oracle Retail Trade Management
Module Type	Integration
Module Technology	Perl
Catalog ID	RMS137
Wrapper Script	rmswrap_perl.ksh

Schedule

Oracle Retail Merchandising Batch Schedule

Design Overview

This Perl script converts the Oracle retail standard interface file format for Amendments to Letters of Credit download to the corresponding S.W.I.F.T file format (MT 707). The input file for this Perl script is the output of the lcmdnld.pc Merchandising batch.

I/O Specification

Integration Type	Download to Merchandising
File Name	Determined by runtime parameter
Integration Contract	IntCon000053 (input) IntCon000138 (output)

Output

The SWIFT MT 707 output file should be in the following format:

- Most output fields are contained in their own line (or 3-4 line for addresses).
- Each amendment consists of only one part, the MT 707. There may be several MT 707s at any given time associated to an LC because they are grouped by amendment number at the time of creation. All TDETL records with the same amend_no will be grouped together in one MT 707.

- Each record starts with a colon and a SWIFT field identifier, followed by another colon: for example, ':40A:'-
- Each amendment is separated by a line with only the ASCII 3 symbol (a heart) on it.

Logic Setup:

The input file will be in standard Merchandising file format. It will potentially have numerous TDETL lines per each THEAD line. There may be numerous TDETL records for one amendment. MT 707 will write one record for each amendment, so if there are multiple TDETL records they need to be combined. There is one TTEXT for each TDETL.

There are three values that need to be calculated. 32B, 33B, 34B. 32B is the total increment or the sum of the positive effect values for each amendment. 33B is the total decrement or the sum of all the negative effect values for each amendment. 32B and 33B are separate totals for each amendment. 34B is the total difference, so it is the sum of the total increment and total decrement. 34B is not just for one amendment though; it is for all amendments of a THEAD record, so this total will run through each TDETL in a THEAD.

For example: if the input file contains:

- THEAD
- TDETL amendment 1, effect +1000
- TTEXT
- TDETL amendment 1, effect +500
- TTEXT
- TDETL amendment 2, effect -2500
- TTEXT
- TDETL amendment 3, effect +4000
- TTEXT
- TDETL amendment 3, effect -1000
- TTEXT
- TDETL amendment 3, effect +500
- TTEXT
- TDETL amendment 4, effect -1000
- TTEXT
- TDETL amendment 4 , effect -2500
- TTEXT
- TTAIL

```
32B for amendment 1 = 1500
33B for amendment 1 = 0
34B for amendemnt 1 = 1500
```

```
32B for amendment 2 = 0
33B for amendment 2 = 2500
34B for amendemnt 2 = -1000
```

```
32B for amendment 3 = 4500
33B for amendment 3 = 1000
```

```
34B for amendemnt 3 = 4500  
  
32B for amendment 4 = 0  
33B for amendment 4 = 3500  
34B for amendemnt 4 = 1000
```

Examples of how individual lines of the M T 707 should look:

```
APPLICANT:  
OPERATOR:  
OPERATION DATE:  
OPERATION TIME:  
TEST KEY:  
BATCH TOTAL:  
SEGMENT TOTAL:  
MT/PRIORITY:707 02  
:27:1/1  
:20:10001981  
:21:1981  
:52D:Bank One  
100 Bank One Way  
Columbus ,OH 41984 US  
:31C:990204  
:30:990204  
:26E:1  
:59:David Fashion Creations P/L Pack  
Wholesale Division  
109 Ackland St.  
St. Kilda ,VA 30280-1234 US  
:32B:USD500,0  
:33B:USD0,0  
:34B:USD500,0  
:79:Letter of Credit: has been changed from 25 to 30  
for Style 10049369, resulting in an effect of 500  
(USD).
```

The layout of the S.W.I.F.T MT 707 (Amendment to a Documentary Credit) file is as follows:

SWIFT I.D. DATA TYPE CODES (refer to SWIFT User Handbook – Standards General Information – October 1998 release for formatting information):

 **Note:**

The field lengths and types in the Oracle Retail Standard Download Format of the MT 707 are important because sometimes they are different from the information that is being placed in them and the fields may have to be truncated, rounded, and so on.

There is always a new line (nl) after every individual SWIFT ID (and there may be more than one line within an individual field (example 59 - Beneficiary, four lines to hold address information)).

In some situations, certain fields will be blank. These fields should be skipped over. In other words, no blank line or tag should be printed indicating the field is blank. Simply ignore it.

SWIFT File Conversion - Letter of Credit Application (lcmt700)

Module Name	lcmt700
Description	SWIFT File Conversion – Letter of Credit Application
Functional Area	Oracle Retail Trade Management
Module Type	Integration
Module Technology	Perl
Catalog ID	RMS136
Wrapper Script	rmswrap_perl.ksh

Schedule

Oracle Retail Merchandising Batch Schedule

Design Overview

This Perl script will convert the standard Merchandising flat file into the bank specific S.W.I.F.T. MT 700 output files. The input file for this Perl script is the output of the lcadnld.pc Merchandising batch. One output file will be created for each issuing bank in the lcadnld.pc output file.

I/O Specification

Integration Type	Download from Merchandising
File Name	Determined by runtime parameter
Integration Contract	IntCon000052 (input) IntCon000137 (output)

Output

All files layouts input and output the SWIFT MT 700. The output file should be in the following format:

- Most output fields are contained in their own line (or 3-4 line for addresses).
- Each application consists of four parts, one MT 700 and three MT 701s, which are ordered through the Sequence of Total field: for example, ':27:1/4 MT 700' is the first (MT 700) part of the application.
- MT 700 and MT 701s will be mingled in the same file.
- Each record starts with a colon and a SWIFT field identifier, followed by another colon: for example, ':40A:'-
- Each application is separated by a line with only the ASCII 3 symbol (a heart) on it.

Examples of how individual lines of the MT 700 or MT 701 should look:

```
:27:1/4
:40A:IRREVOCABLE
:20:29893098
:23:NOREF
:31C:910906
:31D:911022DALLAS
```

```
:51D:NORTHERN TRUST INT'L BANKING CORP.  
      ONE WORLD TRADE CENTER  
SUITE 3941  
NY, NY 10048 USA
```

The layout of the S.W.I.F.T MT 700 (Issue of a Documentary Credit) file is as follows:

SWIFT I.D. DATA TYPE CODES (refer to SWIFT User Handbook - Standards general Information - October 1998 release for formatting information):

 **Note:**

There is always a new line (nl) after every individual SWIFT ID (and there may be more than one line within an individual field [for example, 59 – Beneficiary, four lines to hold address information]).

In some situations, certain fields will be blank. These fields should be skipped over. In other words, no blank line or tag should be printed indicating the field is blank. Simply ignore it.

Invoices

Merchandising has scheduled integration for the following invoices for communication with Oracle Retail Invoice Matching Cloud Service:

- [Download of Invoice for Invoice Matching \(edidlinv\)](#)
- [Stage Complex Deal Invoice Information \(vendinvc\)](#)
- [Stage Fixed Deal Invoice Information \(vendinvf\)](#)

Download of Invoice for Invoice Matching (edidlinv)

Module Name	edidlinv.pc
Description	Download of Invoice For Invoice Matching
Functional Area	Invoice Matching
Module Type	Integration
Module Technology	ProC
Catalog ID	RMS127
Wrapper Script	rmswrap_multi_out.ksh

Schedule

Oracle Retail Merchandising Batch Schedule

Design Overview

The EDIDLINV program extracts invoice information from Merchandising invoice tables (INVC_HEAD, INVC_DETAIL) to a flat file. This flat file is used by Invoice Matching to upload invoice data into tables such as IM_DOC_HEAD,

IM_INVOICE_DETAIL and IM_DOC_NON_MERCH. This batch program is run daily, extracting invoice records whose invoice date falls on the current vdate.

If the batch is run ad hoc, there may be issues when consignment invoices are generated as the sales process can also run multiple times a day. Invoice information can potentially be not the latest when the extract is generated.

Restart/Recovery

Restart/recovery for this program is set up at the invoice ID and line sequence level. The program resumes writing to file starting on the next line where the previous process ended.

I/O Specification

Integration Type	Download from Merchandising
File Name	Determined by runtime parameter
Integration Contract	IntCon000024

Output File Layout

Table 6-18 edidlinv.pc - Output File Layout

Record Name	Field Name	Field Type	Default Value	Description
FHEAD	Record descriptor	Char(5)	FHEAD	Describes file record type. Valid value is FHEAD.k
	Line id	Number(10)	0000000001	Sequential file line number.
	Gentran ID	Char(5)	UPINV	The type of transaction this file represents. Valid value is UPINV
	Current date	Char(14)	N/A	Vdate in YYYYMMDDHH24 MISS format.
	File Version	Number(2)	03	Numeric number of the file version to backward compatibility. Starting with 2, incremented by 1 each time the file format changes
THEAD	Record descriptor	Char(5)	N/A	Describes file record type. Valid value is THEAD.
	Line id	Number (10)	N/A	Sequential file line number.

Table 6-18 (Cont.) edidlinv.pc - Output File Layout

Record Name	Field Name	Field Type	Default Value	Description
	Transaction number	Number(10)	N/A	Sequential transaction number. All records within this transaction will also have this transaction number.
	Document Type	Char(6)	N/A	Describes the type of document being uploaded. The document type will determine the types of detail information that are valid for the document upload. Invoice types are held on the codes table under a code type of 'IMIT'.
	Vendor Document Number	Char (50)	N/A	Vendor's document number.
	Group ID	Char(10)	NULL	The Group ID is an informational field, which can be used to identify groups of invoices that were transmitted to Invoice Matching together. This is not populated by Merchandising.

Table 6-18 (Cont.) edidlinv.pc - Output File Layout

Record Name	Field Name	Field Type	Default Value	Description
	Vendor Type	Char(6)	N/A	Type of vendor (either supplier or partner) for this document. Valid values include Bank 'BK', Agent 'AG', Freight Forwarder 'FF', Importer 'IM', Broker 'BR', Factory 'FA', Applicant 'AP', Consolidator 'CO', Consignee 'CN', Supplier Hierarchy Level 1 'S1', Supplier Hierarchy Level 2 'S2', and Supplier Hierarchy Level 3 'S3'. These partner types will be held on the codes table under the code_type 'PTAL'.
	Vendor ID	Char(10)	N/A	Vendor for this document.
	Vendor Document Date	Char(14)	N/A	Date document was issued by the vendor (in YYYYMMDD24MIS Sformat).
	Order Number / RTV order number	Number(12)	N/A	Merchandising system order number for this document. Required for merchandise invoices and optional for others. This field can also contain the RTV order number if the RTV flag is 'Y'
	Location	Number(10)	N/A	Merchandising system location for this document.

Table 6-18 (Cont.) edidlinv.pc - Output File Layout

Record Name	Field Name	Field Type	Default Value	Description
	Location Type	Char(1)	N/A	Merchandising system location type (either 'S'tore or 'W'arehouse) for this document. Required for merchandise invoices and optional for others.
	Terms	Char(15)	N/A	Terms of this document. If terms are not provided, the vendor's default terms will be associated with this record.
	Due Date	Char(14)	N/A	Terms of this document. If terms are not provided, the vendor's default terms will be associated with this record.
	Payment method	Char(6)	N/A	Method for paying this document.
	Currency code	Char(3)	N/A	Currency code for all monetary amounts on this document.
	Exchange rate	Number(20,4)	N/A	Exchange rate *10000 (implied 4 decimal places) for conversion of document currency to the primary currency.
	Sign Indicator	Char(1)	N/A	Indicates either a positive (+) or a negative (-) total cost amount.
	Total Cost	Number(20,4)	N/A	Total document cost *10000 (implied 4 decimal places), including all items and costs on this document. This value is in the document currency.

Table 6-18 (Cont.) edidlinv.pc - Output File Layout

Record Name	Field Name	Field Type	Default Value	Description
	Sign Indicator	Char(1)	N/A	Indicates either a positive (+) or a negative (-) total vat amount.
	Total VAT Amount	Number(20,4)	N/A	Total VAT amount *10000 (implied 4 decimal places), including all items and costs on this document. This value is in the document currency.
	Sign Indicator	Char(1)	N/A	Indicates either a positive (+) or a negative (-) total quantity amount.
	Total Quantity	Number(12,4)	N/A	Total quantity of items *10000 (implied 4 decimal places) on this document. This value is in EACHES (no other units of measure are supported in Invoice Matching).
	Sign Indicator	Char(1)	N/A	Indicates either a positive (+) or a negative (-) total discount amount.
	Total Discount	Number(12,4)	N/A	Total discount *10000 (implied 4 decimal places) applied to this document. This value is in the document currency.
	Freight Type	Char(6)	NULL	The freight method for this document. Always blank.
	Paid Ind	Char(1)	N/A	Indicates if this document has been paid.
	Multi-Location	Char(1)	N/A	Indicates if this invoice goes to multiple locations.
	Merchandise Type	Char(1)	N/A	Indicates if this invoice is a consignment invoice.

Table 6-18 (Cont.) edidlinv.pc - Output File Layout

Record Name	Field Name	Field Type	Default Value	Description
	Deal Id	Number(10)	NULL	Deal Id from Merchandising if this invoice is a deal bill back invoice. Always blank
	Deal Detail Id	Char(10)	NULL	Complex Deal Component Id. Always blank from Merchandising.
	Ref CNR Ext Doc Id	Char(50)	NULL	Reference to the External Id of Credit Note Request associated with this document. Always blank from Merchandising.
	Ref INV Ext Doc Id	Char(50)	NULL	Reference to the External Id of Invoice associated with this document. Always blank from Merchandising.
	Deal Approval Indicator	Char(1)	NULL	Indicates if the document on IM_DOC_HEAD is to be created in Approved or Submitted status. Always blank from Merchandising.
	RTV indicator	Char(1)	N/A	Indicates if this invoice is a RTV invoice.
	Custom Document Reference 1	Char(90)	NULL	This optional field is included in the upload file for client customization. No validation will be performed on this field. Always blank from Merchandising.
	Custom Document Reference 2	Char(90)	NULL	This optional field is included in the upload file for client customization. No validation will be performed on this field. Always blank from Merchandising.

Table 6-18 (Cont.) edidlinv.pc - Output File Layout

Record Name	Field Name	Field Type	Default Value	Description
	Custom Document Reference 3	Char(90)	NULL	This optional field is included in the upload file for client customization. No validation will be performed on this field. Always blank from Merchandising.
	Custom Document Reference 4	Char(90)	NULL	This optional field is included in the upload file for client customization. No validation will be performed on this field. Always blank from Merchandising.
	Cross-reference document number	Number(10)	N/A	Document that a credit note is for. Blank for all document types other than merchandise invoices.
	Third-Party Payee	Number(10)	NULL	This optional field is included in the upload file for client customization. No validation will be performed on this field. Always blank from Merchandising.
TDETL	Record descriptor	Char(5)	N/A	Describes file record type. Valid value is TDETL
	Line id	Number(10)	N/A	Sequential file line number.
	Transaction number	Number(10)	N/A	Transaction number for this item detail record.
	UPC	Char(25)	NULL	UPC for this detail record. Valid item number will be retrieved for the UPC. Always blank from Merchandising.

Table 6-18 (Cont.) edidlinv.pc - Output File Layout

Record Name	Field Name	Field Type	Default Value	Description
	UPC Supplement	Number(5)	NULL	Supplement for the UPC. Always blank from Merchandising.
	Item	Char(25)	N/A	Item for this detail record.
	VPN	Char(30)	NULL	Vendor Product Number which can (optionally) be used instead of the Oracle Retail Item Number.
	Sign Indicator	Char(1)	N/A	Indicates either a positive (+) or a negative (-) Original Document Quantity amount.
	Original Document Quantity	Number(12,4)	N/A	Quantity *10000 (implied 4 decimal places), in EACHES, of the item on this detail record.
	Sign Indicator	Char(1)	N/A	Indicates either a positive (+) or a negative (-) Original Unit Cost amount.
	Original Unit cost	Number(20,4)	N/A	Unit cost *10000 (implied 4 decimal places), in document currency, of the item on this detail record
	Original VAT Code	Char (6)	N/A	VAT code for item.
	Original VAT rate	Number (20,10)	N/A	VAT Rate for the VAT code/item.
	Sign Indicator	Char(1)	N/A	Indicates either a positive (+) or a negative (-) total allowance. Default is "+" if no allowances exist for this detail record.

Table 6-18 (Cont.) edidlinv.pc - Output File Layout

Record Name	Field Name	Field Type	Default Value	Description
	Total Allowance	Number(20,4)	N/A	Sum of allowance details for this item detail record *10000 (implied 4 decimal places). If no allowances exist for this item detail record, value will be 0.
TNMRC	Record descriptor	Char(5)	N/A	Describes file record type.
	Line id	Number (10)	N/A	Sequential file line number.
	Transaction number	Number(10)	N/A	Transaction number for this non-merchandise record.
	Non Merchandise Code	Char(6)	N/A	Non-Merchandise code that describes this cost.
	Sign Indicator	Char(1)	N/A	Indicates either a positive (+) or a negative (-) Non Merchandise Amt.
	Non Merchandise Amt	Number(20,4)	N/A	Cost *10000 (implied 4 decimal places) in the document currency.
	Non Merch VAT Code	Char (6)	N/A	VAT Code for Non-Merchandise.
	Non Merch Vat Rate at this VAT code	Number (20, 10)	N/A	VAT Rate corresponding to the VAT code.
	Service Performed Indicator	Char(1)	N/A	Indicates if a service has actually been performed.
	Store	Number(10)	N/A	Store at which the service was performed.
TVATS	File record descriptor	Char(5)		Marks costs at VAT rate line. Valid value is TVATS.
	Line id	Char(10)		Sequential file line number.
	Transaction number	Number(10)		Transaction number for this vat detail record.
	VAT code	Char(6)		VAT code that applies to cost.

Table 6-18 (Cont.) edidlinv.pc - Output File Layout

Record Name	Field Name	Field Type	Default Value	Description
	VAT rate	Number (20,10)		VAT Rate corresponding to the VAT code.
	Sign Indicator	Char(1)		Indicates either a positive (+) or a negative (-) Original Document Quantity amount.
	Cost at this VAT code	Number (20,4)		Total amount *10000 (implied 4 decimal places) that must be taxed at the above VAT code.
TTAIL	Record descriptor	Char(5)	N/A	Describes file record type. Default value is TTAIL.
	Line id	Number(10)	N/A	Sequential file line number.
	Transaction number	Number(10)	N/A	Transaction number for the transaction that this record is closing.
	Transaction lines	Number(6)	N/A	Total number of detail lines within this transaction.
FTAIL	Record descriptor	Char(5)	N/A	Describes file record type.
	Line id	Number(10)	N/A	Sequential file line number.
	Number of lines	Number(10)	N/A	Total number of lines within this file excluding FHEAD and FTAIL.

Design Assumptions

N/A

Stage Complex Deal Invoice Information (vendinvc)

Module Name vendinvc.pc
Description Stage Complex Deal Invoice Information
Functional Area Deals
Module Type Integration
Module Technology ProC

Catalog ID	RMS122
Wrapper Script	rmswrap.ksh

Schedule

Oracle Retail Merchandising Batch Schedule

Design Overview

The batch module creates records in invoice match staging tables dealing for complex type deals.

The invoicing logic will be driven from the billing period estimated next invoice date for complex deals. The amount to be invoiced will be the sum of the income accruals of the deal since the previous invoice date (or the deal start date for the first collection).

prepost vendinvc pre - truncates STAGE_COMPLEX_DEAL_HEAD and STAGE_COMPLEX_DEAL_DETAIL tables to remove previous days records.

prepost vendinvc post - calls the process_deal_head() function to update est_next_invoice_date of the deal to NULL.

Restart/Recovery

When the max commit point is reached, the data is updated.

I/O Specification

Integration Type	Download from Merchandising
File Name	N /A
Integration Contract	IntCon000009

Records are written to the stage_complex_deal_head and stage_complex_deal_detail tables.

Design Assumptions

N/A

Stage Fixed Deal Invoice Information (vendinvf)

Module Name	vendinvc.pc
Description	Stage Complex Deal Invoice Information
Functional Area	Deals
Module Type	Integration
Module Technology	ProC
Catalog ID	RMS123
Wrapper Script	rmswrap_multi.ksh

Schedule

Oracle Retail Merchandising Batch Schedule

Design Overview

The batch module creates records in staging tables dealing for fixed type deals.

The invoicing logic will be driven by the collection dates for fixed deals. The amount to be invoiced will be retrieved directly from fixed deal tables for a given deal date.

prepost vendinvf pre - truncates STAGE_FIXED_DEAL_HEAD and STAGE_FIXED_DEAL_DETAIL tables to remove previous days records.

prepost vendinvf post – calls the process_fixed_deal function to update the status of the fixed deal claim to 'I' (inactive)

Restart/Recovery

Data is committed to the database once the number of transactions processed reaches or exceeds the max_commit_ctr.

I/O Specification

Integration Type	Download from Merchandising
File Name	N /A
Integration Contract	IntCon000009

Records are written to the stage_complex_deal_head and stage_complex_deal_detail tables.

Design Assumptions

N/A

Inventory

Merchandising has scheduled integration for inventory and sales data via the following processes:

- [Download Sales and Stock on Hand to Suppliers \(edidlprd\)](#)
- [Future Available Inventory Publication API \(BDI_COFutureAvail_Tx_PF_From_RMS_JOB\)](#)
- [Inventory Publication API \(BDI_Inventory_Tx_PF_From_RMS_EOW_JOB\)](#)
- [Item Location History \(BDI_ItemLocHist_Tx_PF_From_RMS_JOB\)](#)
- [Reject POSU Transactions \(salesgenrej.ksh\)](#)
- [Store Available Inventory Publication API \(BDI_InvAvailStore_Tx_PF_From_RMS_JOB\)](#)
- [Warehouse Inventory Publication API \(BDI_InvAvailWh_Tx_PF_From_RMS_JOB\)](#)

Download Sales and Stock on Hand to Suppliers (edidlprd)

Module Name	edidlprd.pc
Description	Download Sales and Stock On Hand to Suppliers
Functional Area	Inventory
Module Type	Integration
Module Technology	ProC
Catalog ID	RMS47
Wrapper Script	rmswrap_multi_out.ksh

Schedule

Oracle Retail Merchandising Batch Schedule

Design Overview

This program is used to transmit item-level sales and stock-on-hand information to vendors. The report is a summary that will be sent to specified suppliers through EDI, giving sales details, as well as current stock on hand and in transit for all locations for each of the items supplied by that supplier. Only those suppliers which have an EDI sales reporting frequency of either daily or weekly will have files generated by this program. The system parameter EDI Daily Report Lag is used for suppliers receiving daily updates to determine the day lag for sales data sent, to account for late posting sales.

Restart/Recovery

Restart/recovery in this program is achieved through utilizing a global temporary table. Once a supplier is processed, it is deleted from the temporary table to prevent the same supplier from being processed again during recovery.

I/O Specification

Integration Type	Download from Merchandising
File Name	Determined by runtime parameter
Integration Contract	IntCon000013

Output File Layout

Table 6-19 edidlprd.pc - Output File

Record Name	Field Name	Field Type	Default Value	Description
FHEAD	File record descriptor	Char(5)	FHEAD	Describes record type
	Line number	Number(10)	0000000001	Sequential file line number
	File source	Char(5)	DLPRD	File Type

Table 6-19 (Cont.) edidlprd.pc - Output File

Record Name	Field Name	Field Type	Default Value	Description
THEAD	File create date	Char(8)	N/A	Date that the file was created in YYYYMMDD format
	File record descriptor	Char(5)	THEAD	Identifies record type
	Line number	Number(10)	N/A	Sequential file line number
	Transaction number	Number(10)	N/A	Sequential transaction number
	Report date	Char(8)	N/A	For weekly reporting, this will contain the current date. For daily reporting, it will be the date represented by the sales, current date – lag days. Both will be in the YYYYMMDD format
TITEM	Supplier	Number(10)	N/A	Merchandising Supplier Number
	File record descriptor	Char(5)	TITEM	Identifies file record type
	Line number	Number(10)	N/A	Sequential file line number
	Transaction number	Number(10)	N/A	Sequential transaction number
	Item	Char(25)	N/A	Transaction level item to which with the data is related
	Item_Num_Type	Char(6)	N/A	Contains the item number type for the item on ITEM_MASTER
	Ref_Item	Char(25)	N/A	Contains the primary reference item for the item in the file, if defined
	Ref_Item_Num_Type	Char(6)	N/A	Contains the item number type for the reference item from ITEM_MASTER
Vendor catalog number	Char(30)	N/A	Contains the VPN (Vendor Product Number), if defined for the item/supplier	

Table 6-19 (Cont.) edidlprd.pc - Output File

Record Name	Field Name	Field Type	Default Value	Description
TQUTY	Item description	Char(250)	N/A	Contains the transaction level item description from ITEM_MASTER
	File record descriptor	Char(5)	TQUTY	Identifies record type
	Line number	Number(10)	N/A	Sequential file line number
	Transaction number	Number(10)	N/A	Sequential transaction number
	Quantity descriptor	Char(15)	N/A	Indicates what the quantity represents, either 'On-hand' (stock), 'Sold'(sales), or 'In transit'
	Location type	Char(2)	N/A	Indicates the type of location represented in the file: 'ST' for store or 'WH' warehouse
	Location	Number(10)	N/A	Contains the store or warehouse number for which the information applies
	Unit cost	Number(20)	N/A	Contains the current unit cost for the item/ location with 4 implied decimal places. This value will be in the supplier's currency
TTAIL	Quantity	Number(12)	N/A	Indicates the quantity of the item sold, on hand or in transit to the location; the quantity is represented with 4 implied decimal places
	File record descriptor	Char(5)	TTAIL	Identifies record type
	Line number	Number(10)	N/A	Sequential file line number
FTAIL	Transaction lines	Number(6)	N/A	Number of lines for this transaction
	File record descriptor	Char(5)	FTAIL	Identifies record type
	Line number	Number(10)	N/A	Total number of lines in file

Table 6-19 (Cont.) edidlprd.pc - Output File

Record Name	Field Name	Field Type	Default Value	Description
	Number of transaction lines	Number(10)	N/A	Number of transaction lines in file

Design Assumptions

A data translator will be used to convert the flat file produced by Merchandising to the required EDI data format.

Only data for items where the supplier is indicated as the primary supplier/origin country for the item will be included in the report.

Future Available Inventory Publication API (BDI_COFutureAvail_Tx_PF_From_RMS_JOB)

This section describes the Future Available Inventory Publication BDI.

Functional Area

Inventory

Design Overview

BDI (Bulk Data Integration) is an integration layer that facilitates the bulk transfer of on-order quantity for all item/location combinations that are flagged as back-orderable in Merchandising to other Oracle Retail Applications. On this particular integration stream, the data flow is from Merchandising to BDI, and then BDI to downstream applications. To accomplish this data transfer, BDI will be calling a Merchandising-owned API that will pull data from Merchandising and deliver these to the BDI integration layer. This API will be in the form of a PLSQL function inside a PLSQL package.

Package Impact

The following packages are impacted:

Bulk Interface Module

In the bulk interface module:

Filename: bdiavinvb.pls

```
BDI_AV_INV_SQL.CO_FUTURE_AVAIL_UP(O_error_message IN OUT VARCHAR2,
                                   O_control_id     IN OUT NUMBER,
                                   I_job_context    IN     VARCHAR2)
```

This function begins by calling a BDI function that signals the start of the interface process. The BDI function will update the internal BDI control tables to track the progress of the API.

A DML insert statement is then executed to populate the BDI outbound table that resides in the BDI_RMS_INT_SCHEMA schema. This outbound table is loaded with records from the Merchandising Item Inventory tables/view.

After the insert, another call to a BDI function is performed to signify the successful loading of records. This will update the internal BDI control tables.

A database commit is issued, and the control Id is returned by the API.

Data Definition XML

The BDI interface staging tables are generated based on the XML schema definition.

Data Flow	Description	XML Schema Definition (XSD)
CO Future Avail	CO Future Availability	COFutureAvail_Tx_BdiInterfaceModule .xml

Tables

TABLE	SELECT	INSERT	UPDATE	DELETE
CO_FUTURE_AVAIL_OUT	No	Yes	No	No
V_BDI_CO_FUTURE_AVAIL	Yes	No	No	No

Inventory Publication API (BDI_Inventory_Tx_PF_From_RMS_EOW_JOB)

This section describes the Item Inventory Publication BDI.

Functional Area

Inventory

Business Overview

BDI (Bulk Data Integration) is an integration layer that facilitates the bulk transfer of inventory from Merchandising to other Oracle Retail Applications. On this particular integration stream, the data flow is from Merchandising to BDI, and then BDI to downstream applications. To accomplish this data transfer, BDI will be calling a Merchandising-owned API that will pull data from Merchandising and deliver these to the BDI integration layer. This API will be in the form of a PLSQL function inside a PLSQL package.

Package Impact

Filename: bdimfpb.pls

```
BDI_MFP_SQL.INVENTORY_UP(O_error_message IN OUT RTK_ERRORS.RTK_TEXT%TYPE,
                          O_control_id    IN OUT NUMBER,
                          I_job_context   IN    VARCHAR2)
```

This function begins by calling a BDI function that signals the start of the interface process. The BDI function will update the internal BDI control tables to track the progress of the API.

A DML insert statement is then executed to populate the BDI outbound table that resides in the BDI_RMS_INT_SCHEMA schema. This outbound table is loaded with records from the Merchandising Item Inventory tables/view.

After the insert, another call to a BDI function is performed to signify the successful loading of records. This will update the internal BDI control tables.

A database commit is issued, and the control Id is returned by the API.

Data Definition XML

The BDI interface staging tables are generated based on the XML schema definition

Data Flow	Description	XML Schema Definition (XSD)
Inventory	Inventory upload to BDI	Inventory_Tx_BdiInterfaceModule.xml

Table Impact

TABLE	SELECT	INSERT	UPDATE	DELETE
INVENTORY_OUT	No	Yes	No	No
V_BDI_MFP_INVENTORY	Yes	No	No	No

Item Location History (BDI_ItemLocHist_Tx_PF_From_RMS_JOB)

Module Name	BDI_ItemLocHist_Tx_PF_From_RMS_JOB
Description	Extracts Sales History
Functional Area	Sales
Module Type	Integration
Module Technology	BDI job
Catalog ID	N/A
Runtime Parameters	ItemLocHist_Tx_ProcessFlow_From_RMS ItemLocHist_Tx_Extractor

Design Overview

Merchandising extracts item-location sales history on a weekly basis. It utilizes BDI (Bulk Data Integration) to facilitate the bulk data movement from Merchandising to an external solution.

Scheduling Constraints

Schedule Information	Description
Processing Cycle	End of Day
Frequency	Scheduled daily but files will only be generated weekly on End of Week date.
Scheduling Considerations	N/A
Pre-Processing	N/A
Post-Processing	N/A
Threading Scheme	N/A

Restart/Recovery

N/A

Key Tables Affected

Table	Select	Insert	Update	Delete
ITEM_LOC_HIST	Yes	No	No	No
ITEM_LOC_HIST_OUT	Yes	Yes	No	Yes
BDI_DWNLDR_IFACE_MOD_DATA_CTL	Yes	No	No	No
BDI_DWNLDR_IFACE_DATA_CTL	Yes	No	No	No

Integration Contract

Refer to ItemLocHist_Tx_BdiInterfaceModule.xml.

Reject POSU Transactions (salesgenrej.ksh)

Module Name	salesgenrej.ksh
Description	Reject POSU Transactions
Functional Area	Sales Posting
Module Type	Business Processing
Module Technology	KSH
Catalog ID	RMS338
Wrapper Script	batch_salesgenrej.ksh

Schedule

Oracle Retail Merchandising Batch Schedule

Design Overview

The purpose of this module is to archive the rejected transactions and create a reject file based on the recently processed POSU file which is still in the staging table. It will also generate a retry file based on input parameter (Retry Indicator¹) if error was due to locking and if the number of attempts does not exceed the retry lock attempt configuration (RMS_PLSQL_BATCH_CONFIG.RETRY_LOCK_ATTEMPT).

Restart/Recovery

N/A

Performance Considerations

The number of threads, the amount of waiting time, number for retries, and average volume of data should be considered. RETRY_WAIT_TIME shouldn't be increased significantly.

Reject File:

The module will have the ability to re-process the reject file directly. The file format will therefore be identical to the input file layout. A reject line counter will be kept in the program

¹ Input parameter is not required and is defaulted to N that will disable the generate the retry file functionality for locked records.

and is required to ensure that the file line count in the trailer record matches the number of rejected records. If no errors occur, no reject files would be generated.

Retry File:

If the retry input indicator is set to Y, then a retry file is going to be generated if the error is due to locking only. This will be automatically be placed at the input directory ready to be picked up by the next sales upload and processing.

Store Available Inventory Publication API (BDI_InvAvailStore_Tx_PF_From_RMS_JOB)

This section describes the Store Available Inventory Publication BDI.

Functional Area

Foundation

Business Overview

BDI (Bulk Data Integration) is an integration layer that facilitates the bulk transfer of Store Address information from Merchandising to other Oracle Retail Applications. On this particular integration stream, the data flow is from Merchandising to BDI, and then BDI to downstream applications. To accomplish this data transfer, BDI will be calling a Merchandising-owned API that will pull data from Merchandising and deliver these to the BDI integration layer. This API will be in the form of a PLSQL function inside a PLSQL package.

Package Impact

This section describes the package impact.

Bulk Interface Module

Filename: bdiavinvb.pls

```
BDI_AV_INV_SQL.ST_AVAIL_INV_UP(O_error_message IN OUT VARCHAR2,
                               O_control_id    IN OUT NUMBER,
                               I_job_context   IN    VARCHAR2)
```

This function begins by calling a BDI function that signals the start of the interface process. The BDI function will update the internal BDI control tables to track the progress of the API.

A DML insert statement is then executed to populate the BDI outbound table that resides in the BDI_RMS_INT_SCHEMA schema. This outbound table is loaded with records from the Merchandising Merchandise Hierarchy tables.

After the insert, another call to a BDI function is performed to signify the successful loading of records. This will update the internal BDI control tables.

A database commit is issued, and the control Id is returned by the API.

Data Definition XML

The BDI interface staging tables are generated based on the XML schema definition

Data Flow	Description	XML Schema Definition (XSD)
Store Inventory	Store inventory upload to BDI	InvAvailStore_Tx_BdiInterfaceModule.xml

Table Impact

TABLE	SELECT	INSERT	UPDATE	DELETE
INV_AVAIL_STORE_OUT	No	Yes	No	No
ITEM_MASTER	Yes	No	No	No
ITEM_LOC_SOH	Yes	No	No	No
STORE	Yes	No	No	No

Warehouse Inventory Publication API (BDI_InvAvailWh_Tx_PF_From_RMS_JOB)

This section describes the Warehouse Inventory Publication BDI.

Functional Area

Foundation

Business Overview

BDI (Bulk Data Integration) is an integration layer that facilitates the bulk transfer of Warehouse Inventory positions from Merchandising to other Oracle Retail Applications. On this particular integration stream, the data flow is from Merchandising to BDI, and then BDI to downstream applications. To accomplish this data transfer, BDI will be calling a Merchandising-owned API that will pull data from Merchandising and deliver these to the BDI integration layer. This API is in the form of a PLSQL function inside a PLSQL package.

Package Impact

This section describes the package impact.

Bulk Interface Module

Filename: bdiavinvb.pls

```
BDI_AV_INV_SQL.WH_AVAIL_INV_UP(O_error_message IN OUT VARCHAR2,
                                O_control_id    IN OUT NUMBER,
                                I_job_context   IN      VARCHAR2)
```

This function begins by calling a BDI function that signals the start of the interface process. The BDI function will update the internal BDI control tables to track the progress of the API.

A DML insert statement is then executed to populate the BDI outbound table that resides in the BDI_RMS_INT_SCHEMA schema. This outbound table is loaded with records from the Merchandising Item Location table.

After the insert, another call to a BDI function is performed to signify the successful loading of records. This will update the internal BDI control tables.

A database commit is issued, and the control Id is returned by the API.

Data Definition XML

The BDI interface staging tables are generated based on the XML schema definition.

Data Flow	Description	XML Schema Definition (XSD)
Warehouse Inventory Avail	Wh Available Inventory	InvAvailWh_Tx_BdiInterfaceModule.xml

Table Impact

TABLE	SELECT	INSERT	UPDATE	DELETE
INV_AVAIL_WH_OUT	No	Yes	No	No
ITEM_MASTER	Yes	No	No	No
ITEM_LOC_SOH	Yes	No	No	No
WH	Yes	No	No	No

Planning and Forecasting

Merchandising provides critical foundation and transactional information to the Oracle Retail planning and forecasting solutions. Because the planning and forecasting solutions are built on the same platform, several of the integrations from Merchandising are used by more than one of the solutions. The tables below summarize the key outbound integration points by solution.

Table 6-20 Integration to Oracle Retail Merchandise Financial Planning Cloud Service (MFPCS)

Description	Program
Calendar Extract to Planning and Forecasting	BDI_RPAS_Calendar_Fnd_PF_From_RMS_JOB
Currency Rates Extract to Planning and Forecasting	BDI_RPAS_CurrConvRates_Fnd_PF_From_RMS_JOB
Inventory Extract to Planning	BDI_MFP_Inventory_Tx_PF_From_RMS_JOB
Merchandise Hierarchy and Item Extract to Planning and Forecasting	BDI_RPAS_MerchHier_Fnd_PF_From_RMS_JOB
On Order Extract to Planning	BDI_MFP_OnOrder_Tx_PF_From_RMS_JOB
Organization Hierarchy Extract to Planning and Forecasting	BDI_RPAS_OrgHier_Fnd_PF_From_RMS_JOB
Store Extract to Planning and Forecasting	BDI_RPAS_Store_Fnd_PF_From_RMS_JOB
Transaction Data Extract to Planning	BDI_MFP_TransData_Tx_PF_From_RMS_JOB

Table 6-21 Integration to Oracle Retail Assortment and Item Planning for Fashion/ Softlines Cloud Service (A&IP CS)

Description	Program
Brand Extract to Planning	BDI_RPAS_Brand_Fnd_PF_From_RMS_JOB
Calendar Extract to Planning and Forecasting	BDI_RPAS_Calendar_Fnd_PF_From_RMS_JOB
Currency Rates Extract to Planning and Forecasting	BDI_RPAS_CurrConvRates_Fnd_PF_From_RMS_JOB
Differentiator Extract to Planning	BDI_RPAS_Diff_Fnd_PF_From_RMS_JOB
Inventory Extract to Planning	BDI_MFP_Inventory_Tx_PF_From_RMS_JOB
Merchandise Hierarchy and Item Extract to Planning and Forecasting	BDI_RPAS_MerchHier_Fnd_PF_From_RMS_JOB
On Order Extract to Planning	BDI_MFP_OnOrder_Tx_PF_From_RMS_JOB
Organization Hierarchy Extract to Planning and Forecasting	BDI_RPAS_OrgHier_Fnd_PF_From_RMS_JOB
Store Extract to Planning and Forecasting	BDI_RPAS_Store_Fnd_PF_From_RMS_JOB
Supplier Extract to Planning	BDI_RPAS_Supplier_Fnd_PF_From_RMS_JOB
Transaction Data Extract to Planning	BDI_MFP_TranData_Tx_PF_From_RMS_JOB
UDA Extract to Planning	BDI_RPAS_UdaAndUdaValues_Fnd_PF_From_RMS_JOB
UDA Item Extract to Planning and Forecasting	BDI_RDF_UdaltemLov_Fnd_From_RMS_JOB

Table 6-22 Integration to Oracle Retail Demand Forecasting Cloud Service

Description	Program
Calendar Extract to Planning and Forecasting	BDI_RPAS_Calendar_Fnd_PF_From_RMS_JOB
Merchandise Hierarchy and Item Extract to Planning and Forecasting	BDI_RPAS_MerchHier_Fnd_PF_From_RMS_JOB
Organization Hierarchy Extract to Planning and Forecasting	BDI_RPAS_OrgHier_Fnd_PF_From_RMS_JOB
Out of Stock Extract to Forecasting	BDI_RDF_StockOut_Tx_PF_From_RMS_JOB
Store Extract to Planning and Forecasting	BDI_RPAS_Store_Fnd_PF_From_RMS_JOB
UDA Item Extract to Planning and Forecasting	BDI_RDF_UdaltemLov_Fnd_From_RMS_JOB
Weekly Sales Extract to Forecasting	BDI_RDF_WeeklySales_Tx_PF_From_RMS_JOB

Brand Extract to Planning (BDI_RPAS_Brand_Fnd_PF_From_RMS_JOB)

Module Name	BDI_RPAS_Brand_Fnd_PF_From_RMS_JOBbdi_merch_extract_to_file_wrapper.shbdi_rpas_brand_extract.ksh
Description	Extracts Brand information to Planning
Functional Area	Foundation
Module Type	Integration
Module Technology	BDI job, shell scripts
Catalog ID	N/A
Runtime Parameters	Brand_Fnd_ProcessFlow_From_RMSBrand_Fnd_ExtractorDatabase connection, download file location, filename, trigger filename

Design Overview

This process extracts its brand data to Planning on a weekly basis.

Key assumptions for this integration:

- The full set of brands is included in this integration each time it runs.
- Retailers will not create a Diff with an ID of 'BRAND'.
- In order to meet the format required by Planning, the UDA description in this extract is hard coded to "Brand" and does not take into account the primary language configuration in Merchandising.
- The intended targets for this integration are
 - Assortment & Item Planning for Fashion/Softlines Cloud Service and Assortment & Item Planning Enterprise Edition Cloud Service (referred to jointly as APCS)

This process utilizes BDI (Bulk Data Integration) to facilitate the bulk data movement to Planning. The batch job BDI_RPAS_Brand_Fnd_PF_From_RMS_JOB is defined in the Merchandising JOS batch job admin as follows:

```
<job id="BDI_RPAS_Brand_Fnd_PF_From_RMS_JOB" version="1.0" xmlns="http://
xmlns.jcp.org/xml/ns/javaee">
  <properties>
    <property name="description" value="Extracts Brand information and writes it
out to a flat file for processing by AP and IP."/>
  </properties>
  <step id="batchlet-step">
    <batchlet ref="BDIInvokerBatchlet">
      <properties>
        <property name="bdiProcessFlowUrl" value="#SysOpt.bdiProcessFlowUrl"/>
        <property name="bdiProcessFlowCredential"
value="#SysOpt.bdiProcessFlowUrlUserAlias"/>
        <property name="predicateDS" value="RmsDBDS"/>
        <property name="predicateFunction"
value="RMS_BATCH_STATUS_SQL.GET_EOW_RUN_SIGNAL"/>
      </properties>
    </batchlet>
    <end on="COMPLETED"/>
  </step>
</job>
```

When the batch job BDI_RPAS_Brand_Fnd_PF_From_RMS_JOB is executed, a batchlet (BDIInvokerBatchlet) starts the execution flow. It calls a PLSQL function (RMS_BATCH_STATUS_SQL.GET_EOW_RUN_SIGNAL) to ensure the process flow is only executed on an end-of-week date. If the vdate is an end-of-week date, it invokes a BDI process flow (Brand_Fnd_ProcessFlow_From_RMS) to perform a series of steps to extract, download, and transport the downloaded files to target applications:

- Extractor job (Brand_Fnd_Extractor) calls BDI_FOUNDATION_SQL.BRAND_UP function to extract data from Merchandising table BRAND to BDI outbound staging table BRAND_OUT.
- Downloader file creator job calls the wrapper script, bdi_merch_extract_to_file_wrapper.sh, to set the runtime parameters on environment variables. This script will then call bdi_rpas_brand_extract.ksh to write brand information from the BRAND_OUT table into a comma-delimited flat file, which will be consumed by the target applications. A zero-byte trigger file is also generated to signal that the extract process was successful. Two separate copies of the data file and the trigger file are sent to the target applications.
- The downloaded data files and trigger files are written to designated locations as configured via BDI system options:
 - AP_outboundLocation
 - IP_outboundLocation

Scheduling Constraints

Schedule Information	Description
Processing Cycle	End of Day
Frequency	Scheduled daily but files will only be generated weekly on End of Week date.
Scheduling Considerations	N/A
Pre-Processing	N/A
Post-Processing	N/A
Threading Scheme	N/A

Restart/Recovery

N/A

Key Tables Affected

Table	Select	Insert	Update	Delete
BRAND	Yes	No	No	No
BRAND_OUT	Yes	Yes	No	Yes
BDI_DWNLDR_IFACE_MOD_DA TA_CTL	Yes	No	No	No
BDI_DWNLDR_IFACE_DATA_C TL	Yes	No	No	No

Integration Contract

The flat file will contain the following information:

Field Name	Field Type	Required	Description
UDA_ID	Char(6)	Yes	Hardcoded to 'BRAND'
UDA_DESC	Char(120)	Yes	Hardcoded to 'Brand'
BRAND_NAME	Char(30)	Yes	The brand ID from the Merchandising Brand table.
BRAND_DESCRIPTION	Char(120)	Yes	The brand description in the primary language from the Merchandising Brand table.

Calendar Extract to Planning and Forecasting (BDI_RPAS_Calendar_Fnd_PF_From_RMS_JOB)



Note:

This module replaces the ftmednld.pc module from previous releases.

Module Name	BDI_RPAS_Calendar_Fnd_PF_From_RMS_JOB
Description	Extracts calendar information to RPAS from RMS
Functional Area	Foundation
Module Type	Integration
Module Technology	BDI job
Catalog ID	N/A
Runtime Parameters	Calendar_Fnd_ProcessFlow_From_RMS Calendar_Fnd_Extractor

Design Overview

This program extracts calendar data to planning and forecasting on a weekly basis.

Key assumptions for this integration:

- The last two years, current year, and two years into the future are extracted each time this process is run.
- A data set is sent each time the extract runs.
- This extract supports a 4-5-4 calendar only.
- The intended targets for this integration are
 - Oracle Retail Merchandise Financial Planning Cloud Service (MFPCS)
 - Oracle Retail Demand Forecasting Cloud Service (RDFCS)
 - Assortment & Item Planning for Fashion/Softlines Cloud Service and Assortment & Item Planning Enterprise Edition Cloud Service (referred to jointly as APCS)

This program utilizes BDI (Bulk Data Integration) to facilitate the bulk data movement from Merchandising to the target applications.

The batch job `BDI_RPAS_Calendar_Fnd_PF_From_RMS_JOB` is defined in the Merchandising JOS batch job admin as follows:

```
<job id="BDI_RPAS_Calendar_Fnd_PF_From_RMS_JOB" version="1.0" xmlns="http://
xmlns.jcp.org/xml/ns/javaee">
  <properties>
    <property name="description" value="Extracts calendar information and writes
it out to a flat file for processing by both MFP and RDF."/>
  </properties>
  <step id="batchlet-step">
    <batchlet ref="BDIInvokerBatchlet">
      <properties>
        <property name="bdiProcessFlowUrl" value="#SysOpt.bdiProcessFlowUrl"/>
        <property name="bdiProcessFlowCredential"
value="#SysOpt.bdiProcessFlowUrlUserAlias"/>
        <property name="predicateDS" value="RmsDBDS"/>
        <property name="predicateFunction"
value="RMS_BATCH_STATUS_SQL.GET_EOW_RUN_SIGNAL"/>
      </properties>
    </batchlet>
    <end on="COMPLETED"/>
  </step>
</job>
```

When the batch job `BDI_RPAS_Calendar_Fnd_PF_From_RMS_JOB` is executed, a batchlet (`BDIInvokerBatchlet`) starts the execution flow. It calls a PLSQL function (`RMS_BATCH_STATUS_SQL.GET_EOW_RUN_SIGNAL`) to ensure the process flow is only executed on an end-of-week date. If the `vdate` is an end-of-week date, it invokes a BDI process flow (`Calendar_Fnd_ProcessFlow_From_RMS`) to perform a series of steps to extract, download, and transport the downloaded files to target applications:

- Extractor job (`Calendar_Fnd_Extractor`) calls `BDI_FOUNDATION_SQL.CALENDAR_UP` function to extract data from Merchandising view `V_BDI_DAY_LEVEL_CALENDAR` to BDI outbound staging table `CALENDAR_OUT`.
- A generic BDI Downloader file creator job writes calendar information from the `CALENDAR_OUT` table into a comma-delimited flat file, which will be consumed by the target applications. A zero-byte trigger file is also generated to signal that the extract process was successful. Separate copies of the data file and the trigger file are sent to the target applications.
- The downloaded data files and trigger files are written to designated locations as configured via BDI system options:
 - `MFP_outboundLocation`
 - `RDF_outboundLocation`
 - `AP_outboundLocation`
 - `IP_outboundLocation`

Scheduling Constraints

Schedule Information	Description
Processing Cycle	End of Day

Schedule Information	Description
Frequency	Scheduled daily but files will only be generated weekly on End of Week date.
Scheduling Considerations	N/A
Pre-Processing	N/A
Post-Processing	N/A
Threading Scheme	N/A

Restart/Recovery

N/A

Key Tables Affected

Table	Select	Insert	Update	Delete
V_BDI_DAY_LEVEL_CALENDAR	Yes	No	No	No
CALENDAR_OUT	Yes	Yes	No	Yes
BDI_DWNLDR_IFACE_MOD_DATA_CTL	Yes	No	No	No
BDI_DWNLDR_IFACE_DATA_CTL	Yes	No	No	No

Integration Contract

The flat file will contain the following information:

Field Name	Field Type	Required	Description
DAY	Date	Yes	The date for which the data was derived, in YYYYMMDD format
WEEK	Date	Yes	The end of week date for the day, in YYYYMMDD format
MONTH	Number(2)	Yes	The month number of the day in the year; valid values 1-12
QUARTER	Number(1)	Yes	The quarter of the year for the day; valid values 1-4
HALF	Number(1)	Yes	The half of the year for the day; valid values are 1 or 2
YEAR	Number(4)	Yes	The year for the day (YYYY format).
WEEK_OF_YEAR	Number(2)	Yes	The week of the year for the day; valid values 1-53
DAY_OF_WEEK	Number(1)	Yes	The day number within the week; valid values 1-7.

Currency Rates Extract to Planning and Forecasting (BDI_RPAS_CurrConvRates_Fnd_PF_From_RMS_JOB)

Module Name	BDI_RPAS_CurrConvRates_Fnd_PF_From_RMS_JOB bdi_merch_extract_to_file_wrapper.sh bdi_rpas_curr_conv_rates_extract.ksh
Description	Extracts currency rates information to RPAS
Functional Area	Foundation
Module Type	Integration
Module Technology	BDI job, shell scripts
Catalog ID	N/A
Runtime Parameters	CurrConvRates_Fnd_ProcessFlow_From_RMS CurrConvRates_Fnd_Extractor Database connection, download file location, filename, trigger filename

Design Overview

This program extracts its currency rates data to planning and forecasting on a weekly basis.

Key assumptions for this integration:

- Only currency rates for which stores and warehouse exist will be included in the extract.
- Either the consolidated or operational rate will be sent based on the setting of the Consolidation system option. If Y, then the consolidation rates will be sent. If N, then the operational rates are used.
- All applicable currency rates are sent each time this process is run.
- The rates sent in this integration are based on a materialized view. The process that refreshes this view (batch_rfmvcurrconv.ksh) must be scheduled to ensure that the latest currency information is sent each week.
- The intended targets for this integration are
 - Oracle Retail Merchandise Financial Planning Cloud Service (MFPCS)
 - Oracle Retail Demand Forecasting Cloud Service (RDFCS)
 - Assortment & Item Planning for Fashion/Soflines Cloud Service and Assortment & Item Planning Enterprise Edition Cloud Service (referred to jointly as APCS)

This program utilizes BDI (Bulk Data Integration) to facilitate the bulk data movement from Merchandising to the target applications.

The batch job BDI_RPAS_CurrConvRates_Fnd_PF_From_RMS_JOB is defined in the Merchandising JOS batch job admin as follows:

```
<job id="BDI_RPAS_CurrConvRates_Fnd_PF_From_RMS_JOB" version="1.0" xmlns="http://
xmlns.jcp.org/xml/ns/javaee">
  <properties>
    <property name="description" value="Extracts currency conversion rate
information and writes it out to a flat file for processing by both MFP and RDF."/>
  </properties>
  <step id="batchlet-step">
    <batchlet ref="BDIInvokerBatchlet">
```

```

        <properties>
          <property name="bdiProcessFlowUrl"
value="#SysOpt.bdiProcessFlowUrl"/>
          <property name="bdiProcessFlowCredential"
value="#SysOpt.bdiProcessFlowUrlUserAlias"/>
          <property name="predicateDS" value="RmsDBDS"/>
          <property name="predicateFunction"
value="RMS_BATCH_STATUS_SQL.GET_EOW_RUN_SIGNAL"/>
        </properties>
      </batchlet>
    <end on="COMPLETED"/>
  </step>
</job>

```

When the batch job BDI_RPAS_CurrConvRates_Fnd_PF_From_RMS_JOB is executed, a batchlet (BDIInvokerBatchlet) starts the execution flow. It calls a PLSQL function (RMS_BATCH_STATUS_SQL.GET_EOW_RUN_SIGNAL) to ensure the process flow is only executed on an end-of-week date. If the vdate is an end-of-week date, it invokes a BDI process flow (CurrConvRates_Fnd_ProcessFlow_From_RMS) to perform a series of steps to extract, download, and transport the downloaded files to target applications:

- Extractor job (CurrConvRates_Fnd_Extractor) calls BDI_FOUNDATION_SQL.CURR_CONV_RATES_UP function to extract data from Merchandising view MV_CURRENCY_CONVERSION_RATES to BDI outbound staging table CURR_CONV_RATES_OUT.
 - Only the currencies for which stores and warehouses exist in Merchandising will be extracted.
 - Either consolidated or operational rates will be included based on Merchandising system options (consolidation_ind).
- Downloader file creator job calls the wrapper script, bdi_merch_extract_to_file_wrapper.sh, to set the runtime parameters on environment variables. This script will then call bdi_rpas_curr_conv_rates_extract.ksh to write currency rates information from the CURR_CONV_RATES_OUT table into a comma-delimited flat file, which will be consumed by the target applications. A zero-byte trigger file is also generated to signal that the extract process was successful. Separate copies of the data file and the trigger file are sent to the target applications.
- The downloaded data files and trigger files are written to designated locations as configured via BDI system options:
 - MFP_outboundLocation
 - RDF_outboundLocation
 - AP_outboundLocation
 - IP_outboundLocation

Scheduling Constraints

Schedule Information	Description
Processing Cycle	End of Day
Frequency	Scheduled daily but files will only be generated weekly on End of Week date

Schedule Information	Description
Scheduling Considerations	N/A
Pre-Processing	batch_rfmvcrrconv.ksh
Post-Processing	N/A
Threading Scheme	N/A

Restart/Recovery

N/A

Key Tables Affected

Table	Select	Insert	Update	Delete
MV_CURRENCY_CONVERSI N_RATES	Yes	No	No	No
SYSTEM_OPTIONS	Yes	No	No	No
CURR_CONV_RATES_OUT	Yes	Yes	No	Yes
BDI_DWNLDR_IFACE_MOD_DA TA_CTL	Yes	No	No	No
BDI_DWNLDR_IFACE_DATA_C TL	Yes	No	No	No

Integration Contract

The flat file will contain the following information:

Field Name	Field Type	Required	Description
EFFECTIVE_DATE	Date	Yes	Holds the effective date of the exchange rate for the currencies and the exchange type
FROM_CURRENCY_CODE	Char(3)	Yes	Holds the convert from currency code.
TO_CURRENCY_CODE	Char(3)	Yes	Holds the convert to currency code.
EXCHANGE_TYPE	Char(1)	Yes	Identifies the type of exchange rate. This will be either C (consolidation) or O (operational).
EXCHANGE_RATE	Number(20,10)	Yes	Contains the exchange rate between the from and to currencies for the specified exchange type on the next effective date. It is expressed in terms of the to-currency.

Differentiator Extract to Planning (BDI_RPAS_Diff_Fnd_PF_From_RMS_JOB)

Module Name	BDI_RPAS_Diff_Fnd_PF_From_RMS_JOBbdi_merch_extract_to_file_wrapper.shbdi_rpas_diff_extract.ksh
Description	Extracts Diff Types and Diff ID information to Planning
Functional Area	Foundation
Module Type	Integration
Module Technology	BDI job, shell scripts
Catalog ID	N/A
Runtime Parameters	Diff_Fnd_ProcessFlow_From_RMS Diff_Fnd_Extractor Database connection, download file location, filename, trigger filename

Design Overview

This process extracts its differentiator data to Planning on a weekly basis.

Key assumptions for this integration:

- The full set of differentiators and diff types are included in this integration each time it runs.
- The intended targets for this integration are
 - Assortment & Item Planning for Fashion/Softlines Cloud Service and Assortment & Item Planning Enterprise Edition Cloud Service (referred to jointly as APCS)

This process utilizes BDI (Bulk Data Integration) to facilitate the bulk data movement to Planning. The batch job BDI_RPAS_Diff_Fnd_PF_From_RMS_JOB is defined in the Merchandising JOS batch job admin as follows:

```
<job id="BDI_RPAS_Diff_Fnd_PF_From_RMS_JOB" version="1.0" xmlns="http://
xmlns.jcp.org/xml/ns/javaee">
  <properties>
    <property name="description" value="Extracts Diff Types and Diff ID
information and writes it out to a flat file for processing by AP and IP."/>
  </properties>
  <step id="batchlet-step">
    <batchlet ref="BDIInvokerBatchlet">
      <properties>
        <property name="bdiProcessFlowUrl" value="#SysOpt.bdiProcessFlowUrl"/>
        <property name="bdiProcessFlowCredential"
value="#SysOpt.bdiProcessFlowUrlUserAlias"/>
        <property name="predicateDS" value="RmsDBDS"/>
        <property name="predicateFunction"
value="RMS_BATCH_STATUS_SQL.GET_EOW_RUN_SIGNAL"/>
      </properties>
    </batchlet>
    <end on="COMPLETED"/>
  </step>
</job>
```

When the batch job BDI_RPAS_Diff_Fnd_PF_From_RMS_JOB is executed, a batchlet (BDIInvokerBatchlet) starts the execution flow. It calls a PLSQL function

(RMS_BATCH_STATUS_SQL.GET_EOW_RUN_SIGNAL) to ensure the process flow is only executed on an end-of-week date. If the vdate is an end-of-week date, it invokes a BDI process flow (Diff_Fnd_ProcessFlow_From_RMS) to perform a series of steps to extract, download, and transport the downloaded files to target applications:

- Extractor job (Diff_Fnd_Extractor) calls BDI_CROSS_PILLAR_SQL.DIFF_UP function to extract data from DIFF_IDS and DIFF_TYPE to BDI outbound staging table DIFF_OUT.
- Downloader file creator job calls the wrapper script, bdi_merch_extract_to_file_wrapper.sh, to set the runtime parameters on environment variables. This script will then call bdi_rpas_diff_extract.ksh to write differentiator information from the DIFF_OUT table into a comma-delimited flat file, which will be consumed by the target applications. A zero-byte trigger file is also generated to signal that the extract process was successful. Separate copies of the data file and the trigger file are sent to the target applications.
- The downloaded data files and trigger files are written to designated locations as configured via BDI system options:
 - AP_outboundLocation
 - IP_outboundLocation

Scheduling Constraints

Schedule Information	Description
Processing Cycle	End of Day
Frequency	Scheduled daily but files will only be generated weekly on End of Week date.
Scheduling Considerations	N/A
Pre-Processing	N/A
Post-Processing	N/A
Threading Scheme	N/A

Restart/Recovery

N/A

Key Tables Affected

Table	Select	Insert	Update	Delete
DIFF_IDS	Yes	No	No	No
DIFF_TYPE	Yes	No	No	No
DIFF_OUT	Yes	Yes	No	Yes
BDI_DWNLDR_IFACE_MOD_DA TA_CTL	Yes	No	No	No
BDI_DWNLDR_IFACE_DATA_C TL	Yes	No	No	No

Integration Contract

The flat file will contain the following information:

Field Name	Field Type	Required	Description
DIFF_TYPE_ID	Char(6)	Yes	The ID of the diff type (for example, C for color).
DIFF_TYPE_DESC	Char(120)	Yes	The description of the diff type (for example, Color) in the primary language.
DIFF_ID	Char(10)	Yes	The ID of the diff (for example, S for Small).
DIFF_DESC	Char(120)	Yes	The description of the diff (for example, Small) in the primary language.

Inventory Extract to Planning (BDI_MFP_Inventory_Tx_PF_From_RMS_JOB)

Module Name	BDI_MFP_Inventory_Tx_PF_From_RMS_JOB
Description	Extracts inventory information to Planning
Functional Area	Inventory
Module Type	Integration
Module Technology	BDI job
Catalog ID	N/A
Runtime Parameters	Inventory_Tx_ProcessFlow_From_RMS Inventory_Tx_Extractor

Design Overview

This process extracts owned inventory information for inventoried, non-pack approved transaction items to planning on a weekly basis, at the end of the week. The integration captures the current on-hand and in-transit for all the included item/locations at the point in time that the integration is run.

Key assumptions for this integration:

- Only inventoried, approved transaction items are included in the integration.
- Any inventory for pack items is aggregated with inventory for the component items.
- Only stockholding stores are included in the integration.
- Cost values are based on system configuration for cost:
 - For a cost department with the system configured for average cost, the cost basis is the item/location's weighted average cost, converted to primary currency.
 - For a cost department with the system configured for standard cost, the cost basis is the item/locations unit cost, converted to primary currency.
 - For a retail department, the cumulative mark-on percentage is used to calculate cost based on the retail price, converted to primary currency.
- Retail values sent are based on the current item/location retail price, converted to primary currency. The retail will include VAT if the system option to include VAT in

the stock ledger is set to include VAT so that the retail values in this integration are consistent with other data sent to planning.

- All unit values are sent in terms of the standard unit of measure for the item.
- Planning will interpret inventory as being clearance if the clearance flag sent in this integration shows the item/location to be on clearance at the end of the week.
- The intended targets for this integration are
 - Oracle Retail Merchandise Financial Planning Cloud Service (MFPCS)
 - Assortment & Item Planning for Fashion/Softlines Cloud Service and Assortment & Item Planning Enterprise Edition Cloud Service (referred to jointly as APCS)

This process utilizes BDI (Bulk Data Integration) to facilitate the bulk data movement from Merchandising to the target applications. The batch job `BDI_MFP_Inventory_Tx_PF_From_RMS_JOB` is defined in the Merchandising JOS batch job admin as follows:

```
<job id="BDI_MFP_Inventory_Tx_PF_From_RMS_JOB" version="1.0" xmlns="http://
xmlns.jcp.org/xml/ns/javaee">
  <properties>
    <property name="description" value="Extracts information regarding inventory
for use by the MFP application"/>
  </properties>
  <step id="batchlet-step">
    <batchlet ref="BDIInvokerBatchlet">
      <properties>
        <property name="bdiProcessFlowUrl" value="#SysOpt.bdiProcessFlowUrl"/>
        <property name="bdiProcessFlowCredential"
value="#SysOpt.bdiProcessFlowUrlUserAlias"/>
        <property name="predicateDS" value="RmsDBDS"/>
        <property name="predicateFunction"
value="RMS_BATCH_STATUS_SQL.GET_EOW_RUN_SIGNAL"/>
      </properties>
    </batchlet>
    <end on="COMPLETED"/>
  </step>
</job>
```

When the batch job `BDI_MFP_Inventory_Tx_PF_From_RMS_JOB` is executed, a batchlet (`BDIInvokerBatchlet`) starts the execution flow. It calls a PLSQL function (`RMS_BATCH_STATUS_SQL.GET_EOW_RUN_SIGNAL`) to ensure the process flow is only executed on an end-of-week date. If the `vdate` is an end-of-week date, it invokes a BDI process flow (`Inventory_Tx_ProcessFlow_From_RMS`) to perform a series of steps to extract, download, and transport the downloaded files to target applications:

- Extractor job (`Inventory_Tx_ExtractorJob`) calls `BDI_MFP_SQL.INVENTORY_UP` function to extract data from Merchandising view `V_BDI_MFP_INVENTORY` to BDI outbound staging table `INVENTORY_OUT`.
- A generic BDI Downloader file creator job writes inventory quantities information from the `INVENTORY_OUT` table into a comma-delimited flat file, which will be consumed by the target applications. A zero-byte trigger file is also generated to signal that the extract process was successful. Two separate copies of the data file and the trigger file are sent to the target applications.
- The downloaded data files and trigger files are written to designated locations as configured via BDI system options:
 - `MFP_outboundLocation`

- AP_outboundLocation
- IP_outboundLocation

Scheduling Constraints

Schedule Information	Description
Processing Cycle	End of Day
Frequency	Scheduled daily but files will only be generated weekly on End of Week date
Scheduling Considerations	N/A
Pre-Processing	N/A
Post-Processing	N/A
Threading Scheme	N/A

Restart/Recovery

N/A

Key Tables Affected

Table	Select	Insert	Update	Delete
V_BDI_MFP_INVENTORY	Yes	No	No	No
INVENTORY_OUT	Yes	Yes	No	Yes
BDI_DWNLDR_IFACE_MOD_DATA_CTL	Yes	No	No	No
BDI_DWNLDR_IFACE_DATA_CTL	Yes	No	No	No

Integration Contract

The flat file will contain the following information:

Field Name	Field Type	Required	Description
EOW	Date	Yes	Indicates the end of week date that the on order information pertains to.
ITEM	Varchar2(25)	Yes	Transaction level item only.
LOCATION	Number(10)	Yes	Could be a store or virtual warehouse.
LOC_TYPE	Varchar2(1)	Yes	Indicates if the location is a store or warehouse - S = Store; W = Warehouse.
CLEAR_IND	Number(1)	Yes	Indicates if the item/location is currently on clearance.

Field Name	Field Type	Required	Description
REGULAR_INVENTOR Y_UNITS	Number(12,4)	Yes	Current owned inventory for the item/location in units based on the standard unit of measure; calculated as stock on hand + pack component stock on hand + in transit + pack component in transit.
REGULAR_INVENTOR Y_COST	Number(20,4)	Yes	The cost value of current owned inventory for the item/location; calculated based on unit inventory and the cost basis of the item's department, as described above.
REGULAR_INVENTOR Y_RETAIL	Number(20,4)	Yes	The retail value of current owned inventory for the item/location; calculated based on the unit inventory value shown above and the current item/location unit retail.
UNIT_COST	Number(20,4)	Yes	The current supplier purchase cost for the item/location.
AV_COST	Number(20,4)	Yes	The current weighted average cost for the item/location.
UNIT_RETAIL	Number(20,4)	Yes	The current unit retail for the item/location. If the item is on clearance, this would be the clearance price.

Merchandise Hierarchy and Item Extract to Planning and Forecasting (BDI_RPAS_MerchHier_Fnd_PF_From_RMS_JOB)

Module Name	BDI_RPAS_MerchHier_Fnd_PF_From_RMS_JOB bdi_merch_extract_to_file_wrapper.sh bdi_rpas_merchhier_extract.ksh
Description	Extracts merchandise hierarchy and item information to RPAS
Functional Area	Foundation
Module Type	Integration
Module Technology	BDI job, shell scripts
Catalog ID	N/A
Runtime Parameters	ItemHdrAndMerchHier_Fnd_ProcessFlow_From_RMS ItemHdr_Fnd_Extractor Database connection, download file location, filename, trigger filename

Design Overview

This program extracts the merchandise hierarchy from company to transaction level item to planning and forecasting on a weekly basis. Additional key attributes about the items are also

included, such as the primary supplier, brand, and any differentiators (for example, colors, sizes, and so on) that exist for the item.

Key assumptions for this integration:

- The full merchandise hierarchy and all items are sent each time this process is run.
- Only approved, inventoried and sellable transaction-level items will be included in the integration. Pack items are not included.
- All descriptions are sent in the primary language as defined in Merchandising.
- For transaction items that do not have a parent item, then the transaction item is also displayed as the parent item, as well as the parent/diff level.
- For a parent item that is not marked as an aggregate item or does not have any of its diffs flagged as aggregates, the parent item is sent as the parent/diff level for all of its transaction items.
- A single unit of measure is assumed for all items and therefore the standard units of measure for the items are not sent.
- The intended targets for this integration are
 - Oracle Retail Merchandise Financial Planning Cloud Service (MFPCS)
 - Oracle Retail Demand Forecasting Cloud Service (RDFCS)
 - Assortment & Item Planning for Fashion/Softlines Cloud Service and Assortment & Item Planning Enterprise Edition Cloud Service (referred to jointly as APCS)

This program utilizes BDI (Bulk Data Integration) to facilitate the bulk data movement to the target applications. The batch job `BDI_RPAS_MerchHier_Fnd_PF_From_RMS_JOB` is defined in the Merchandising JOS batch job admin as follows:

```
<job id="BDI_RPAS_MerchHier_Fnd_PF_From_RMS_JOB" version="1.0" xmlns="http://
xmlns.jcp.org/xml/ns/javaee">
  <properties>
    <property name="description" value="Extracts Merch Hierarchy information
and writes it out to a flat file for processing by both MFP and RDF."/>
  </properties>
  <step id="batchlet-step">
    <batchlet ref="BDIInvokerBatchlet">
      <properties>
        <property name="bdiProcessFlowUrl"
value="#SysOpt.bdiProcessFlowUrl"/>
        <property name="bdiProcessFlowCredential"
value="#SysOpt.bdiProcessFlowUrlUserAlias"/>
        <property name="predicateDS" value="RmsDBDS"/>
        <property name="predicateFunction"
value="RMS_BATCH_STATUS_SQL.GET_EOW_RUN_SIGNAL"/>
      </properties>
    </batchlet>
    <end on="COMPLETED"/>
  </step>
</job>
```

When the batch job `BDI_RPAS_MerchHier_Fnd_PF_From_RMS_JOB` is executed, a batchlet (`BDIInvokerBatchlet`) starts the execution flow. It calls a PLSQL function (`RMS_BATCH_STATUS_SQL.GET_EOW_RUN_SIGNAL`) to ensure the process flow

is only executed on an end-of-week date. If the vdate is an end-of-week date, it invokes a BDI process flow (ItemHdrAndMerchHier_Fnd_ProcessFlow_From_RMS) to perform a series of steps to extract, download, and transport the downloaded files to the target applications:

- Extractor jobs (MerchHier_Fnd_Extractor, ItemHdr_Fnd_Extractor) call respective BDI_MERCH_SQL and BDI_ITEM_SQL functions to extract data from Merchandising tables to BDI outbound staging tables MERCH_HIER_OUT and ITEM_HDR_OUT.
- Downloader file creator job calls the wrapper script, bdi_merch_extract_to_file_wrapper.sh, to set the runtime parameters on environment variables. This script will then call bdi_rpas_merchhier_extract.ksh to write merchandise hierarchy and item information from the MERCH_HIER_OUT and ITEM_HDR_OUT tables into a comma-delimited flat file, which will be consumed by the target applications. A zero-byte trigger file is also generated to signal that the extract process was successful. Separate copies of the data file and the trigger file are sent to the target applications.
- The downloaded data files and trigger files are written to designated locations as configured via BDI system options:
 - MFP_outboundLocation
 - RDF_outboundLocation
 - AP_outboundLocation
 - IP_outboundLocation

Scheduling Constraints

Schedule Information	Description
Processing Cycle	End of Day
Frequency	Scheduled daily but files will only be generated weekly on End of Week date.
Scheduling Considerations	N/A
Pre-Processing	N/A
Post-Processing	N/A
Threading Scheme	N/A

Restart/Recovery

N/A

Key Tables Affected

Table	Select	Insert	Update	Delete
COMPHEAD	Yes	No	No	No
DIVISION	Yes	No	No	No
GROUPS	Yes	No	No	No
DEPS	Yes	No	No	No
CLASS	Yes	No	No	No
SUBCLASS	Yes	No	No	No

Table	Select	Insert	Update	Delete
ITEM_MASTER	Yes	No	No	No
DIFF_GROUP_HEAD	Yes	No	No	No
DIFF_IDS	Yes	No	No	No
SYSTEM_OPTIONS	Yes	No	No	No
MERCH_HIER_OUT	Yes	Yes	No	Yes
ITEM_HDR_OUT	Yes	Yes	No	Yes
BDI_DWNLDR_IFACE_MOD_DATA_CTL	Yes	No	No	No
BDI_DWNLDR_IFACE_DATA_CTL	Yes	No	No	No
ITEM_SUPPLIER_OUT	Yes	Yes	No	Yes

Integration Contract

The flat file will contain the following information:

Field Name	Field Type	Required	Description
ITEM	Char(25)	Yes	The transaction level item ID.
ITEM_DESC	Char(250)	Yes	The transaction level item description.
ITEM_PARENT_DIFF	Char(30)	Yes	Concatenated value consisting of item parent ID with the composite diff aggregate. If there is no item parent, this will contain the transaction level item.
ITEM_PARENT_DIFF_DESC	Char(250)	Yes	Description of the item parent diff. Concatenated value consisting of the item parent description and the diff IDs for all diffs associated to the parent marked as aggregates. If there is no item parent, it will contain the transaction level item description.
ITEM_PARENT	Char(25)	Yes	If there is no item parent, it will contain the transaction level item.
ITEM_PARENT_DESC	Char(250)	Yes	If there is no item parent, it will contain the transaction level item description.
SUBCLASS_ID	Number(10)	Yes	Unique subclass ID
SUBCLASS_NAME	Char(120)	Yes	Concatenated value consisting of the subclass number with name.
CLASS_ID	Number(10)	Yes	Unique class ID

Field Name	Field Type	Required	Description
CLASS_NAME	Char(120)	Yes	Concatenated value consisting of the class number with name.
DEPT	Number(4)	Yes	Department ID
DEPT_NAME	Char(120)	Yes	Concatenated value consisting of the department ID and name.
GROUP_NO	Number(4)	Yes	Group ID
GROUP_NAME	Char(120)	Yes	Group name
DIVISION	Number(4)	Yes	Division ID
DIV_NAME	Char(120)	Yes	Division name
COMPANY	Number(4)	Yes	Company ID
COMPANY_NAME	Char(120)	Yes	Company name
FORECAST_IND	Char(1)	Yes	Indicates whether or not the item should be forecasted. Valid values are Y or N.
CLASS	Number(10)	Yes	The class ID that is displayed in the Merchandising screens.
SUBCLASS	Number(10)	Yes	The subclass ID that is displayed in the Merchandising screens.
BRAND_NAME	Char(30)	Yes	If a brand is not assigned, this is defaulted to 'NA'.
BRAND_DESCRIPTION	Char(120)	Yes	The brand description for the transaction item. If a brand is not assigned, this is defaulted to 'Not Assigned'.
SUPPLIER	Number(10)	Yes	The ID of the primary supplier for the transaction item.
SUPPLIER_NAME	Char(240)	Yes	The name of the primary supplier for the transaction item.
DIFF_1	Char(10)	No	The ID of the first diff for the transaction level item. If a diff is not assigned, this is defaulted to 'NA'.
DIFF_1_DESC	Char(120)	No	The name of the first diff for the transaction item. If a diff is not assigned, this is defaulted to 'unassigned'.
DIFF_2	Char(10)	No	The ID of the second diff for the transaction item. If a diff is not assigned, this is defaulted to 'NA'.
DIFF_2_DESC	Char(120)	No	The name of the second diff for the transaction item. If a diff is not assigned, this is defaulted to 'unassigned'.

Field Name	Field Type	Required	Description
DIFF_3	Char(10)	No	The ID of the third diff for the transaction item. If a diff is not assigned, this is defaulted to 'NA'.
DIFF_3_DESC	Char(120)	No	The name of the third diff for the transaction item. If a diff is not assigned, this is defaulted to 'unassigned'.
DIFF_4	Char(10)	No	The ID of the fourth diff for the transaction item. If a diff is not assigned, this is defaulted to 'NA'.
DIFF_4_DESC	Char(120)	No	The name of the fourth diff for the transaction item. If a diff is not assigned, this is defaulted to 'unassigned'.

On Order Extract to Planning (BDI_MFP_OnOrder_Tx_PF_From_RMS_JOB)



Note:

This module replaces the onordext.pc and onordddld.pc modules from previous releases.

Module Name	BDI_MFP_OnOrder_Tx_PF_From_RMS_JOB
Description	Extracts inventory information to Planning
Functional Area	Inventory Tracking
Module Type	Integration
Module Technology	BDI job
Catalog ID	N/A
Runtime Parameters	OnOrder_Tx_ProcessFlow_From_RMS OnOrder_Tx_Extractor

Design Overview

This process extracts its quantities on order to planning and forecasting on a weekly basis, at the end of the week. The integration sends any open on order quantities aggregated by week, grouped by the open to buy end of week date. Any on order quantity that is still open and has an OTB EOW date in the past will be combined with the current week's on order.

Key assumptions for this integration:

- Only orderable, inventoried, approved transaction items are included in the integration.
- Any on order for pack items is sent based on the component items.

- Purchase orders flagged to not be included in "on order" are not included in the integration.
- Cost and retail values sent are based on the purchase order's cost and retail value, converted to primary currency.
- Retail values will include VAT if the system option to include VAT in the stock ledger is set to include VAT so that the retail values in this integration are consistent with other data sent to planning.
- All unit values are sent in terms of the standard unit of measure for the item.
- Planning will interpret the on order as being clearance if the clearance flag sent in this integration shows the item/location to be on clearance at the end of the week.
- The intended targets for this integration are
 - Oracle Retail Merchandise Financial Planning Cloud Service (MFPCS)
 - Assortment & Item Planning for Fashion/Softlines Cloud Service and Assortment & Item Planning Enterprise Edition Cloud Service (referred to jointly as APCS)

This process utilizes BDI (Bulk Data Integration) to facilitate the bulk data movement to the target applications.

The batch job `BDI_MFP_OnOrder_Tx_PF_From_RMS_JOB` is defined in the Merchandising JOS batch job admin as follows:

```
<job id="BDI_MFP_OnOrder_Tx_PF_From_RMS_JOB" version="1.0" xmlns="http://
xmlns.jcp.org/xml/ns/javaee">
  <properties>
    <property name="description" value="Extracts information regarding quantities
on order for use by the MFP application"/>
  </properties>
  <step id="batchlet-step">
    <batchlet ref="BDIInvokerBatchlet">
      <properties>
        <property name="bdiProcessFlowUrl" value="#SysOpt.bdiProcessFlowUrl"/>
        <property name="bdiProcessFlowCredential"
value="#SysOpt.bdiProcessFlowUrlUserAlias"/>
        <property name="predicateDS" value="RmsDBDS"/>
        <property name="predicateFunction"
value="RMS_BATCH_STATUS_SQL.GET_EOW_RUN_SIGNAL"/>
      </properties>
    </batchlet>
    <end on="COMPLETED"/>
  </step>
</job>
```

When the batch job `BDI_MFP_OnOrder_Tx_PF_From_RMS_JOB` is executed, a batchlet (`BDIInvokerBatchlet`) starts the execution flow. It calls a PLSQL function (`RMS_BATCH_STATUS_SQL.GET_EOW_RUN_SIGNAL`) to ensure the process flow is only executed on an end of week date. If the `vdate` is an end of week date, it invokes a BDI process flow (`OnOrder_Tx_ProcessFlow_RMS`) to perform a series of steps to extract, download, and transport the downloaded files to target applications:

- Extractor job (`OnOrder_Tx_Extractor`) calls `BDI_MFP_SQL.ON_ORDER_UP` function to extract data from Merchandising view `V_BDI_MFP_ON_ORDER` to BDI outbound staging table `ON_ORDER_OUT`.
- A generic BDI Downloader file creator job writes quantities on order information from the `ON_ORDER_OUT` table into a comma-delimited flat file, which will be consumed by the target applications. A zero-byte trigger file is also generated to signal that the extract

process was successful. Separate copies of the data file and the trigger file are sent to the target applications.

- The downloaded data files and trigger files are written to designated locations as configured via BDI system options:
 - MFP_outboundLocation
 - AP_outboundLocation
 - IP_outboundLocation

Scheduling Constraints

Schedule Information	Description
Processing Cycle	End of Day
Frequency	Scheduled daily but files will only be generated weekly on End of Week date.
Scheduling Considerations	N/A
Pre-Processing	N/A
Post-Processing	N/A
Threading Scheme	N/A

Restart/Recovery

N/A

Key Tables Affected

Table	Select	Insert	Update	Delete
V_BDI_MFP_ON_ORDER	Yes	No	No	No
ON_ORDER_OUT	Yes	Yes	No	Yes
BDI_DWNLDR_IFACE_MOD_DATA_CTL	Yes	No	No	No
BDI_DWNLDR_IFACE_DATA_CTL	Yes	No	No	No

Integration Contract

The flat file will contain the following information:

Field Name	Field Type	Required	Description
EOW	Date	Yes	Indicates the end of week date that the on order information pertains to.
ITEM	Varchar2(25)	Yes	Transaction level item only.
LOCATION	Number(10)	Yes	Could be a store or virtual warehouse.

Field Name	Field Type	Required	Description
LOC_TYPE	Varchar2(1)	Yes	Indicates if the location is a store or warehouse - S = Store; W = Warehouse.
CLEAR_IND	Number(1)	Yes	Indicates if the item/location is currently on clearance.
ON_ORDER_UNITS	Number(12)	Yes	Indicates the total quantity of the item in the order in standard unit of measure.
ON_ORDER_COST	Number(20,4)	Yes	on order * PO cost in primary currency
ON_ORDER_RETAIL	Number(20,4)	Yes	on order * PO retail in primary currency

Organization Hierarchy Extract to Planning and Forecasting (BDI_RPAS_OrgHier_Fnd_PF_From_RMS_JOB)

Module Name	BDI_RPAS_OrgHier_Fnd_PF_From_RMS_JOB bdi_merch_extract_to_file_wrapper.sh bdi_rpas_orghier_extract.ksh
Description	Extracts organizational hierarchy information to RPAS
Functional Area	Foundation
Module Type	Integration
Module Technology	BDI job, shell scripts
Catalog ID	N/A
Runtime Parameters	StoreAndWhAndOrgHier_Fnd_ProcessFlow_From_RMS Store_Fnd_Extractor Wh_Fnd_Extractor OrgHier_Fnd_Extractor Database connection, download file location, filename, trigger filename

Design Overview

This program extracts the organization hierarchy data from company to location, which can be stores or warehouses to planning and forecasting on a weekly basis. Additional key attributes about the organizational hierarchy will also be sent to assist in building alternate hierarchies for planning, such as channel.

Key assumptions for this integration:

- MFPCS will use the third level of the Merchandising hierarchy (area) to represent channel.
- The full organizational hierarchy is sent each time this process is run.
- All names and descriptions are sent in the primary language only.
- The location in the file can represent either a store or a virtual warehouse location.

- Because warehouses live outside the organization hierarchy, for the levels of the organizational hierarchy above location (chain through district) when the location is a warehouse, the warehouse ID and description will be repeated.
- The intended targets for this integration are
 - Oracle Retail Merchandise Financial Planning Cloud Service (MFPCS)
 - Oracle Retail Demand Forecasting Cloud Service (RDFCS)
 - Assortment & Item Planning for Fashion/Softlines Cloud Service and Assortment & Item Planning Enterprise Edition Cloud Service (referred to jointly as APCS)

This program utilizes BDI (Bulk Data Integration) to facilitate the bulk data movement to the target applications. The batch job `BDI_RPAS_OrgHier_Fnd_PF_From_RMS_JOB` is defined in the Merchandising JOS batch job admin as follows:

```
<job id="BDI_RPAS_OrgHier_Fnd_PF_From_RMS_JOB" version="1.0" xmlns="http://
xmlns.jcp.org/xml/ns/javaee">
  <properties>
    <property name="description" value="Extracts Org Hierarchy information
and writes it out to a flat file for processing by both MFP and RDF."/>
  </properties>
  <step id="batchlet-step">
    <batchlet ref="BDIInvokerBatchlet">
      <properties>
        <property name="bdiProcessFlowUrl"
value="#SysOpt.bdiProcessFlowUrl"/>
        <property name="bdiProcessFlowCredential"
value="#SysOpt.bdiProcessFlowUrlUserAlias"/>
        <property name="predicateDS" value="RmsDBDS"/>
        <property name="predicateFunction"
value="RMS_BATCH_STATUS_SQL.GET_EOW_RUN_SIGNAL"/>
      </properties>
    </batchlet>
    <end on="COMPLETED"/>
  </step>
</job>
```

When the batch job `BDI_RPAS_OrgHier_Fnd_PF_From_RMS_JOB` is executed, a batchlet (`BDIInvokerBatchlet`) starts the execution flow. It calls a PLSQL function (`RMS_BATCH_STATUS_SQL.GET_EOW_RUN_SIGNAL`) to ensure the process flow is only executed on an end-of-week date. If the `vdate` is an end-of-week date, it invokes a BDI process flow (`StoreAndWhAndOrgHier_Fnd_ProcessFlow_From_RMS`) to perform a series of steps to extract, download, and transport the downloaded files to target applications:

- Extractor jobs (`Store_Fnd_Extractor`, `Wh_Fnd_Extractor`, `OrgHier_Fnd_Extractor`) call respective `BDI_ORG_SQL` functions to extract data from Merchandising tables to BDI outbound staging tables `ORG_HIER_OUT`, `STORE_OUT`, and `WH_OUT`.
- Downloader file creator job calls the wrapper script, `bdi_merch_extract_to_file_wrapper.sh`, to set the runtime parameters on environment variables. This script will then call `bdi_rpas_orghier_extract.ksh` to write organization hierarchy information from the `ORG_HIER_OUT`, `STORE_OUT`, and `WH_OUT` tables into a comma-delimited flat file, which will be consumed by the target applications. A zero-byte trigger file is also generated to signal that the extract process was successful. Separate copies of the data file and the trigger file are sent to the target applications.

- The downloaded data files and trigger files are written to designated locations as configured via BDI system options:
 - MFP_outboundLocation
 - RDF_outboundLocation
 - AP_outboundLocation
 - IP_outboundLocation

Scheduling Constraints

Schedule Information	Description
Processing Cycle	End of Day
Frequency	Scheduled daily but files will only be generated weekly on End of Week date.
Scheduling Considerations	N/A
Pre-Processing	N/A
Post-Processing	N/A
Threading Scheme	N/A

Restart/Recovery

N/A

Key Tables Affected

Table	Select	Insert	Update	Delete
STORE	Yes	No	No	No
WH	Yes	No	No	No
AREA	Yes	No	No	No
CHAIN	Yes	No	No	No
DISTRICT	Yes	No	No	No
REGION	Yes	No	No	No
COMPHEAD	Yes	No	No	No
CHANNELS	Yes	No	No	No
CODE_DETAIL	Yes	No	No	No
STORE_FORMAT	Yes	No	No	No
LANG	Yes	No	No	No
VAT_REGION	Yes	No	No	No
TSFZONE	Yes	No	No	No
ORG_HIER_OUT	Yes	Yes	No	Yes
STORE_OUT	Yes	Yes	No	Yes
WH_OUT	Yes	Yes	No	Yes
BDI_DWNLDR_IFACE_MOD_DA TA_CTL	Yes	No	No	No

Table	Select	Insert	Update	Delete
BDI_DWNLDR_IFACE_DATA_C TL	Yes	No	No	No

Integration Contract

The flat file will contain the following information:

Field Name	Field Type	Required	Description
LOCATION	Number(10)	Yes	Store or virtual warehouse ID
LOC_NAME	Char(150)	Yes	Store or warehouse name
DISTRICT	Number(10)	Yes	District ID; for warehouses, repeat the warehouse ID with the prefix "WH"
DISTRICT_NAME	Char(120)	Yes	District name; for warehouses, repeat the warehouse name
REGION	Number(10)	Yes	Region ID; for warehouses, repeat the warehouse ID with the prefix "WH"
REGION_NAME	Char(120)	Yes	Region name; for warehouses, repeat the warehouse name
AREA	Number(10)	Yes	Area ID; for warehouses, repeat the warehouse ID with the prefix "WH"
AREA_NAME	Char(120)	Yes	Area name; for warehouses, repeat the warehouse name
CHAIN	Number(10)	Yes	Chain ID; for warehouses, repeat the warehouse ID with the prefix "WH"
CHAIN_NAME	Char(120)	Yes	Chain name; for warehouses, repeat the warehouse name
COMPANY	Number(4)	Yes	Company ID
COMPANY_NAME	Char(120)	Yes	Company name
COMPANY_CURRENC Y	Char(3)	Yes	The currency code for the base currency defined in system options
LOC_TYPE	Char(1)	Yes	'S' for store, 'W' for warehouse
LOC_TYPE_NAME	Char(120)	Yes	Store or Warehouse depending on location type
PHYSICAL_WH	Number(10)	Yes	Physical warehouse ID for warehouses, repeat store ID for store
PHYSICAL_WH_NAME	Char(120)	Yes	Physical warehouse name for warehouse, repeat store name for stores
CHANNEL_ID	Number(4)	Yes	Channel ID for the store or virtual warehouse; if no channel is defined, then NA

Field Name	Field Type	Required	Description
CHANNEL_NAME	Char(120)	Yes	Channel name; if no channel is defined, then 'unassigned'
STORE_CLASS	Char(1)	Yes	For stores, the store class ID; for warehouses or if no store class is defined; then NA.
STORE_CLASS_DESC RIPTION	Char(250)	Yes	For stores, the description of the store class, if defined; for warehouses or if not defined for a store, then 'unassigned'.
STORE_FORMAT	Number(4)	Yes	For stores, the store format ID; for warehouses or if no store class is defined; then NA.
STORE_FORMAT_NA ME	Char(60)	Yes	For stores, the description of the store format, if defined; for warehouses or if not defined for a store, then 'unassigned'.

Out of Stock Extract to Forecasting (BDI_RDF_StockOut_Tx_PF_From_RMS_JOB)



Note:

This module replaces the soutdnl.pc module from previous releases.

Module Name	BDI_RDF_StockOut_Tx_PF_From_RMS_JOB
Description	Extracts out of stock item location information to Forecasting
Functional Area	Foundation
Module Type	Integration
Module Technology	BDI job
Catalog ID	N/A
Runtime Parameters	StockOut_Tx_ProcessFlow_From_RMS StockOut_Tx_Extractor

Design Overview

This process extracts items which are out of stock for use by Forecasting on a weekly basis. This integration sends all item/store combinations that meet the criteria for review and have a stock-on-hand position of less than or equal to zero at the end of the week.

Key assumptions for this integration:

- Only stockholding stores are included in this integration.
- Only forecasted items are included in this integration.
- Only item/store combinations that have a status of Active and a ranged flag of Yes are reviewed for stock out conditions.

- Only item/store combinations that have a last sold date that is between the end of week date and x number of days back are reviewed for stock out conditions, where x is the value reports system option value Days Since Last Transaction.
- The intended targets for this integration are
 - Oracle Retail Demand Forecasting Cloud Service (RDFCS)

This process utilizes BDI (Bulk Data Integration) to facilitate the bulk data movement to the target applications. The batch job BDI_RDF_StockOut_Tx_PF_From_RMS_JOB is defined in the Merchandising JOS batch job admin as follows:

```
<job id="BDI_RDF_StockOut_Tx_PF_From_RMS_JOB" version="1.0" xmlns="http://
xmlns.jcp.org/xml/ns/javaee">
  <properties>
    <property name="description" value="Extracts information for items which
are out of stock for use by the RDF application"/>
  </properties>
  <step id="batchlet-step">
    <batchlet ref="BDIInvokerBatchlet">
      <properties>
        <property name="bdiProcessFlowUrl"
value="#SysOpt.bdiProcessFlowUrl"/>
        <property name="bdiProcessFlowCredential"
value="#SysOpt.bdiProcessFlowUrlUserAlias"/>
        <property name="predicateDS" value="RmsDBDS"/>
        <property name="predicateFunction"
value="RMS_BATCH_STATUS_SQL.GET_EOW_RUN_SIGNAL"/>
      </properties>
    </batchlet>
    <end on="COMPLETED"/>
  </step>
</job>
```

When the batch job BDI_RDF_StockOut_Tx_PF_From_RMS_JOB is executed, a batchlet (BDIInvokerBatchlet) starts the execution flow. It calls a PLSQL function (RMS_BATCH_STATUS_SQL.GET_EOW_RUN_SIGNAL) to ensure the process flow is only executed on an end-of-week date. If the vdate is an end-of-week date, it invokes a BDI process flow (StockOut_Tx_ProcessFlow_From_RMS) to perform a series of steps to extract, download, and transport the downloaded files to target applications:

- Extractor job (StockOut_Tx_ExtractorJob) calls BDI_RDF_SQL.STOCKOUT_UP function to extract data from the Merchandising view V_BDI_RDF_STOCKOUT to outbound staging table STOCKOUT_OUT.
- A generic BDI Downloader file creator job writes out of stock item information from the STOCKOUT_OUT table into a comma-delimited flat file, which will be consumed by the target applications. A zero-byte trigger file is also generated to signal that the extract process was successful.
- The downloaded data files and trigger files are written to designated location as configured through BDI system options:
 - RDF_outboundLocation

Scheduling Constraints

Schedule Information	Description
Processing Cycle	End of Day
Frequency	Scheduled daily but files will only be generated weekly on End of Week date.
Scheduling Considerations	N/A
Pre-Processing	N/A
Post-Processing	N/A
Threading Scheme	N/A

Restart/Recovery

N/A

Key Tables Affected

Table	Select	Insert	Update	Delete
V_BDI_RDF_STOCKOUT	Yes	No	No	No
STOCKOUT_OUT	Yes	Yes	No	Yes
BDI_DWNLDR_IFACE_MOD_DA TA_CTL	Yes	No	No	No
BDI_DWNLDR_IFACE_DATA_C TL	Yes	No	No	No

Integration Contract

The flat file will contain the following information:

Field Name	Field Type	Required	Description
ITEM	Varchar2(25)	Yes	Item that is out of stock at the store.
STORE	Number(10)	Yes	Store that is out of stock for the item.
EOW_DATE	Date	Yes	Indicates the end of week date for which the data applies.
OUT_OF_STOCK	Number(1)	Yes	Flag to indicate if the item/store is out of stock at end of week. This will always be 1, as only out-of-stock items are sent.

Store Extract to Planning and Forecasting (BDI_RPAS_Store_Fnd_PF_From_RMS_JOB)

Module Name	BDI_RPAS_Store_Fnd_PF_From_RMS_JOB bdi_merch_extract_to_file_wrapper.sh bdi_rpas_store_extract.ksh
Description	Extracts store information to RPAS
Functional Area	Foundation
Module Type	Integration
Module Technology	BDI, shell scripts
Catalog ID	N/A
Runtime Parameters	Store_Fnd_ProcessFlow_From_RMS Store_Fnd_Extractor Database connection, download file location, filename, trigger filename

Design Overview

This program extracts store data to planning and forecasting on a weekly basis. This data supplements the store information included in the organizational hierarchy feed.

Key assumptions for this integration:

- Both stockholding and non-stockholding stores are included.
- Both company and franchise types of stores are included.
- All stores are sent each time this process is run.
- Planning will derive the status of the store (e.g. open or closed) based on the dates sent in this integration. For example, if the open date is in the past and there is no close date defined or it is a future date, then the store is considered open.
- All descriptions are sent in the primary language as defined in Merchandising.
- The intended targets for this integration are
 - Oracle Retail Merchandise Financial Planning Cloud Service (MFPCS)
 - Oracle Retail Demand Forecasting Cloud Service (RDFCS)
 - Assortment & Item Planning for Fashion/Softlines Cloud Service and Assortment & Item Planning Enterprise Edition Cloud Service (referred to jointly as APCS)

This program utilizes BDI (Bulk Data Integration) to facilitate the bulk data movement from Merchandising to the target applications.

The batch job BDI_RPAS_Store_Fnd_PF_From_RMS_JOB is defined in the Merchandising JOS batch job admin as follows:

```
<job id="BDI_RPAS_Store_Fnd_PF_From_RMS_JOB" version="1.0" xmlns="http://
xmlns.jcp.org/xml/ns/javaee">
  <properties>
    <property name="description" value="Extracts store information and
writes it out to a flat file for processing by both MFP and RDF."/>
  </properties>
```

```

<step id="batchlet-step">
  <batchlet ref="BDIInvokerBatchlet">
    <properties>
      <property name="bdiProcessFlowUrl" value="#SysOpt.bdiProcessFlowUrl"/>
      <property name="bdiProcessFlowCredential"
value="#SysOpt.bdiProcessFlowUrlUserAlias"/>
      <property name="predicateDS" value="RmsDBDS"/>
      <property name="predicateFunction"
value="RMS_BATCH_STATUS_SQL.GET_EOW_RUN_SIGNAL"/>
    </properties>
  </batchlet>
  <end on="COMPLETED"/>
</step>
</job>

```

When the batch job `BDI_RPAS_Store_Fnd_PF_From_RMS_JOB` is executed, a batchlet (`BDIInvokerBatchlet`) starts the execution flow. It calls a PLSQL function (`RMS_BATCH_STATUS_SQL.GET_EOW_RUN_SIGNAL`) to ensure the process flow is only executed on an end-of-week date. If the `vdate` is an end-of-week date, it invokes a BDI process flow (`Store_Fnd_ProcessFlow_From_RMS`) to perform a series of steps to extract, download, and transport the downloaded files to target applications:

- Extractor job (`Store_Fnd_Extractor`) calls `BDI_ORG_SQL.STORE_UP` function to extract data from Merchandising tables to BDI outbound staging table `STORE_OUT`.
- Downloader file creator job calls the wrapper script, `bdi_merch_extract_to_file_wrapper.sh`, to set the runtime parameters on environment variables. This script will then call `bdi_rpas_store_extract.ksh` to write store information from the `STORE_OUT` table into a comma-delimited flat file, which will be consumed by the target application. A zero-byte trigger file is also generated to signal that the extract process was successful. Two separate copies of the data file and the trigger file are sent to the target application.
- The downloaded data files and trigger files are written to designated locations as configured via BDI system options:
 - `MFP_outboundLocation`
 - `RDF_outboundLocation`
 - `AP_outboundLocation`
 - `IP_outboundLocation`

Scheduling Constraints

Schedule Information	Description
Processing Cycle	End of Day
Frequency	Scheduled daily but files will only be generated weekly on End of Week date.
Scheduling Considerations	N/A
Pre-Processing	N/A
Post-Processing	N/A
Threading Scheme	N/A

Restart/Recovery

N/A

Key Tables Affected

Table	Select	Insert	Update	Delete
STORE	Yes	No	No	No
CHANNELS	Yes	No	No	No
CODE_DETAIL	Yes	No	No	No
STORE_FORMAT	Yes	No	No	No
LANG	Yes	No	No	No
VAT_REGION	Yes	No	No	No
TSFZONE	Yes	No	No	No
STORE_OUT	Yes	Yes	No	Yes
BDI_DWNLDR_IFACE_MOD_DATA_CTL	Yes	No	No	No
BDI_DWNLDR_IFACE_DATA_CTL	Yes	No	No	No

Integration Contract

The flat file will contain the following information:

Field Name	Field Type	Required	Description
STORE	Number(10)	Yes	Store ID
STORE_NAME	Char(150)	Yes	Store name
DISTRICT	Number(10)	Yes	District in which the store is a member.
STORE_CLOSE_DATE	DATE	Yes	Date on which the store closed. If NULL, set to NA.
STORE_OPEN_DATE	DATE	Yes	Date on which the store opened
REMODEL_DATE	DATE	Yes	Date on which the store was last remodeled. If NULL, set to NA.
STORE_CLASS	Char(1)	Yes	ID for the store class of which the store is a member.
STORE_CLASS_DESCRIPTION	Char(250)	Yes	Store class description
STORE_FORMAT	Number(4)	Yes	Store format. If NULL, set to NA.
STORE_FORMAT_NAME	Char(60)	Yes	Store format name. If NULL, set to 'unassigned'.
CURRENCY	Char(3)	Yes	Currency under which the store operates.

Field Name	Field Type	Required	Description
STORE_TYPE	Char(6)	Yes	Indicates whether the store is a franchise (F) or company store (C).
STOCKHOLDING_IND	Char(1)	Yes	Indicates whether the store can hold stock. Valid values are Y or N.

Supplier Extract to Planning (BDI_RPAS_Supplier_Fnd_PF_From_RMS_JOB)

Module Name	BDI_RPAS_Supplier_Fnd_PF_From_RMS_JOB bdi_merch_extract_to_file_wrapper.sh bdi_rpas_supplier_extract.ksh
Description	Extracts Supplier information to Planning
Functional Area	Foundation
Module Type	Integration
Module Technology	BDI job, shell scripts
Catalog ID	N/A
Runtime Parameters	Supplier_Fnd_ProcessFlow_From_RMS Supplier_Fnd_Extractor Database connection, download file location, filename, trigger filename

Design Overview

This process extracts supplier data to Planning on a weekly basis.

Key assumptions for this integration:

- All active, orderable supplier sites will be included in this integration each time it runs.
- Retailers will not create a Diff with an ID of 'SUP'.
- In order to meet the format required by Planning, the UDA description in this extract is hard coded to "Supplier" and does not take into account the primary language configuration in Merchandising.
- The intended targets for this integration are
 - Assortment & Item Planning for Fashion/Softlines Cloud Service and Assortment & Item Planning Enterprise Edition Cloud Service (referred to jointly as APCS)

This process utilizes BDI (Bulk Data Integration) to facilitate the bulk data movement to Planning.

The batch job BDI_RPAS_Supplier_Fnd_PF_From_RMS_JOB is defined in the Merchandising JOS batch job admin as follows:

```
<job id="BDI_RPAS_Supplier_Fnd_PF_From_RMS_JOB" version="1.0" xmlns="http://
xmlns.jcp.org/xml/ns/javaee">
  <properties>
    <property name="description" value="Extracts Supplier information and writes it
out to a flat file for processing by AP and IP."/>
  </properties>
  <step id="batchlet-step">
```

```

<batchlet ref="BDIInvokerBatchlet">
  <properties>
    <property name="bdiProcessFlowUrl" value="#SysOpt.bdiProcessFlowUrl"/>
    <property name="bdiProcessFlowCredential"
value="#SysOpt.bdiProcessFlowUrlUserAlias"/>
    <property name="predicateDS" value="RmsDBDS"/>
    <property name="predicateFunction"
value="RMS_BATCH_STATUS_SQL.GET_EOW_RUN_SIGNAL"/>
  </properties>
</batchlet>
<end on="COMPLETED"/>
</step>
</job>

```

When the batch job `BDI_RPAS_Supplier_Fnd_PF_From_RMS_JOB` is executed, a batchlet (`BDIInvokerBatchlet`) starts the execution flow. It calls a PLSQL function (`RMS_BATCH_STATUS_SQL.GET_EOW_RUN_SIGNAL`) to ensure the process flow is only executed on an end-of-week date. If the `vdate` is an end-of-week date, it invokes a BDI process flow (`Supplier_Fnd_ProcessFlow_From_RMS`) to perform a series of steps to extract, download, and transport the downloaded files to target applications:

- Extractor job (`Supplier_Fnd_Extractor`) calls `BDI_FOUNDATION_SQL.SUPS_UP` function to extract data from the Merchandising table `SUPS` to BDI outbound staging table `SUPS_OUT`. Only supplier sites will be extracted.
- Downloader file creator job calls the wrapper script, `bdi_merch_extract_to_file_wrapper.sh`, to set the runtime parameters on environment variables. This script will then call `bdi_rpas_supplier_extract.ksh` to write supplier information from the `SUPS_OUT` table into a comma-delimited flat file, which will be consumed by the target applications. A zero-byte trigger file is also generated to signal that the extract process was successful. Two separate copies of the data file and the trigger file are sent to the target applications.
- The downloaded data files and trigger files are written to designated locations as configured via BDI system options:
 - `AP_outboundLocation`
 - `IP_outboundLocation`

Scheduling Constraints

Schedule Information	Description
Processing Cycle	End of Day
Frequency	Scheduled daily but files will only be generated weekly on End of Week date.
Scheduling Considerations	N/A
Pre-Processing	N/A
Post-Processing	N/A
Threading Scheme	N/A

Restart/Recovery

N/A

Key Tables Affected

Table	Select	Insert	Update	Delete
SUPS	Yes	No	No	No
SUPS_OUT	Yes	Yes	No	Yes
BDI_DWNLDR_IFACE_MOD_DA TA_CTL	Yes	No	No	No
BDI_DWNLDR_IFACE_DATA_C TL	Yes	No	No	No

Integration Contract

The flat file will contain the following information:

Field Name	Field Type	Required	Description
UDA_ID	Char(6)	Yes	Hardcoded 'SUP'
UDA_DESC	Char(120)	Yes	Hardcoded 'Supplier'
SUPPLIER	Char(30)	Yes	The supplier site ID.
SUP_NAME	Char(120)	Yes	The supplier site name in the primary language.

Transaction Data Extract to Planning (BDI_MFP_TranData_Tx_PF_From_RMS_JOB)

Module Name	BDI_MFP_TranData_Tx_PF_From_RMS_JOB
Description	Extracts Transaction data to Planning from RMS
Functional Area	Transactional Data
Module Type	Integration
Module Technology	BDI job
Catalog ID	N/A
Runtime Parameters	TranData_Tx_ProcessFlow_From_RMS TranData_Tx_Extractor

Design Overview

This process extracts transactional data to planning on a weekly basis, aggregating all transactions that posted in the last week, which could include transactions for previous weeks that posted late.

Key assumptions in this integration:

- Only orderable, inventoried, approved transaction items are included in the integration.
- Pack items are not included in this integration; any transactions involving pack items will be sent in terms of the pack's component items.
- Cost and retail values sent in primary currency.

- Sales sent will always be net sales. If gross sales are needed in Planning, then net sales can be combined with returns.
- Retail values will include VAT if the system option to include VAT in the stock ledger is set to include VAT so that the retail values in this integration are consistent with other data sent to planning.
- All unit values are sent in terms of the standard unit of measure for the item.
- Late posted transactions included in this integration may be for any week in the open stock ledger month, as well as any week in the previous month that posted during the week but before the previous month closed, if the month close ran during the current week.
- The intended targets for this integration are
 - Oracle Retail Merchandise Financial Planning Cloud Service (MFPCS)
 - Assortment & Item Planning for Fashion/Softlines Cloud Service and Assortment & Item Planning Enterprise Edition Cloud Service (referred to jointly as APCS)

This process utilizes BDI (Bulk Data Integration) to facilitate the bulk data movement to the target applications. The batch job BDI_MFP_TranData_Tx_PF_From_RMS_JOB is defined in the Merchandising JOS batch job admin as follows:

```
<job id="BDI_MFP_TranData_Tx_PF_From_RMS_JOB" version="1.0" xmlns="http://
xmlns.jcp.org/xml/ns/javaee">
  <properties>
    <property name="description" value="Extracts information regarding
transaction data for use by the MFP application"/>
  </properties>
  <step id="batchlet-step">
    <batchlet ref="BDIInvokerBatchlet">
      <properties>
        <property name="bdiProcessFlowUrl"
value="#SysOpt.bdiProcessFlowUrl"/>
        <property name="bdiProcessFlowCredential"
value="#SysOpt.bdiProcessFlowUrlUserAlias"/>
        <property name="predicateDS" value="RmsDBDS"/>
        <property name="predicateFunction"
value="RMS_BATCH_STATUS_SQL.GET_EOW_RUN_SIGNAL"/>
      </properties>
    </batchlet>
    <end on="COMPLETED"/>
  </step>
</job>
```

When the batch job BDI_MFP_TranData_Tx_PF_From_RMS_JOB is executed, a batchlet (BDIInvokerBatchlet) starts the execution flow. It calls a PLSQL function (RMS_BATCH_STATUS_SQL.GET_EOW_RUN_SIGNAL) to ensure the process flow is only executed on an end-of-week date. If the vdate is an end-of-week date, it invokes a BDI process flow (Trandata_Tx_ProcessFlow_From_RMS) to perform a series of steps to extract, download, and transport the downloaded files to target applications:

- Extractor job (TranData_Tx_Extractor) calls BDI_MFP_SQL.TRAN_DATA_UP function to extract data from the Merchandising view V_BDI_MFP_TRAN_DATA to BDI outbound staging table TRAN_DATA_OUT.
- A generic BDI Downloader file creator job writes transactional information from the TRAN_DATA_OUT table into a comma-delimited flat file, which will be consumed

by the target applications. A zero-byte trigger file is also generated to signal that the extract process was successful. Separate copies of the data file and the trigger file are sent to the target applications.

- The downloaded data files and trigger files are written to designated MFP location as configured via BDI system options:
 - MFP_outboundLocation
 - AP_outboundLocation
 - IP_outboundLocation

Scheduling Constraints

Schedule Information	Description
Processing Cycle	End of Day
Frequency	Scheduled daily but files will only be generated weekly on End of Week date
Scheduling Considerations	N/A
Pre-Processing	N/A
Post-Processing	N/A
Threading Scheme	N/A

Restart/Recovery

N/A

Key Tables Affected

Table	Select	Insert	Update	Delete
V_BDI_MFP_TRAN_DATA	Yes	No	No	No
TRAN_DATA_OUT	Yes	Yes	No	Yes
BDI_DWNLDR_IFACE_MOD_DATA_CTL	Yes	No	No	No
BDI_DWNLDR_IFACE_DATA_CTL	Yes	No	No	No

Integration Contract

The flat file will contain the following information:

Field Name	Field Type	Required	Description
EOW	Date	Yes	Indicates the end of week date that the information pertains to.
ITEM	Varchar2(25)	Yes	Transaction level item only.
LOCATION	Number(10)	Yes	Could be a store or virtual warehouse.

Field Name	Field Type	Required	Description
LOC_TYPE	Varchar2(1)	Yes	Indicates if the location is a store or warehouse - S = Store; W = Warehouse.
CLEAR_IND	Number(1)	Yes	If Y, item/location is currently on clearance.
NET_SALES_REG_UNIT S	Number(12,4)	No	tran_data_history.units: tran_code = 1 and sales type = R
NET_SALES_REG_COS T	Number(20,4)	No	tran_data_history.total_cost: tran_code = 1 and sales type = R
NET_SALES_REG_RETA IL	Number(20,4)	No	tran_data_history.total_retail: tran_code = 1 and sales type = R
NET_SALES_PROMO_U NITS	Number(12,4)	No	tran_data_history.units: tran_code = 1 and sales type = P
NET_SALES_PROMO_C OST	Number(20,4)	No	tran_data_history.total_cost: tran_code = 1 and sales type = P
NET_SALES_PROMO_R ETAIL	Number(20,4)	No	tran_data_history.total_retail: tran_code = 1 and sales type = P
NET_SALES_CLEAR_UN ITS	Number(12,4)	No	tran_data_history.units: tran_code = 1 and sales type = C
NET_SALES_CLEAR_C OST	Number(20,4)	No	tran_data_history.total_cost: tran_code = 1 and sales type = C
NET_SALES_CLEAR_RE TAIL	Number(20,4)	No	tran_data_history.total_retail: tran_code = 1 and sales type = C
NET_SALES_REG_RETA IL_VAT_EXCL	Number(20,4)	No	tran_data_history.total_retail: tran_code = 2 and sales type = R
NET_SALES_PROMO_R TL_VAT_EXCL	Number(20,4)	No	tran_data_history.total_retail: tran_code = 2 and sales type = P
NET_SALES_CLR_RETA IL_VAT_EXCL	Number(20,4)	No	tran_data_history.total_retail: tran_code = 2 and sales type = C
RETURNS_REG_UNITS	Number(12,4)	No	tran_data_history.units: tran_code = 4 and sales type = R
RETURNS_REG_COST	Number(20,4)	No	tran_data_history.total_cost: tran_code = 4 and sales type = R
RETURNS_REG_RETAIL	Number(20,4)	No	tran_data_history.total_retail: tran_code = 4 and sales type = R
RETURNS_PROMO_UNI TS	Number(12,4)	No	tran_data_history.units: tran_code = 4 and sales type = P
RETURNS_PROMO_CO ST	Number(20,4)	No	tran_data_history.total_cost: tran_code = 4 and sales type = P
RETURNS_PROMO_RET AIL	Number(20,4)	No	tran_data_history.total_retail: tran_code = 4 and sales type = P
RETURNS_CLEAR_UNIT S	Number(20,4)	No	tran_data_history.units: tran_code = 4 and sales type = C
RETURNS_CLEAR_COS T	Number(20,4)	No	tran_data_history.total_cost: tran_code = 4 and sales type = C
RETURNS_CLEAR_RET AIL	Number(20,4)	No	tran_data_history.total_cost: tran_code = 4 and sales type = C

Field Name	Field Type	Required	Description
REG_MARKDOWN_RET AIL	Number(20,4)	No	tran_data_history.total_retail: tran_code 13 - tran_code 14 (Markdown Cancel) - tran_code 11 (Markup)
PROMO_MARKDOWN_R ETAIL_REG	Number(20,4)	No	tran_data_history.total_retail: tran_code = 15 - if the item is not on clearance EOW
PROMO_MARKDOWN_R ETAIL_CLEAR	Number(20,4)	No	tran_data_history.total_retail: tran_code = 15 - if the item is on clearance EOW
CLEAR_MARKDOWN_R ETAIL	Number(20,4)	No	tran_data_history.total_retail: tran_code = 16
WF_MARKDOWN_RETAI L	Number(20,4)	No	tran_data_history.total_retail: tran_code = 85
WF_MARKUP_RETAIL	Number(20,4)	No	tran_data_history.total_retail: tran_code = 84
SHRINK_UNITS	Number(12,4)	No	tran_data_history.units: tran_code 22
SHRINK_COST	Number(20,4)	No	tran_data_history.total_cost: tran_code 22
SHRINK_RETAIL	Number(20,4)	No	tran_data_history.total_retail: tran_code 22
DEAL_INCOME_COST	Number(20,4)	No	tran_data_history.total_cost: tran_code 6 & 7
RECEIPT_UNITS	Number(12,4)	No	tran_data_history.units: tran_code = 20 + 44
RECEIPT_COST	Number(20,4)	No	tran_data_history.total_cost: tran_code = 20 + 44
RECEIPT_RETAIL	Number(20,4)	No	tran_data_history.total_retail: tran_code = 20 + 44
NON_SHRINK_ADJ_UNI TS	Number(12,4)	No	tran_data_history.units: tran_code = 23
NON_SHRINK_ADJ_CO ST	Number(20,4)	No	tran_data_history.total_cost: tran_code = 23
NON_SHRINK_ADJ_RET AIL	Number(20,4)	No	tran_data_history.total_retail: tran_code = 23
DEAL_INCOME_PURCH ASES	Number(20,4)	No	tran_data_history.total_cost: tran_code 7
MARKUP	Number(20,4)	No	tran_data_history.total_retail: tran_code 11
MARKDOWN_CANCEL	Number(20,4)	No	tran_data_history.total_retail: tran_code 14
INTERCOMPANY_MARK UP	Number(20,4)	No	tran_data_history.total_retail: tran_code 17
INTERCOMPANY_MARK DOWN	Number(20,4)	No	tran_data_history.total_retail: tran_code 18
RTV_UNITS	Number(12,4)	No	tran_data_history.units: tran_code = 24

Field Name	Field Type	Required	Description
RTV_COST	Number(20,4)	No	tran_data_history.total_cost: tran_code = 24
RTV_RETAIL	Number(20,4)	No	tran_data_history.total_retail: tran_code = 24
TSF_IN_UNITS	Number(12,4)	No	tran_data_history.units: tran_code = 30
TSF_IN_COST	Number(20,4)	No	tran_data_history.total_cost: tran_code = 30
TSF_IN_RETAIL	Number(20,4)	No	tran_data_history.total_retail: tran_code = 30
TSF_IN_UNITS_BOOK	Number(12,4)	No	tran_data_history.units: tran_code = 31
TSF_IN_COST_BOOK	Number(20,4)	No	tran_data_history.total_cost: tran_code = 31
TSF_IN_RETAIL_BOOK	Number(20,4)	No	tran_data_history.total_retail: tran_code = 31
TSF_OUT_UNITS	Number(12,4)	No	tran_data_history.units: tran_code = 32
TSF_OUT_COST	Number(20,4)	No	tran_data_history.total_cost: tran_code = 32
TSF_OUT_RETAIL	Number(20,4)	No	tran_data_history.total_retail: tran_code = 32
TSF_OUT_UNITS_BOOK	Number(12,4)	No	tran_data_history.units: tran_code = 33
TSF_OUT_COST_BOOK	Number(20,4)	No	tran_data_history.total_cost: tran_code = 33
TSF_OUT_RETAIL_BOO K	Number(20,4)	No	tran_data_history.total_retail: tran_code = 33
RECLASS_IN_UNITS	Number(12,4)	No	tran_data_history.units: tran_code = 34
RECLASS_IN_COST	Number(20,4)	No	tran_data_history.total_cost: tran_code = 34
RECLASS_IN_RETAIL	Number(20,4)	No	tran_data_history.total_retail: tran_code = 34
RECLASS_OUT_UNITS	Number(12,4)	No	tran_data_history.units: tran_code = 36
RECLASS_OUT_COST	Number(20,4)	No	tran_data_history.total_cost: tran_code = 36
RECLASS_OUT_RETAIL	Number(20,4)	No	tran_data_history.total_retail: tran_code = 36
TSF_IN_UNITS_ICT	Number(12,4)	No	tran_data_history.units: tran_code = 37
TSF_IN_COST_ICT	Number(20,4)	No	tran_data_history.total_cost: tran_code = 37
TSF_IN_RETAIL_ICT	Number(20,4)	No	tran_data_history.total_retail: tran_code = 37

Field Name	Field Type	Required	Description
TSF_OUT_UNITS_ICT	Number(12,4)	No	tran_data_history.units: tran_code = 38
TSF_OUT_COST_ICT	Number(20,4)	No	tran_data_history.total_cost: tran_code = 38
TSF_OUT_RETAIL_ICT	Number(20,4)	No	tran_data_history.total_retail: tran_code = 38
INTERCOMPANY_MARG IN	Number(20,4)	No	tran_data_history.total_retail: tran_code = 39
TSF_RECEIPT_UNITS	Number(12,4)	No	tran_data_history.units: tran_code = 44
TSF_RECEIPT_COST	Number(20,4)	No	tran_data_history.total_cost: tran_code = 44
TSF_RECEIPT_RETAIL	Number(20,4)	No	tran_data_history.total_retail: tran_code = 44
RTV_RESTOCK_FEE	Number(20,4)	No	tran_data_history.total_cost: tran_code = 65
FRANCHISE_SALES_UN ITS	Number(12,4)	No	tran_data_history.units: tran_code = 82
FRANCHISE_SALES_CO ST	Number(20,4)	No	tran_data_history.total_cost: tran_code = 82
FRANCHISE_SALES_RE TAIL	Number(20,4)	No	tran_data_history.total_retail: tran_code = 82
FRANCHISE_RETURNS _UNITS	Number(12,4)	No	tran_data_history.units: tran_code = 83
FRANCHISE_RETURNS _COST	Number(20,4)	No	tran_data_history.total_cost: tran_code = 83
FRANCHISE_RETURNS _RETAIL	Number(20,4)	No	tran_data_history.total_retail: tran_code = 83
FRANCHISE_RESTOCK_ FEE	Number(20,4)	No	tran_data_history.total_cost: tran_code = 86

UDA Extract to Planning (BDI_RPAS_UdaAndUdaValues_Fnd_PF_From_RMS_JOB)

Module Name	BDI_RPAS_UdaAndUdaValues_Fnd_PF_From_RMS_JOB bdi_merch_extract_to_file_wrapper.sh bdi_rpas_uda_extract.ksh
Description	Extracts LOV Type UDA information to Planning
Functional Area	Foundation
Module Type	Integration
Module Technology	BDI job, shell scripts
Catalog ID	N/A

Runtime Parameters	UdaAndUdaValues_Fnd_ProcessFlow_From_RMS Uda_Fnd_Extractor UdaValues_Fnd_Extractor Database connection, download file location, filename, trigger filename
---------------------------	---

Design Overview

This process extracts its UDA data to Planning on a weekly basis.

Key assumptions for this integration:

- The full set of user defined attributes (UDAs) is included in this integration each time it runs.
- Only list of value type UDAs will be included in the integration.
- The intended targets for this integration are
 - Assortment & Item Planning for Fashion/Softlines Cloud Service and Assortment & Item Planning Enterprise Edition Cloud Service (referred to jointly as APCS)

This process utilizes BDI (Bulk Data Integration) to facilitate the bulk data movement to Planning. The batch job BDI_RPAS_UdaAndUdaValues_Fnd_PF_From_RMS_JOB is defined in the Merchandising JOS batch job admin as follows:

```
<job id="BDI_RPAS_UdaAndUdaValues_Fnd_PF_From_RMS_JOB" version="1.0"
xmlns="http://xmlns.jcp.org/xml/ns/javaee">
  <properties>
    <property name="description" value="Extracts LOV Type UDA information and
writes it out to a flat file for processing by AP and IP."/>
  </properties>
  <step id="batchlet-step">
    <batchlet ref="BDIInvokerBatchlet">
      <properties>
        <property name="bdiProcessFlowUrl" value="#SysOpt.bdiProcessFlowUrl"/>
        <property name="bdiProcessFlowCredential"
value="#SysOpt.bdiProcessFlowUrlUserAlias"/>
        <property name="predicateDS" value="RmsDBDS"/>
        <property name="predicateFunction"
value="RMS_BATCH_STATUS_SQL.GET_EOW_RUN_SIGNAL"/>
      </properties>
    </batchlet>
    <end on="COMPLETED"/>
  </step>
</job>
```

When the batch job BDI_RPAS_UdaAndUdaValues_Fnd_PF_From_RMS_JOB is executed, a batchlet (BDIInvokerBatchlet) starts the execution flow. It calls a PLSQL function (RMS_BATCH_STATUS_SQL.GET_EOW_RUN_SIGNAL) to ensure the process flow is only executed on an end-of-week date. If the vdate is an and-of-week date, it invokes a BDI process flow (UdaAndUdaValues_Fnd_ProcessFlow_From_RMS) to perform a series of steps to extract, download, and transport the downloaded files to target applications:

- Extractor jobs (Uda_Fnd_Extractor, UdaValues_Fnd_Extractor) call respective BDI_FOUNDATION_SQL functions to extract data from Merchandising tables

UDA and UDA_VALUES to BDI outbound staging tables UDA_OUT and UDA_VALUES_OUT.

- Downloader file creator job calls the wrapper script, `bdi_merch_extract_to_file_wrapper.sh`, to set the runtime parameters on environment variables. This script will then call `bdi_rpas_uda_extract.ksh` to write UDA information from the UDA_OUT and UDA_VALUES_OUT tables into a comma-delimited flat file, which will be consumed by the target applications. Only LOV type UDAs will be extracted. A zero-byte trigger file is also generated to signal that the extract process was successful. Separate copies of the data file and the trigger file are sent to the target applications.
- The downloaded data files and trigger files are written to designated locations as configured via BDI system options:
 - AP_outboundLocation
 - IP_outboundLocation

Scheduling Constraints

Schedule Information	Description
Processing Cycle	End of Day
Frequency	Scheduled daily but files will only be generated weekly on End of Week date.
Scheduling Considerations	N/A
Pre-Processing	N/A
Post-Processing	N/A
Threading Scheme	N/A

Restart/Recovery

N/A

Key Tables Affected

Table	Select	Insert	Update	Delete
UDA	Yes	No	No	No
UDA_VALUES	Yes	No	No	No
UDA_OUT	Yes	Yes	No	Yes
UDA_VALUES_OUT	Yes	Yes	No	Yes
BDI_DWNLDR_IFACE_MOD_DATA_CTL	Yes	No	No	No
BDI_DWNLDR_IFACE_DATA_CTL	Yes	No	No	No

Integration Contract

The flat file will contain the following information:

Field Name	Field Type	Required	Description
UDA_ID	Number(5)	Yes	The ID of the UDA assigned to the item.
UDA_DESC	Char(120)	Yes	The description of the UDA (for example, Fabric Content).
UDA_VALUE	Number(5)	Yes	The ID of the UDA value for the UDA assigned to the item.
UDA_VALUE_DESC	Char(250)	Yes	The description of the UDA value (for example, Cotton).

UDA Item Extract to Planning and Forecasting (BDI_RDF_UdaltemLov_Fnd_From_RMS_JOB)

Module Name	BDI_RDF_UdaltemLov_Fnd_From_RMS_JOB bdi_merch_extract_to_file_wrapper.sh bdi_rdf_itemuda_extract.ksh
Description	Extracts information for LOV type of UDAs to Planning and Forecasting
Functional Area	Foundation
Module Type	Integration
Module Technology	BDI job, shell scripts
Catalog ID	N/A
Runtime Parameters	ItemHdrAndUdaltemLov_Fnd_ProcessFlow_From_RMS ItemHdr_Fnd_Extractor UdaltemLov_Fnd_Extractor Database connection, download file location, filename, trigger filename

Design Overview

This process extracts user-defined attributes (UDAs) assigned to item to Planning and Forecasting on a weekly basis.

Key assumptions for this integration:

- Only list of value (LOV) type UDAs will be included.
- Both forecasted and non-forecasted items are included in this extract, with the forecast flag included.
- Planning and Forecasting can only support a specific UDA being associated with an item once. Merchandising has a configuration that allows the same UDA to be associated with an item more than one time. However, when implementing with Planning or Forecasting, this should be avoided for LOV-type UDAs to prevent issues with interpreting the data. If more than one is associated with the item, then only the last UDA with a particular ID will be visible in Planning and Forecasting.
- The intended targets for this integration are
 - Oracle Retail Demand Forecasting Cloud Service (RDFCS)

- Assortment & Item Planning for Fashion/Softlines Cloud Service and Assortment & Item Planning Enterprise Edition Cloud Service (referred to jointly as APCS)

This process utilizes BDI (Bulk Data Integration) to facilitate the bulk data movement from Merchandising to the target applications. The batch job BDI_RDF_UdaltemLov_Fnd_From_RMS_JOB is defined in the Merchandising JOS batch job admin as follows:

```
<job id="BDI_RDF_UdaltemLov_Fnd_From_RMS_JOB" version="1.0" xmlns="http://
xmlns.jcp.org/xml/ns/javaee">
  <properties>
    <property name="description" value="Extracts UDA item LOV information and
writes it out to a flat file for processing by RDF."/>
  </properties>
  <step id="batchlet-step">
    <batchlet ref="BDIInvokerBatchlet">
      <properties>
        <property name="bdiProcessFlowUrl" value="#SysOpt.bdiProcessFlowUrl"/>
        <property name="bdiProcessFlowCredential"
value="#SysOpt.bdiProcessFlowUrlUserAlias"/>
        <property name="predicateDS" value="RmsDBDS"/>
        <property name="predicateFunction"
value="RMS_BATCH_STATUS_SQL.GET_EOW_RUN_SIGNAL"/>
      </properties>
    </batchlet>
    <end on="COMPLETED"/>
  </step>
</job>
```

When the batch job BDI_RDF_UdaltemLov_Fnd_From_RMS_JOB is executed, a batchlet (BDIInvokerBatchlet) starts the execution flow. It calls a PLSQL function (RMS_BATCH_STATUS_SQL.GET_EOW_RUN_SIGNAL) to ensure the process flow is only executed on an end-of-week date. If the vdate is an end-of-week date, it invokes a BDI process flow (ItemHdrAndUdaltemLov_Fnd_ProcessFlow_From_RMS) to perform a series of steps to extract, download, and transport the downloaded files to the target applications:

- Extractor jobs (ItemHdr_Fnd_Extractor, UdaltemLov_Fnd_Extractor) call respective BDI_ITEM_SQL functions to extract data from Merchandising tables to BDI outbound staging tables ITEM_HDR_OUT and UDA_ITEM_LOV_OUT.
- Downloader file creator job calls the wrapper script, bdi_merch_extract_to_file_wrapper.sh, to set the runtime parameters on environment variables. This script will then call bdi_rdf_itemuda_extract.ksh to write LOV type of UDA information from the ITEM_HDR_OUT and UDA_ITEM_LOV_OUT tables into a comma-delimited flat file, which will be consumed by the target applications. A zero-byte trigger file is also generated to signal that the extract process was successful. The data file and the trigger file are then sent to the target applications.
- The downloaded data file and trigger file are written to designated locations as configured through BDI system options:
 - RDF_outboundLocation
 - AP_outboundLocation
 - IP_outboundLocation

Scheduling Constraints

Schedule Information	Description
Processing Cycle	End of Day
Frequency	Scheduled daily but files will only be generated weekly on End of Week date.
Scheduling Considerations	N/A
Pre-Processing	N/A
Post-Processing	N/A
Threading Scheme	N/A

Restart/Recovery

N/A

Key Tables Affected

Table	Select	Insert	Update	Delete
ITEM_MASTER	Yes	No	No	No
CLASS	Yes	No	No	No
SUBCLASS	Yes	No	No	No
DIFF_GROUP_HEAD	Yes	No	No	No
DIFF_IDS	Yes	No	No	No
SYSTEM_OPTION	Yes	No	No	No
UDA_ITEM_LOV	Yes	No	No	No
UDA	Yes	No	No	No
UDA_VALUES	Yes	No	No	No
UDA_ITEM_LOV_OUT	Yes	Yes	No	Yes
ITEM_HDR_OUT	Yes	Yes	No	Yes
BDI_DWNLDR_IFACE_MOD_DATA_CTL	Yes	No	No	No
BDI_DWNLDR_IFACE_DATA_CTL	Yes	No	No	No

Integration Contract

The flat file will contain the following information:

Field Name	Field Type	Required	Description
ITEM	Char(25)	Yes	The ID of the item.
UDA_ID	Number(5)	Yes	The ID of the UDA assigned to the item.
UDA_DESC	Char(120)	Yes	The description of the UDA (for example, Fabric Content)

Field Name	Field Type	Required	Description
UDA_VALUE	Number(5)	Yes	The ID of the UDA value for the UDA assigned to the item.
UDA_VALUE_DESC	Char(250)	Yes	The description of the UDA value (for example, Cotton).
FORECAST_IND	Char(1)	Yes	Indicates whether or not the item is to be forecasted. Valid values are Y or N.

Weekly Sales Extract to Forecasting (BDI_RDF_WeeklySales_Tx_PF_From_RMS_JOB)

Module Name	BDI_RDF_WeeklySales_Tx_PF_From_RMS_JOB
Description	Extracts weekly sales information to Forecasting
Functional Area	Foundation
Module Type	Integration
Module Technology	BDI job
Catalog ID	N/A
Runtime Parameters	WeeklySales_Tx_ProcessFlow_From_RMS WeeklySales_Tx_Extractor

Design Overview

This process extracts weekly sales for use by Forecasting on a weekly basis. It sends only the sales from the last week.

Key assumptions for this integration:

- This integration sends gross sales. Returns are not netted out of the sales values.
- Warehouse issues are not included in this integration. Only sales for stores.
- Only forecasted items are included in this integration.
- The intended targets for this integration are
 - Oracle Retail Demand Forecasting Cloud Service (RDFCS)

This process utilizes BDI (Bulk Data Integration) to facilitate the bulk data movement to the target applications.

The batch job BDI_RDF_WeeklySales_Tx_PF_From_RMS_JOB is defined in the Merchandising JOS batch job admin as follows:

```
<job id="BDI_RDF_WeeklySales_Tx_PF_From_RMS_JOB" version="1.0" xmlns="http://
xmlns.jcp.org/xml/ns/javaee">
  <properties>
    <property name="description" value="Extracts weekly sales information for use
by the RDF application"/>
  </properties>
  <step id="batchlet-step">
    <batchlet ref="BDIInvokerBatchlet">
      <properties>
        <property name="bdiProcessFlowUrl" value="#SysOpt.bdiProcessFlowUrl"/>
      </properties>
    </batchlet>
  </step>
</job>
```

```

        <property name="bdiProcessFlowCredential"
value="#SysOpt.bdiProcessFlowUrlUserAlias"/>
        <property name="predicateDS" value="RmsDBDS"/>
        <property name="predicateFunction"
value="RMS_BATCH_STATUS_SQL.GET_EOW_RUN_SIGNAL"/>
    </properties>
    </batchlet>
    <end on="COMPLETED"/>
</step>
</job>

```

When the batch job BDI_RDF_WeeklySales_Tx_PF_From_RMS_JOB is executed, a batchlet (BDIInvokerBatchlet) starts the execution flow. It calls a PLSQL function (RMS_BATCH_STATUS_SQL.GET_EOW_RUN_SIGNAL) to ensure the process flow is only executed on an end-of-week date. If the vdate is an end-of-week date, it invokes a BDI process flow (WeeklySales_Tx_ProcessFlow_From_RMS) to perform a series of steps to extract, download, and transport the downloaded files to target applications:

- Extractor job (WeeklySales_Tx_ExtractorJob) calls BDI_RDF_SQL.WEEKLY_SALES_UP function to extract data from a Merchandising view V_BDI_RDF_WEEKLY_SALES to outbound staging table WEEKLY_SALES_OUT.
- A generic BDI Downloader file creator job writes weekly sales information from the WEEKLY_SALES_OUT table into a comma-delimited flat file, which will be consumed by the target applications. A zero-byte trigger file is also generated to signal that the extract process was successful. Separate copies of the data file and the trigger file are sent to the target applications.
- The downloaded data files and trigger files are written to designated location as configured via BDI system options:
 - RDF_outboundLocation

Scheduling Constraints

Schedule Information	Description
Processing Cycle	End of Day
Frequency	Scheduled daily but files will only be generated weekly on End of Week date.
Scheduling Considerations	N/A
Pre-Processing	N/A
Post-Processing	N/A
Threading Scheme	N/A

Restart/Recovery

N/A

Key Tables Affected

Table	Select	Insert	Update	Delete
V_BDI_RDF_WEEKLY_SALES	Yes	No	No	No
WEEKLY_SALES_OUT	Yes	Yes	No	Yes
BDI_DWNLDR_IFACE_MOD_DA TA_CTL	Yes	No	No	No
BDI_DWNLDR_IFACE_DATA_C TL	Yes	No	No	No

Integration Contract

The flat file will contain the following information:

Field Name	Field Type	Required	Description
ITEM	Varchar2(25)	Yes	Indicates the item.
STORE	Number(10)	Yes	Indicates the store.
EOW_DATE	Date	Yes	Indicates the end of week date for which the data applies.
SALES_UNITS	Number(12,4)	No	This value will be the total sales units for the item/location for the week.
SALES_TYPE	Varchar2(1)	Yes	Indicates the sales type. For example, R (Regular Sales), P (Promotional Sales) or C (Clearance Sales).

Sales Audit

The purpose of Sales Audit is to accept transaction data from point-of-sale (POS) and order management (OMS) solutions and move the data through a series of processes that culminate in "clean" data. Data that Sales Audit finds to be inaccurate is brought to the attention of the auditors who can use the features in Sales Audit to correct the exceptions.

For more information on Sales Audit processing see *Merchandising Operations Guide Volume 1*.

This section contains details about the following integration processes used to export data from Sales Audit to other solutions:

- [Download from Sales Audit to Account Clearing House \(ACH\) System \(saexpach\)](#)
- [Download of Escheated Vouchers from Sales Audit for Payment \(saescheat\)](#)
- [Export DSD and Escheatment from Sales Audit to Invoice Matching \(saexpim\)](#)
- [Export from Sales Audit to Oracle Retail Insights \(saexpdw\)](#)
- [Export Inventory Reservation/Release for In Store Customer Order & Layaway Transactions \(saordinvexp\)](#)
- [Export of POS transactions from Sales Audit to Merchandising \(saexprms\)](#)

- [Export of Revised Sale/Return Transactions from ReSA to SIM/SIOCS \(saexpsim\)](#)
- [Export to Universal Account Reconciliation System from Sales Audit \(saexpuar\)](#)
- [Extract of POS Transactions by Store/Date from Sales Audit for Web Search \(ang_sapngen\)](#)
- [Post User Defined Totals from Sales Audit to General Ledger \(saexpgl\)](#)

Download from Sales Audit to Account Clearing House (ACH) System (saexpach)

Module Name	saexpash.pc
Description	Download from Sales Audit to Account Clearing House (ACH) System
Functional Area	Oracle Retail Sales Audit
Module Type	Integration
Module Technology	ProC
Catalog ID	RSA03
Wrapper Script	rmswrap_out.ksh

Schedule

Oracle Retail Merchandising Batch Schedule

Design Overview

This module will post store/day deposit totals to the SA_STORE_ACH table and bank deposit totals for a given day in a file formatted for export to an ACH (Account Clearing House). The ACH export deviations from the typical Sales Audit export in that store/days must be exported even though errors may have occurred for a given day or store (depending on the unit of work defined), and also, the store/day does not need to be closed for the export to occur. The nature of the ACH process is such that as much money as possible must be sent as soon as possible to the consolidating bank. Any adjustments to the amount sent can be made using the sabnkach screen in the online system.

Deposits for store/days that have not been Fully (F) loaded will not be transferred to the consolidating bank. After they are fully loaded, their deposits will be picked up by the next run of this program.

Restart/Recovery

This module is in two distinct parts, with two different logical units of work. Thus, restart/recovery has to be implemented so that the first part does not get reprocessed in case the program is being restarted. Details on the implementation follow.

The first driving cursor in this module retrieves a store/day to generate ACH totals. Once the first cursor is complete, the second retrieves bank locations by account numbers.

The first Logical Unit of Work (LUW) is defined as a unique store/day combination. Records will be fetched, using the first driving cursor, in batches of commit_max_ctr, but processed and committed one store/day at a time.

The first driving cursor will fetch all store/days that have been Fully Loaded (F), whose audit status is Audited (A), HQ Errors Pending (H), or Store Errors Pending (S) and that are ready to be exported to ACH. Before processing starts, a write lock is obtained using `get_lock ()`. This driving cursor only fetches store/days with a `sa_export_log.status` of `SAES_R`. After a store/day is processed, `sa_export_log.status` is set to `SAES_P` so that this store/day will not be selected again if the program is restarted. The commit is performed using `rettek_force_commit` after each store/day has been processed and `sa_export_log` updated, so as to release the lock.

In case a store/day could not be processed due to locking, the store/day information is placed on a list (called locked store/day list) and the next store/day is processed. This list is kept in memory and is available only during processing. If the store for a store/day obtained from the first driving cursor, is on the locked store/day list, then this store/day cannot be processed. This is the case because there is a data dependency such that data from a particular store/day is dependent on data for the same store but at an earlier date. Thus, if a store/day cannot be processed, then subsequent store/days for the same store cannot be processed either. After the driving cursor returns no more data, the program attempts to process each store/day on the list two more times. If the store/day is still locked, then it is skipped entirely and a message is printed to the error log.

The second LUW is a bank account number. Again, records will be fetched in batches of `commit_max_ctr`. The second driving cursor cannot retrieve information by the LUW because it is possible for the store's currency to be different from the local bank's currency. In that case, a currency conversion is needed.

For each store/day, the query should retrieve the required ACH transfer. The latter is determined by adding the estimated deposit for the next day, the adjustment to the estimate for the current day, and any manual adjustment to the estimate.

Since a store can be associated with different accounts at different banks, only accounts that are consolidated should be retrieved. Since it is possible for the local bank to be in a different country than the consolidating bank, the currency of the partner should also be fetched.

Since processing is dependent on the type of account at the RDFI, the account type should be fetched by this cursor.

Due to differences in transaction processing in cases when the bank is outside the United States, the partner's country should also be fetched. The results of the query should be sorted by partner country. The results of the query should also be ordered by accounts.

Security Considerations

The fact that this program automates the transfer of funds on behalf of the user makes it a likely target for electronic theft. It must be made clear that the responsibility of electronic protection lies with the users themselves.

Following are some tips and recommendation to users:

- A specific user should be used to run the program. This user would be the only one (or one of a few) who has access to this program.
- The umask for this user should be set up so as to prevent other users from reading/writing its files. This would ensure that when the output file is created, it would not be accessible to other users.
- The appropriate permissions should be set up on the directory, which holds the ACH files. The most restrictive decision would be to not allow any other user to view the contents of the directory.

- A secure means of communication should be implemented for transferring the file from where it has been created to the ACH network. This may be done through encryption, or by copying the file to a disk and trusting the courier to deliver the files intact.
- The ACH network needs to be secure.

I/O Specification

Integration Type Download from Sales Audit
File Name ACH_ appended with the consolidating routing number, consolidating account number, and current system date.
Integration Contract IntCon000040

Output File

Table 6-23 Output File

Record Name	Field Name	Field Type	Default Value	Description
ACH File Header	Section No.	Number(3)	101	Constant number.
	Console Route No	Number(10)	N/A	The routing number of the consolidating bank.
	Sender ID	Char(10)	N/A	ID used by the Originator to identify itself.
	Current Date	Char(6)	N/A	Vdate in YYMMDD format.
	Day Time	Char(4)	N/A	Time of file creation in HH24MM format.
	File Header No.	Number(7)	0094101	Constant number.
	Console Bank Name	Char(23)	N/A	Name of the Originating Financial Depository Institution.
	Company Name	Char(23)	N/A	The name of the company name.
ACH CCD Batch Header	Ref Code	Char (8)	N/A	Reference code.
	Section No.	Number(4)	5225	Constant number.
	Company Name	Char(16)	N/A	The name of the company.
	Comp Disc Data	Char(20)	NULL	Any kind of data specific to the company.
	Comp Id	Char(10)	N/A	Alphanumeric code to identify the company.
	CCD Header Id	Char(3)	CCD	Constant value.
	Comp Entry Desc	Char(10)	CONSOL	A short description from the Originator about the purpose of the entry.
	Tomorrow	Char(6)	N/A	Vdate+1 in YYMMDD format.

Table 6-23 (Cont.) Output File

Record Name	Field Name	Field Type	Default Value	Description
ACH CBR Batch Header	Tomorrow	Char(6)	N/A	Vdate+1 in YYMMDD format.
	Settle Date	Char(3)	NULL	This is inserted by receiving the ACH Operator.
	Reserved	Number(1)	1	Constant number.
	Odfi Id	Number(8)		8-digit routing number of the ODFI.
	Batch No	Number(7)		Batch number.
	Section No.	Number(4)	5225	Constant number.
	Company Name	Char(16)	N/A	The name of the company.
	Reserved	Char(3)	FV1	Constant value.
	Exch Rate	Number(15)		Exchange rate for the specified currency.
	Reserved	Char(2)	US	Constant value.
	Comp Id	Char(10)		Alphanumeric code to identify the company
	CBR Header Id	Char(3)	CBR	Constant value.
	Comp Entry Desc	Char(10)	"CONSOL "	A short description from the Originator about the purpose of the entry.
	Partner Curr Code	Char(3)	N/A	Code identifying the currency the partner uses for business transactions.
ACH CCD Entry	Reserved	Char(3)	USD	Constant value.
	Tomorrow	Char(6)	N/A	Vdate+1 in YYMMDD forma.
	Settle Date	Char(3)	NULL	This is inserted by the receiving ACH Operator.
	Reserved	Number(1)	1	Constant number.
	Odfi Id	Number(8)	N/A	8-digit routing number of the ODFI.
	Batch No	Number(7)	N/A	Batch number.
	Section No.	Number(1)	6	Constant number.
	Trans Code	Char(2)		Code used to identify the type of debit and credit. Value accepted are 27 and 37.
	Routing No	Number(9)		Routing number for the bank account.
	Acct No	Char(17)		Account number of the bank.
	Deposit	Number(10)		The amount involved in the transaction* 10000 (4 implied decimal places).

Table 6-23 (Cont.) Output File

Record Name	Field Name	Field Type	Default Value	Description
	Id	Char(15)	Null	Identification number. Optional field containing a number used by the Originator to insert its own number for tracing purposes.
	Store Name	Char(22)		Name of the local store.
	Disc Data	Char(2)	Null	Discretionary data. Any kind of data specific to the transaction.
	Reserved	Number(1)	0	Constant number.
	Trace No	Number(15)		Used to uniquely identify each entry within a batch. The first 8 digits contain the routing number of the ODFI and the other 7 contains a sequence number.
ACH CBR Entry	Section No.	Number(1)	6	Constant number.
	Trans Code	Char(2)	N/A	Code used to identify the type of debit and credit. Values accepted are 27 and 37.
	Routing No	Number(9)	N/A	Routing number for the bank account.
	Acct No	Char(17)	N/A	Account number of the bank
	Deposit	Number(10)	N/A	The amount involved in the transaction* 10000 (4 implied decimal places).
	Id	Char(15)	NULL	Identification number. Optional field containing a number used by the Originator to insert its own number for tracing purposes.
	Store Name	Char(22)	N/A	Name of the local store.
	Disc Data	Char(2)	NULL	Discretionary data. Any kind of data specific to the transaction.
	Reserved	Number(1)	1	Constant number.
	Trace No	Number(15)	N/A	Used to uniquely identify each entry within a batch. The first 8 digits contain the routing number of the ODFI and the other 7 contains a sequence number.
ACH CBR Addendum	Section No.	Number(3)	701	Constant number.
	Payment Info	Char(80)	Null	Payment related information.
	Reserved	Number(4)	0001	Constant number

Table 6-23 (Cont.) Output File

Record Name	Field Name	Field Type	Default Value	Description
	Trace Seq No	Number(7)	N/A	Sequence number part of the Trace Number of the entry record to which this addendum is referring.
ACH Batch Control	Section No.	Number(4)	8225	Constant number.
	Batch Line Count	Number(6)	N/A	The number of entries and addenda in the batch.
	Hash Count	Number(10)	N/A	Sum of the RDFI IDs in the detail records.
	Total Batch Debit	Number(12)	N/A	Contains the accumulated debit and debit for the file * 10000 (4 implied decimal places).
	Total Batch Credit	Number(12)	N/A	Contains the accumulated credit and credit for the file * 10000 (4 implied decimal places).
	Comp Id	Char(10)	N/A	An alphanumeric code identifying the company.
	Auth	Char(19)	Null	Message Authentication Code. The first 8 characters represent a code from the Data Encryption Standard (DES) algorithm. The remaining eleven characters are blanks.
	Reserved	Char(6)	Null	Reserved.
	ODFI Id	Number(8)	N/A	8-digit routing number of the ODFI.
ACH File Control	Batch No	Number(7)	N/A	Batch number.
	Section No.	Number(1)	9	Constant number.
	Batch count	Number(6)	N/A	The number of batches sent in the file.
	Block count	Number(6)	N/A	The number of physical blocks in the file, including both File Header and File Control Records. This is the ceiling of the number of records divided by the blocking factor, which is 10.
	Entry count	Number(8)	N/A	The number of entries and addenda in the file.
	Total hash count	Number(10)	N/A	Sum of the Entry Hash fields on the Batch Control Records.

Table 6-23 (Cont.) Output File

Record Name	Field Name	Field Type	Default Value	Description
	Total file debit	Number(12)	N/A	Contains the accumulated debit and debit for the file * 10000 (4 implied decimal places).
	Total file credit,	Number(12)	N/A	Contains the accumulated credit and credit for the file * 10000 (4 implied decimal places).
	Reserved	Char(39)	NULL	Reserved.
ACH Completed Block	End string	Char(94)	N/A	Mark the end of the file: a string of 94 '9' characters. The number of end lines with a string of 94 '9' characters is identified by the following equation: $10 - \text{mod}(\text{number of lines in the file}, 10)$.

Design Assumptions

N/A

Download of Escheated Vouchers from Sales Audit for Payment (saescheat)

Module Name	saescheat.pc
Description	Download of Escheated Vouchers from Sales Audit for Payment
Functional Area	Oracle Retail Sales Audit
Module Type	Integration
Module Technology	ProC
Catalog ID	RSA05
Wrapper Script	rmswrap.ksh

Schedule

Oracle Retail Merchandising Batch Schedule

Design Overview

The laws of individual states and countries may require a retailer to return monies for aged, unclaimed gift certificates, and vouchers. This process is called escheatment. This program writes records for this data to tables that are read into Invoice Matching by the program saexpim.pc. The data can then be sent as invoices approved for payment to a financial application.

The saescheat batch program will set the status of vouchers that have met certain state's escheats rules or have expired to the proper status and produce a total for later

export to Invoice Matching. The rules for escheatment are defined on the sa_escheatment_options table.

Restart/Recovery

The logical unit of work is a store/day. The program commits when the number of store/day records processed has reached the commit_max_ctr.

I/O Specification

Integration Type	Download from Sales Audit
File Name	N/A
Integration Contract	IntCon000039

Design Assumptions

N/A

Export DSD and Escheatment from Sales Audit to Invoice Matching (saexpim)

Module Name	saexpim.pc
Description	Export DSD and Escheatment from Sales Audit to Invoice Matching
Functional Area	Oracle Retail Sales Audit
Module Type	Integration
Module Technology	ProC
Catalog ID	RSA04
Wrapper Script	rmswrap.ksh

Schedule

Oracle Retail Merchandising Batch Schedule

Design Overview

The purpose of this program is to support interfacing invoices from Direct Store Delivery and Escheatment sales audit transactions to the Invoice Matching application. Direct Store Delivery invoices refer to products or services that are delivered to the store and paid for at the store. This program will take DSD invoices that have been staged to the transaction header table by the saimptlog.pc program and move them into the invoice header table. All DSD transactions will be assumed paid. They can be assumed received if there is a proof of delivery number listed on them. Transactions with a vendor invoice ID or a proof of delivery number should be matched to any existing invoice in the invoice header, and that invoice updated with the new information being interfaced. Invoices that do not match an existing invoice in the invoice head table will need to be inserted. Each transaction will be exported to the invoice head table only once.

The Sales Audit Transaction type used to identify invoices for Direct Store Delivery transactions will be "Paid Out". The Paid Out transaction has a code of 'PAIDOU'. The Sales Audit sub-transaction types will be used to identify whether the invoice is an "Expense Vendor Payout" or a "Merchandise Vendor Payout". The codes are 'EV' for Expense Vendor Payout and 'MV' for Merchandise Vendor Payout. Any Paid Out transaction with a sub transaction

type of Expense Vendor will create a non-merchandise invoice and cause a record to be written to the invoice non-merchandise table. Sales Audit will store non-merchandise codes in the reason_code field on sa_tran_head. Valid values for these reason codes should correspond to the codes stored on the non_merch_code_head table.

In addition to DSD invoices, this program will also interface Escheatment totals to Invoice Matching. Escheatment is the process where an unredeemed gift certificate/ voucher or credit voucher will, after a set period of time, be paid out as income to the issuing retailer, or in some states, the state receives this escheatment income. Sales Audit will be the governing system that determines who receives this income, but Invoice Matching will send the totals, with the related Partner, to an Accounts Payable system. Escheatment information will be stored on the Sales Audit SA_TOTALS table and will be used to create non-merchandise invoices in Invoice Matching. These invoices will be assumed not paid.

Restart/Recovery

The logical unit of work for this module is defined as a unique store/day combination. Records will be fetched, updated, and inserted based on the commit_max_ctr specified on the restart_control table. Only two commits will be done, one to establish the store/day lock and another at the end, to release the lock after a store/day has been completely processed. In case of failure, all work done will be rolled back to the point right after the call to get_lock and releases the lock. Thus, the rollback segment should be large enough to hold all inserts into sa_exported for one store_day.

Integration Contract

Integration Type	Download from Sales Audit
File Name	N/A
Integration Contract	IntCon00004 INVC_HEAD table

Design Assumptions

N/A

Export from Sales Audit to Oracle Retail Analytics (saexpdw)

Module Name	saexpdw.pc
Description	Export from Sales Audit to Oracle Retail Analytics
Functional Area	Oracle Retail Sales Audit
Module Type	Integration
Module Technology	ProC
Catalog ID	RSA02
Wrapper Script	batch_resa2dw.ksh

Schedule

Oracle Retail Merchandising Batch Schedule

Design Overview

The purpose of this batch module is to fetch all sales and return transactions that do not have Retail Analytics errors from the Sales Audit database tables for transmission to the Oracle Retail Analytics application. The data will be sent at the store day level. If the transaction has a status of Deleted, and if it has been previously Transmitted, a reversal of the transaction will be sent.



Note:

This batch program can be run in two modes - trickle mode and batch mode. If 'Y' is passed as a parameter while running the batch program, then the batch runs in trickle mode. If 'N' or no parameter is passed, it runs in normal batch mode.

Restart/Recovery

The logical unit of work for this module is defined as a unique store/day combination. Records will be fetched, updated, and inserted based on the commit_max_ctr. Only two commits will be done: one to establish the store/day lock and another at the end, to release the lock after a store/day has been completely processed. The RDWT, RDWF, RDWS, and RDWC formatted output files will be created with temporary names and renamed just before the end of store/day commit.

In case of a failure, all the work done will be rolled back to the point right after the call to get_lock() and the lock is released. Thus, the rollback segment should be large enough to hold all inserts into sa_exported for one store/day.

I/O Specification

Integration Type	Download from Sales Audit
File Name	RDWT_ appended with store number, business date, and system date. RDWF_ appended with store number, business date, and system date. RDWS_ appended with store number, business date, and system date. RDWC_ appended with store number, business date, and system date.
Integration Contract	IntCon000041 (RDWT) IntCon000156 (RDWF) IntCon000157 (RDWS) IntCon000158 (RDWC)

Four output files will be created for each store_day:

- RDWT - Transaction File
- RDWF - Form of Payment (Tender) file
- RDWS - Store Totals output file
- RDWC - Cashier output File

Each output file is converted into a format for loading into Retail Analytics by the resa2dw Perl script.

Sales Audit - File Layout - Retail Analytics

- File layouts for the interface between sales audit and Retail Analytics.
- Char fields are left justified and blank filled.
- Number fields are right justified and zero filled. They can contain only numbers.
- Numeric fields are left justified and blank filled. They can contain only numbers.

RDWT File

Record Name	Field Name	Field Type	Default Value	Description	Required
File Header	File Type Record Descriptor	Char(5)	FHEAD	Identifies file record type	
	File Line Identifier	Number(10)	specified by external system	ID of current line being processed by input file.	Yes
	File Type Definition	Char(4)	RDWT	Identifies file as 'Retail Analytics Transaction file'	Yes
	File Create Date	Number(14)	create date	Date file was written by external system. Format YYYYMMDDHH24MISS	Yes
Transaction Header	File Type Record Descriptor	Char(5)	THEAD	Identifies transaction record type	
	File Line Identifier	Number(10)	specified by external system	ID of current line being processed by input file.	Yes
	Business date	Number(8)		Format YYYYMMDD (Note, This is the date the Retail Analytics will consider the transaction date)	Yes
	Transaction Date	Number(14)	transaction date	Date sale/return transaction was processed at the POS. Format YYYYMMDDHH24MISS (Note, the Retail Analytics only uses the HH24MI part of this date)	Yes

Record Name	Field Name	Field Type	Default Value	Description	Required
	Location	Number(10)	specified by external system	Store or warehouse identifier. This value is now being determined based on either the Account for Sale or Account for Return system option.	Yes
	Register ID	Char(5)		The register identifier	Yes, -1 for null
	Banner ID	Char(4)		The unique identifier of the banner.	Yes, -1 for null
	Line Media ID	Char(10)		The identifier of the media for the order line. For non-merchandise items, such as Shipping & Handling, Service Lines and gift certificates, the media code will be that of the order line it's associated.	Yes, -1 for null
	Selling Item ID	Char(25)		The unique identifier of a selling item.	Yes, -1 for null
	Customer Order Header ID	Char(48)		The unique identifier of a customer order.	Yes, -1 for null
	Customer Order Line ID	Char(30)		The identifier of a customer order line. For a Value Added Service, like monogramming, this will be the line number for the item which the service was applied.	Yes, -1 for null
	Customer Order Create Date	Char(8)		The date when the customer order was created/placed.	Yes, -1 for null

Record Name	Field Name	Field Type	Default Value	Description	Required
	Cashier Identifier	Char(10)		The cashier number. This will be the unique employee number.	Yes, -1 for null
	Salesperson Identifier	Char(10)		The salesperson number. This will be the unique employee number.	Yes, -1 for null
	Customer ID Type	Char(6)		The type of ID number used by this customer.	Yes, -1 for null
	Customer ID Number	Char(16)		Customer id associated with the transaction.	Yes, -1 for null
	Transaction Number	Number(10)		The unique transaction reference number generated by the POS.	Yes
	Original Register ID	Char(5)		Register ID of the original transaction.	Yes for a transaction type of 'PVOID'.
	Original Transaction Number	Number(10)		Transaction number of the original transaction.	Yes for a transaction type of 'PVOID', 'EEXCH' and 'RETURN'
	Transaction Header Number	Numeric(20)		Unique reference used within sales audit to represent the date/store/register/tran_no	Yes
	Revision number	Number(3)		Number used to identify the version of the transaction being sent.	Yes
	Sales Sign	Char(1)	'P'- positive 'N' – negative	Determines if the Total Sales Quantity and Total Sales Value are positive or negative.	Yes
	Transaction Type	Char(6)		Transaction type code	Yes
	Sub Transaction Type	Char(6)		The Sub Transaction type	Yes, -1 for null

Record Name	Field Name	Field Type	Default Value	Description	Required
	Retail Type	Char(1)	'R'egular, 'P'romo, or 'C'learance		Yes
	Item_Seq_No	Number(4)		The order in which items were entered during the transaction.	No
	Employee Number (Cashier)	Char(10)		Employee identification number. This will only be populated if the sub transaction type is 'EMP'.	Yes, -1 for null
	Receipt Indicator	Char(1)		Flag that identifies returns that have been processed without a receipt. This field will only be populated if the transaction type is 'RETURN'.	No
	Reason Code	Char(6)		A reason is required with a Paid In/Out transaction type, and optional with a return transaction.	Yes, -1 for null
	Vendor number	Number(10)		This will only get populated when the paid in code is Expense Vendor	No
	Item Type	Char(6)	item type identifier	Type of item sold, 'ITEM', 'REF', 'GCN' (gift certificate number), or 'NMITEM'	No
	Item	Char(25)		ID number of the item or gift certificate.	No. Required if Item Type is not null.
	Ref Item	Char(25)		Sub-transaction level item	No. Also, this field can never be populated without a transaction level item in the item field.
	Taxable Indicator	Char(1)		Taxable/non-taxable status indicator	No

Record Name	Field Name	Field Type	Default Value	Description	Required
	Entry/mode	Char(6)		Indicator that identifies whether the item was scanned or manually entered	No
	Department	Number(4)		Department of item sold or returned. Yes need to validate if using ReSA.	No
	Class	Number(4)		Class of item sold or returned. Yes need to validate if using ReSA.	No
	Subclass	Number(4)		Subclass of item sold or returned. Yes need to validate if using ReSA.	No
	Total Sales Quantity	Number(12)		Number of units sold at a particular location with 4 implied decimal places.	No
	Total Transaction Value	Number(20)		Sales value, net sales value of goods sold/ returned with 4 implied decimal places.	No
	Override Reason	Char(6)		This column will be populated when an item's price has been overridden at the POS to define why it was overridden. This will also always be sent if the transaction originated in RCOM.	Yes, -1 for null
	Return Reason	Char(6)		The reason an item was returned.	Yes, -1 for null
	Total original sign	Char(1)	'P'- positive 'N' – negative		No

Record Name	Field Name	Field Type	Default Value	Description	Required
	Total Original Sales Value	Number(20)		This column will be populated when the item's price was overridden at the POS and the item's original unit retail is known. This will always be written when the transaction originated in RCOM. This has 4 implied decimals.	No
	Weather	Char(6)		For transaction types of 'COND', this field will store the type of weather for the store-day.	No
	Temperature	Char(6)		For transaction types of 'COND', this field will store the type of temperature for the store-day.	No
	Traffic	Char(6)		For transaction types of 'COND', this field will store the type of traffic for the store-day.	No
	Construction	Char(6)		For transaction types of 'COND', this field will store info regarding any construction on that store-day.	No
	Drop Shipment Indicator	Char(1)	'Y' or 'N'	Indicates whether item is involved in a drop shipment.	No
	Item Status	Char(6)		The status of the item, required for voided or exchanged items. Valid values are found in the code_detail table under code_type SASI.	Y, -1 for null

Record Name	Field Name	Field Type	Default Value	Description	Required
	Tran Process Sys	Char(3)		This column holds the name of the system that processed the transaction. This will be used for filtering duplicate transactions coming from the different systems for export to downstream systems. Expected values are POS – Point of Sale, OMS – Order Management System and SIM – Store Inventory Management.	Y, -1 for null
	Return Wh	Number(10)		This column contains the physical warehouse ID for the warehouse identifier where the item was returned.	N, -1 for null
	Fulfill Order No	Char(48)		This column holds the number from OMS related to the fulfillment details. One or more fulfillment orders could relate back to a single customer order in OMS. This column is required if the order is a cross channel order (i.e. Sales Type = 'E') and the item status is 'ORD'.	N, -1 for null

Record Name	Field Name	Field Type	Default Value	Description	Required
	No Inventory Return Ind	Char(1)		This column contains an indicator that identifies a return without inventory. This is generally a non-required column, but in case of Returns, this is required.	N
	Sales Type	Char(1)		This column indicates whether the line item is a Regular Sale, a customer order serviced by OMS (External CO) or a customer order serviced by a store (In Store CO).	Y
	Return Disposition	Char(10)		This column will contain the disposition code published by RWMS as part of the Returns upload to OMS.	N, -1 for null
	Original Store	Char(10)		This column contains the store ID for the original store.	N
	Original Transaction Number	Number(10))	Original transaction number for the returned item.	N
	Reference Number 1	Char(30)			No
	Reference Number 2	Char(30)			No
	Reference Number 3	Char(30)			No
	Reference Number 4	Char(30)			No
	Reference Number 5	Char(30)			No
	Reference Number 6	Char(30)			No
	Reference Number 7	Char(30)			No

Record Name	Field Name	Field Type	Default Value	Description	Required
	Reference Number 8	Char(30)			No
	Reference Number 25	Char(30)			No
	Reference Number 26	Char(30)			No
	Reference Number 27	Char(30)			No
	Reference Number 28	Char(30)			No
	Reference Number 29	Char(30)			No
	Reference Number 30	Char(30)			No
	Reference Number 31	Char(30)			No
	Fulfill Loc Type	Char(2)		This column contains the fulfillment location type of the customer order. Valid values are 'S' for physical store and 'V' for virtual store.	N, -1 for null
	Fulfill Loc ID	Number(10)		This column contains the fulfillment location of the customer order. It can only be either a physical store or a virtual store.	N, -1 for null
	Posting Store	Number(10)		This column contains the store at which the item sale/return should be accounted for in case of cross-store sales happening at co-located stores. It is expected that this field will be populated only for items that are checked out at a different store from the one at which they are originally managed.	N, -1 for null

Record Name	Field Name	Field Type	Default Value	Description	Required
Transaction Detail	File Type Record Descriptor	Char(5)	TDETL	Identifies transaction record type	
	File Line Identifier	Number(10)	specified by external system	ID of current line being processed by input file.	Yes
	Discount Type	Char(6)		Code for discount type from code_detail, code_type = 'SADT'	No
	Promotional Transaction Type	Char(6)		Code for promotional type from code_detail, code_type = 'PRMT'	Yes
	Promotion Number	Number(10)	promotion number	Promotion number from Merchandising	No
	Promotion Component Number	Number(10)	Offer ID from Pricing		Required if it is a promotional sale.
	Coupon Number	Char(40)			Yes if Discount Type is 'SCOUP'.
	Coupon Reference Number	Char(16)			No
	Sales Quantity	Number(12)		Number of units sold in this prom type with 4 implied decimal places.	No
	Transaction Sign	Char(1)	'P'- positive 'N' – negative		Yes
	Transaction Value	Number(20)		Value of units sold in this promotion type with 4 implied decimal places.	Yes
	Discount Value	Number(20)		Value of discount given in this prom type with 4 implied decimal places.	Yes
	Reference Number 13	Char(30)			No
Reference Number 14	Char(30)			No	

Record Name	Field Name	Field Type	Default Value	Description	Required
	Reference Number 15	Char(30)			No
	Reference Number 16	Char(30)			No
Transaction Trailer	File Type Record Descriptor	Char(5)	TTAIL	Identifies file record type	
	File Line Identifier	Number(10)	specified by external system	ID of current line being processed by input file.	Yes
	Transaction Count	Number(6)	specified by external system	Number of TDETL records in this transaction set	Yes
File Trailer	File Type Record Descriptor	Char(5)	FTAIL	Identifies file record type	
	File Line Identifier	Number(10)	specified by external system	ID of current line being processed by input file.	Yes
	File Record Counter	Number(10)		Number of records/ transactions processed in current file (only records between head & tail)	Yes

Transaction Item Information Produced by saexpdw.pc after Translation by resa2dw

Record Name	Field Name	Field Type	Default Value	Description	Required
	Business date	Number(8)		Format YYYYMMDD	Yes
	Transaction Date	Number(14)	transaction date	Date sale/return transaction was processed at the POS. Format YYYYMMDDHH24MISS	Yes
	Location	Number(10)	specified by external system	Store or warehouse identifier. This value is now being determined based on either the Account for Sale or Account for Return system option.	Yes

Record Name	Field Name	Field Type	Default Value	Description	Required
	Register ID	Char(5)		The register identifier	Yes, -1 for null
	Banner ID	Char(4)		The unique identifier of the banner.	Yes, -1 for null
	Line Media ID	Char(10)		The identifier of the order line media. For non-merchandise items, such as Shipping & Handling, Service Lines and gift certificates, the media code will be that of the order line it's associated.	Yes, -1 for null
	Selling Item ID	Char(25)		The unique identifier of a selling item.	Yes, -1 for null
	Customer Order Header ID	Char(48)		The unique identifier of a customer order.	Yes, -1 for null
	Customer Order Line ID	Char(30)		The identifier of a customer order line. For a Value Added Service, like monogramming, this will be the line number for the item, which the service was applied.	Yes, -1 for null
	Customer Order Create Date	Number(8)		The customer order creation date	Yes, 'transaction date' for null
	Cashier Identifier	Char(10)		The cashier number. This will be the unique employee number.	Yes, -1 for null
	Salesperson Identifier	Char(10)		The salesperson number. This will be the unique employee number.	Yes, -1 for null
	Customer ID Type	Char(6)		The type of ID number used by this customer.	Yes, -1 for null
	Customer ID Number	Char(16)		Customer id associated with the transaction.	Yes, -1 for null

Record Name	Field Name	Field Type	Default Value	Description	Required
	Transaction Number	Number(10)		The unique transaction reference number generated by the POS.	Yes
	Original Register ID	Char(5)		Register ID of the original transaction.	Yes for a transaction type of 'PVOID'.
	Original Transaction Number	Number(10)		Transaction number of the original transaction.	Yes for a transaction type of 'PVOID'.
	Transaction Header Number	Numeric(20)		Unique reference used within sales audit to represent the date/store/register/tran_no	Yes
	Revision number	Number(3)		Number used to identify the version of the transaction being sent.	Yes
	Sales Sign	Char(1)	'P'- positive 'N' – negative	Determines if the Total Sales Quantity and Total Sales Value are positive or negative.	Yes
	Transaction Type	Char(6)		Transaction type code	Yes
	Sub Transaction Type	Char(6)		The Sub Transaction type	Yes, -1 for null
	Retail Type	Char(1)	'R'egular, 'P'romo, or 'C'learance		Yes
	Item_Seq_No	Number(4)		The order in which items were entered during the transaction.	No
	Employee Number (Cashier)	Char(10)		Employee identification number. This will only be populated if the sub transaction type is 'EMP'.	Yes, -1 for null

Record Name	Field Name	Field Type	Default Value	Description	Required
	Receipt Indicator	Char(1)		Flag that identifies returns that have been processed without a receipt. This field will only be populated if the transaction type is 'RETURN'.	No
	Reason Code	Char(6)		A reason is required with a Paid In/Out transaction type, and optional with a return transaction.	Yes, -1 for null
	Vendor number	Numeric(10)		This will only get populated when the paid in code is Expense Vendor	No
	Item Type	Char(6)	item type identifier	Type of item sold, 'ITEM', 'REF', 'GCN' (gift certificate number), or 'NMITEM'	No
	Item	Char(25)		ID number of the item or gift certificate.	No. Required if Item Type is not null.
	Ref Item	Char(25)		Sub-transaction level item	No. Also, this field can never be populated without a transaction level item in the item field.
	Taxable Indicator	Char(1)		Taxable/non-taxable status indicator	No
	Entry/mode	Char(6)		Indicator that identifies whether the item was scanned or manually entered	No
	Department	Number(4)		Department of item sold or returned. Yes need to validate if using ReSA.	No
	Class	Number(4)		Class of item sold or returned. Yes need to validate if using ReSA.	No

Record Name	Field Name	Field Type	Default Value	Description	Required
	Subclass	Number(4)		Subclass of item sold or returned. Yes need to validate if using ReSA.	No
	Total Sales Quantity	Number(12)		Number of units sold at a particular location with 4 implied decimal places.	No
	Total Transaction Value	Number(20)		Sales value, net sales value of goods sold/ returned with 4 implied decimal places.	No
	Override Reason	Char(6)		This column will be populated when an item price has been overridden at the POS to define why it was overridden. This will always be sent if the transaction originated in RCOM.	Yes, -1 for null
	Return Reason	Char(6)		The reason an item was returned.	Yes, -1 for null
	Total original sign	Char(1)	'P'- positive 'N' – negative		No
	Total Original Sales Value	Number(20)		This column will be populated when the item's price was overridden at the POS and the item's original unit retail is known. This will always be sent if the transaction originated in RCOM. This has 4 implied decimals.	No
	Weather	Char(6)		For transaction types of 'COND', this field will store the type of weather for the store-day.	No

Record Name	Field Name	Field Type	Default Value	Description	Required
	Temperature	Char(6)		For transaction types of 'COND', this field will store the type of temperature for the store-day.	No
	Traffic	Char(6)		For transaction types of 'COND', this field will store the type of traffic for the store-day.	No
	Construction	Char(6)		For transaction types of 'COND', this field will store info regarding any construction on that store-day.	No
	Drop Shipment Indicator	Char(1)	'Y' or 'N'	Indicates whether item is involved in a drop shipment.	No
	Item Status	Char(6)		The status of the item, required for voided or exchanged items. Valid values are found in the code_detail table under code_type SASI.	Y, -1 for null
	Tran Process Sys	Char(3)		This column holds the name of the system that processed the transaction. This will be used for filtering duplicate transactions coming from the different systems for export to downstream systems. Expected values are POS – Point of Sale, OMS – Order Management System and SIM – Store Inventory Management.	Y, -1 for null

Record Name	Field Name	Field Type	Default Value	Description	Required
	Return Wh	Number(10)		This column contains the physical warehouse ID for the warehouse identifier where the item was returned.	N, -1 for null
	Fulfill Order No	Char(48)		This column holds the number from OMS related to the fulfillment details. One or more fulfillment orders could relate back to a single customer order in OMS. This column is required if the order is a cross channel order (i.e. Sales Type = 'E') and the item status is 'ORD'.	N, -1 for null
	No Inventory Return Ind	Char(1)		This column contains an indicator that identifies a return without inventory. This is generally a non-required column, but in case of Returns, this is required.	N
	Sales Type	Char(1)		This column indicates whether the line item is a Regular Sale, a customer order serviced by OMS (External CO) or a customer order serviced by a store (In Store CO).	Y
	Return Disposition	Char(10)		This column will contain the disposition code published by RWMS as part of the Returns upload to OMS.	N, -1 for null

Record Name	Field Name	Field Type	Default Value	Description	Required
	Original Store	Char(10)		This column contains the store ID for the original store.	No
	Original Transaction Number	Number(10)		Original transaction number for the returned item.	No
	Discount Type	Char(6)		Code for discount type from code_detail, code_type = 'SADT'	No
	Promotional Transaction Type	Char(6)		Code for promotional type from code_detail, code_type = 'PRMT'	Yes
	Promotion Number	Number(10)	promotion number	Promotion number from Merchandising	No
	Promotion Component Number	Numbr(10)	Offer ID from Pricing		Required if it is a promotional sale.
	Coupon Number	Char(40)			Yes if Discount Type is 'SCOUP'.
	Coupon Reference Number	Char(16)			No
	Sales Quantity	Number(12)		Number of units sold in this prom type with 4 implied decimal places.	No
	Transaction Sign	Char(1)	'P'- positive 'N' – negative		Yes
	Transaction Value	Number(20)		Value of units sold in this promotion type with 4 implied decimal places.	Yes
	Discount Value	Number(20)		Value of discount given in this prom type with 4 implied decimal places.	Yes
	Reference Number 1	Char(30)			No
	Reference Number 2	Char(30)			No

Record Name	Field Name	Field Type	Default Value	Description	Required
	Reference Number 3	Char(30)			No
	Reference Number 4	Char(30)			No
	Reference Number 5	Char(30)			No
	Reference Number 6	Char(30)			No
	Reference Number 7	Char(30)			No
	Reference Number 8	Char(30)			No
	Reference Number 13	Char(30)			No
	Reference Number 14	Char(30)			No
	Reference Number 15	Char(30)			No
	Reference Number 16	Char(30)			No
	Reference Number 25	Char(30)			No
	Reference Number 26	Char(30)			No
	Reference Number 27	Char(30)			No
	Reference Number 28	Char(30)			No
	Reference Number 29	Char(30)			No
	Reference Number 30	Char(30)			No
	Reference Number 31	Char(30)			No
	Fulfill Loc Type	Char(2)		This column contains the fulfillment location type of the customer order. Valid values are 'S' for physical store and 'V' for virtual store.	N, -1 for null

Record Name	Field Name	Field Type	Default Value	Description	Required
	Fulfill Loc ID	Number(10)		This column contains the fulfillment location of the customer order. It can only be either a physical store or a virtual store.	N, -1 for null
	Posting Store	Number(10)		This column contains the store at which the item sale/return should be accounted for in case of cross-store sales happening at co-located stores. It is expected that this field will be populated only for items that are checked out at a different store from the one at which they are originally managed.	N, -1 for null

RDWF File

Record Name	Field Name	Field Type	Default Value	Description	Required
File Header	File Type Record Descriptor	Char(5)	FHEAD	Identifies file record type	
	File Line Identifier	Number(10)	specified by external system	ID of current line being processed by input file.	Yes
	File Type Definition	Char(4)	RDWF	Identifies file as 'Retail Analytics Form of Payment (Tender) file'	Yes
	File Create Date	Number(14)	create date	Date file was written by external system. Format YYYYMMDDHH24MISS	Yes
File Detail	File Type Record Descriptor	Char(5)	FDETL	Identifies file record type	

Record Name	Field Name	Field Type	Default Value	Description	Required
	File Line Identifier	Number(10)	specified by external system	ID of current line being processed by input file.	Yes
	Business date	Number(8)		Format YYYYMMDD	Yes
	Transaction Date	Number(14)	transaction date	Date sale/return transaction was processed at the POS. Format YYYYMMDDHH24MISS	Yes
	Location	Number(10)	specified by external system	Store or warehouse identifier.	Yes
	Cashier Identifier	Char(10)		The cashier number. This will be the unique employee number.	Yes, -1 for null
	Register Identifier	Char(5)			Yes, -1 for null
	Sales Sign	Char(1)	'P'- positive 'N' – negative	Determines if the Total Sales Quantity and Total Sales Value are positive or negative.	Yes
	Transaction Sequence Number	Numeric(20)		Unique reference used within sales audit to represent the date/store/register/transaction number	Yes
	Revision number	Number(3)		Number used to identify the version of the transaction being sent.	Yes
	Transaction Type	Char(6)		Transaction type code.	Yes
	Tender type group	Char(6)			Yes
	Tender type id	Number(6)		Tender type code.	Yes
	Tender amount	Number(20)		Tender amount.	Yes

Record Name	Field Name	Field Type	Default Value	Description	Required
	Credit Card Entry Mode	Char(6)		Contains the method in which the transaction was entered at the POS. Possible entry modes could include: Terminal Used, Magnetic Strip Track One Read, Magnetic Strip Two Read, Magnetic Strip One Transmitted, or Magnetic Strip Two Transmitted. The code type for this field is 'CCEM'.	No
	Voucher Number	Char(25)			No
	Voucher Age	Numeric(5)		Age of the gift certificate. Redeemed date minus sold date.	Yes if Tender Type Group is 'VOUCH'.
	Escheat Date	Numeric(8)		Date on which this gift certificate escheats. Format is YYYYMMDD.	Yes if voucher can escheat.
	Coupon Number	Char(40)			Yes if Tender Type Group is 'COUPON'.
	Coupon Reference Number	Char(16)			No. Only if Tender Type Group is 'COUPON'.
	Transaction Status	Char(1)		Determines if the transaction is Present ('P') or Voided/Deleted ('R' - Reverse)	NO
	Reference Number 9	Char(30)			No
	Reference Number 10	Char(30)			No
	Reference Number 11	Char(30)			No
	Reference Number 12	Char(30)			No
File Trailer	File Type Record Descriptor	Char(5)	FTAIL	Identifies file record type	

Record Name	Field Name	Field Type	Default Value	Description	Required
	File Line Identifier	Number(10)	specified by external system	ID of current line being processed by input file.	Yes
	File Record Counter	Number(10)		Number of records/ transactions processed in current file (only records between head & tail)	Yes

Retail Analytics Form of Payment File after Translation by resa2dw

Record Name	Field Name	Field Type	Default Value	Description	Required
	Business date	Number(8)		Format YYYYMMDD	Yes
	Transaction Date	Number(14)	transaction date	Date sale/return transaction was processed at the POS. Format YYYYMMDDHH24MISS	Yes
	Location	Number(10)	specified by external system	Store or warehouse identifier.	Yes
	Cashier Identifier	Char(10)		The cashier number. This will be the unique employee number.	Yes, -1 for null
	Register Identifier	Char(5)			Yes, -1 for null
	Sales Sign	Char(1)	'P'- positive 'N' – negative	Determines if the Total Sales Quantity and Total Sales Value are positive or negative.	Yes
	Transaction Sequence Number	Numeric(20))	Unique reference used within sales audit to represent the date/store/register/ transaction number	Yes

Record Name	Field Name	Field Type	Default Value	Description	Required
	Revision number	Number(3)		Number used to identify the version of the transaction being sent.	Yes
	Transaction Type	Char(6)		Transaction type code.	Yes
	Tender type group	Char(6)			Yes
	Tender type id	Number(6)		Tender type code.	Yes
	Tender amount	Number(20)		Tender amount.	Yes
	Credit Card Entry Mode	Char(6)		Contains the method in which the transaction was entered at the POS. Possible entry modes could include: Terminal Used, Magnetic Strip Track One Read, Magnetic Strip Two Read, Magnetic Strip One Transmitted, or Magnetic Strip Two Transmitted. The code type for this field is 'CCEM'.	No
	Voucher Number	Char(25)			No
	Voucher Age	Number(5)		Age of the gift certificate. Redeemed date minus sold date.	Yes if Tender Type Group is 'VOUCH'.
	Escheat Date	Number(8)		Date on which this gift certificate escheats. Format is YYYYMMDD.	Yes if voucher can escheat.
	Coupon Number	Char(40)			Yes if Tender Type Group is 'COUPON'.

Record Name	Field Name	Field Type	Default Value	Description	Required
	Coupon Reference Number	Char(16)			No. Only if Tender Type Group is 'COUPON'.
	Transaction Status	Char(1)		Determines if the transaction is Present ('P') or Voided/ Deleted ('R' - Reverse)	No
	Reference Number 9	Char(30)			No
	Reference Number 10	Char(30)			No
	Reference Number 11	Char(30)			No
	Reference Number 12	Char(30)			No

RDWS File

Record Name	Field Name	Field Type	Default Value	Description	Required
File Header	File Type Record Descriptor	Char(5)	FHEAD	Identifies file record type	
	File Line Identifier	Number(10)	specified by external system	ID of current line being processed by input file.	Yes
	File Type Definition	Char(4)	RDWS	Identifies file as 'Retail Analytics Store Totals file'	Yes
	File Create Date	Numeric(14)	create date	Date file was written by external system. Format YYYYMMDDHH24MISS	Yes
File Detail	File Type Record Descriptor	Char(5)	FDETL	Identifies transaction record type	
	File Line Identifier	Number(10)	specified by external system	ID of current line being processed by input file.	Yes
	Business date	Number(8)		Format YYYYMMDD	Yes
	Location	Number(10)	specified by external system	Store or warehouse identifier	Yes

Record Name	Field Name	Field Type	Default Value	Description	Required
	Sales Sign	Char(1)	'P'- positive 'N' – negative	Determines if the Total Sales Quantity and Total Sales Value are positive or negative.	Yes
	Total ID	Char(10)		Category identifier used to determine the type of total.	Yes
	Reference Number 1	Char(30)			No
	Reference Number 2	Char(30)			No
	Reference Number 3	Char(30)			No
	Total Sign	Char(1)	'P'- positive 'N' – negative		Yes
	Total Amount	Number(20)		Total over/short amount with 4 implied decimal places.	Yes
File Trailer	File Type Record Descriptor	Char(5)	FTAIL	Identifies file record type	
	File Line Identifier	Number(10)	specified by external system	ID of current line being processed by input file.	Yes
	File Record Counter	Number(10)		Number of records/ transactions processed in current file (only records between head & tail)	Yes

Store Totals Information after Translation by resa2dw

Record Name	Field Name	Field Type	Default Value	Description	Required
	Business date	Number(8)		Format YYYYMMDD	Yes
	Location	Number(10)	specified by external system	Store or warehouse identifier	Yes

Record Name	Field Name	Field Type	Default Value	Description	Required
	Sales Sign	Char(1)	'P'- positive 'N' – negative	Determines if the Total Sales Quantity and Total Sales Value are positive or negative.	Yes
	Total ID	Char(10)		Category identifier used to determine the type of total.	Yes
	Reference Number 1	Char(30)			No
	Reference Number 2	Char(30)			No
	Reference Number 3	Char(30)			No
	Total Sign	Char(1)	'P'- positive 'N' – negative		Yes
	Total Amount	Number(20)		Total over/short amount with 4 implied decimal places.	Yes

RDWC File

Record Name	Field Name	Field Type	Default Value	Description	Required
File Header	File Type Record Descriptor	Char(5)	FHEAD	Identifies file record type	
	File Line Identifier	Number(10)	specified by external system	ID of current line being processed by input file.	Yes
	File Type Definition	Char(4)	RDWC	Identifies file as 'Retail Analytics Cashier/Register Totals file'	Yes
	File Create Date	Number(14)	create date	Date file was written by external system. Format YYYYMMDDHH24MISS	Yes
File Detail	File Type Record Descriptor	Char(5)	FDETL	Identifies transaction record type	
	File Line Identifier	Number(10)	specified by external system	ID of current line being processed by input file.	Yes

Record Name	Field Name	Field Type	Default Value	Description	Required
	Business date	Number(8)		Format YYYYMMDD	Yes
	Location	Number(10)	specified by external system	Store or warehouse identifier	Yes
	Cashier Identifier	Char(10)		The cashier number	If Cashier_id is NULL then Register_id has value. If Cashier_id has value then Register_id is NULL. Yes, -1 for null
	Register ID	Char(5)		The register identifier	If Cashier_id is NULL then Register_id has value. If Cashier_id has value then Register_id is NULL. Yes, -1 for null
	Sales Sign	Char(1)	'P'- positive 'N' – negative	Determines if the Total Sales Quantity and Total Sales Value are positive or negative.	Yes
	Total ID	Char(10)		Category identifier used to determine the type of total.	Yes
	Reference Number 1	Char(30)			No
	Reference Number 2	Char(30)			No
	Reference Number 3	Char(30)			No
	Total Sign	Char(1)	'P'- positive 'N' – negative		Yes
	Total Amount	Number(20)		Total over/short amount with 4 implied decimal places.	Yes
File Trailer	File Type Record Descriptor	Char(5)	FTAIL	Identifies file record type	

Record Name	Field Name	Field Type	Default Value	Description	Required
	File Line Identifier	Number(10)	specified by external system	ID of current line being processed by input file.	Yes
	File Record Counter	Number(10)		Number of records/ transactions processed in current file (only records between head & tail)	Yes

Cashier/ Register Totals Information after Translation by resa2dw

Record Name	Field Name	Field Type	Default Value	Description	Required
	Business date	Number(8)		Format YYYYMMDD	Yes
	Location	Number(10)	specified by external system	Store or warehouse identifier	Yes
	Cashier Identifier	Char(10)		The cashier number	If Cashier_id is NULL then Register_id has value. If Cashier_id has value then Register_id is NULL. Yes, -1 for null
	Register ID	Char(5)		The register identifier	If Cashier_id is NULL then Register_id has value. If Cashier_id has value then Register_id is NULL. Yes, -1 for null
	Sales Sign	Char(1)	'P' - positive 'N' - negative	Determines if the Total Sales Quantity and Total Sales Value are positive or negative.	Yes
	Total ID	Char(10)		Category identifier used to determine the type of total.	Yes
	Reference Number 1	Char(30)			No

Record Name	Field Name	Field Type	Default Value	Description	Required
	Reference Number 2	Char(30)			No
	Reference Number 3	Char(30)			No
	Total Sign	Char(1)	'P'- positive 'N' – negative		Yes
	Total Amount	Number(20)		Total over/short amount with 4 implied decimal places.	Yes

Design Assumptions

N/A

Export Inventory Reservation/Release for In Store Customer Order & Layaway Transactions (saordinvexp)

Module Name	saordinvexp.pc
Description	Export Inventory Reservation/Release for In Store Customer Order & Layaway Transactions from Sales Audit
Functional Area	Sales Audit
Module Type	Integration
Module Technology	ProC
Catalog ID	RSA12
Wrapper Script	rmswrap_multi_dnld_in.ksh

Schedule

Oracle Retail Merchandising Batch Schedule

Design Overview

This batch program will generate a flat file to reserve or un-reserve the inventory for items on in-store customer order or layaway transactions. Inventory will be reserved for items on customer order/layaway initiate and un-reserved for customer order/layaway cancel or complete transactions.

Customer orders can be categorized into two categories: In-Store Customer Orders and External Customer Orders. The In-Store Customer Orders are defined as orders that are serviced at the store and inventory reservation is done in Oracle Retail Store Inventory Management (SIM). While the External Customer orders are serviced by an external order management system, no inventory reservation will be made at the store in SIM.

This batch should only process records where the sales type is not equal to External Customer Sales, as it handles only the in-store type orders.

Restart/Recovery

The logical unit of work for this module is defined as a unique store/day combination.

Records are fetched, updated, and inserted in batches of `pl_commit_max_ctr`. Only two commits are done, one to establish the store/day lock and another at the end, to release the lock after a store/day is completely processed. The ORIN formatted output file is created with a temporary name and renamed just before the end of store/day commit.

In case of failure, all work done is rolled back to the point right after the call to `get_lock()` and the lock is released. Thus, the rollback segment should be large enough to hold all inserts into `sa_exported` for one store/day.

I/O Specification

Integration Type Inventory Export from Sales Audit to Merchandising
File Name ORIN_<store>_<tran_date>_<sysdate>
Integration Contract IntCon000049

Output File Layout

Table 6-24 Output File Layout

Record Name	Field Name	Field Type	Default Value	Description
FHEAD	Record descriptor	Char(5)	FHEAD	Identifies the file record type.
	File Line Id	Char(10)	0000000001	Sequential file line number.
	File type Definition	Char(4)	ORIN	Identifies the file type.
	File Create Date	Char(14)	N/A	File Create Date in YYYYMMDDHHMMSS format.
THEAD	Location	Number(10)	N/A	Store location number.
	Record descriptor	Char(5)	THEAD	Identifies the file record type.
	File Line Id	Char(10)		Sequential file line number.
	Transaction Date & Time	Char(14)	Transaction Date	Date and time of the order processed.
TDETL	Transaction Type	Char(6)	SALE	Transaction type code specifies whether the transaction is sale or return.
	Record descriptor	Char(5)	TDETL	Identifies the file record type.
	File Line Id	Char(10)	N/A	Sequential file line number.
	Item Type	Char(3)	REF or ITM	Can be REF or ITM.

Table 6-24 (Cont.) Output File Layout

Record Name	Field Name	Field Type	Default Value	Description
	Item	Char(25)	N/A	ID number of the ITM or REF.
	Item Status	Char(6)	LIN - Layaway Initiate LCA - Layaway Cancel LCO - Layaway Complete PVLCO - Post void of Layaway complete ORI - Pickup/delivery Initiate ORC - Pickup/delivery Cancel ORD - Pickup/delivery Complete PVORD - Post void of Pick-up/delivery complete	Type of transaction.
	Dept	Number(4)	N/A	Department of item sold or returned.
	Class	Number(4)	N/A	Class of item sold or returned
	Sub class	Number(4)	N/A	Subclass of item sold or returned.
	Pack Ind	Char(1)	N/A	Pack indicator of item sold or returned.
	Quantity Sign	Char(1)	P or N	Sign of the quantity.
	Quantity	Number(12)	N/A	Quantity * 10000 (4 implied decimal places), number of units for the given order (item) status.
	Selling UOM	Char(4)	N/A	UOM at which this item was sold.
	Catchweight Ind	Char(1)	N/A	Indicates if the item is a catchweight item. Valid values are Y or NULL.
	Customer Order number	Char(48)	N/A	Customer Order number.

Table 6-24 (Cont.) Output File Layout

Record Name	Field Name	Field Type	Default Value	Description
	Posting Store	Number(10)		Contains the store at which the item reservation/ reservation cancellation should occur in case of cross-store transactions happening at co-located stores. It is expected that this field will be populated only for items that are to be reserved (or have the reservation canceled) at a different store from the one at which the checkout happened.
TTAIL	File Type Record Descriptor	Char(5)	TTAIL	Identifies file record type.
	File Line Identifier	Number(10)	Specified by Sales Audit	ID of current line being processed by input file.
	Transaction count	Number(6)	Specified by Sales Audit	Number of TDETL records in this transaction set.
FTAIL	File Type Record Descriptor	Char(5)	FTAIL	Identifies file record type.
	File Line Identifier	Number(10)	Specified by external system	ID of the current line being processed by input file.
	File Record Counter	Number(10)		Number of records/ transactions processed in the current file (only records between FHEAD and FTAIL).

Design Assumptions

N/A

Export of POS Transactions from Sales Audit to Merchandising (saexprms)

Module Name	saexprms.pc
Description	Export of POS transactions from Sales Audit to Merchandising
Functional Area	Oracle Retail Sales Audit
Module Type	Integration
Module Technology	ProC
Catalog ID	RSA01
Wrapper Script	rmswrap_multi_dnld_in.ksh

Schedule

Oracle Retail Merchandising Batch Schedule

Design Overview

The purpose of this batch module is to fetch all sale and return transactions that do not have Merchandising errors from the Sales Audit database tables for transmission to the Merchandising system. Transaction data is rolled up to the item/store/day/price point/sales type level for SALES transaction type and item/store/day/price point/sales type/no inventory return indicator/return disposition/return warehouse level for RETURN transaction types.

If unit of work system parameter is defined as 'S', then the whole store/day is skipped if any Merchandising error is found. If this value is 'T', then only transactions with Merchandising errors are skipped.

If the transaction has a status of Deleted and it has previously been transmitted, a reversal of the transaction will be sent.

A file is generated for each store/day.

Restart/Recovery

The logical unit of work for this module is defined as a unique store/day combination. Records will be fetched, updated, and inserted in batches of `pl_commit_max_ctr`. Only two commits will be done, one to establish the store/day lock and another at the end, to release the lock after a store/day has been completely processed. The POSU formatted output file will be created with a temporary name and renamed just before the end of store/day commit.

In case of failure, all work done will be rolled back to the point right after the call to `get_lock()` and the lock will be released. Thus, the rollback segment should be large enough to hold all inserts into `sa_exported` for one store/day.

I/O Specification

Integration Type	Download from Sales Audit
File Name	"POSU_" appended with store number, business date and system date
Integration Contract	IntCon000044

Table 6-25 File Layout

Record Name	Field Name	Field Type	Default Value	Description
FHEAD	Record descriptor	Char(5)	FHEAD	Identifies the file record type.
	File Line Id	Char(10)	0000000001	Sequential file line number.
	File type definition	Char(4)	POSU	Identifies the file type
	File Create Date	Char(14)	N/A	File Create Date in YYYYMMDDHHMMSS format.
	Store	Number(10)	N/A	Store location.

Table 6-25 (Cont.) File Layout

Record Name	Field Name	Field Type	Default Value	Description
	Vat include indicator	Char(1)	N/A	Determines whether or not the store values include VAT. Not required, but populated by Sales Audit.
	Vat region	Number(4)	N/A	VAT region the given location is in. Not required, but populated by Sales Audit.
	Currency code	Char(3)	N/A	Currency of the given location. Not required, but populated by Sales Audit.
	Currency retail decimals	Number(1)	N/A	Number of decimals supported by given the currency for retails. Not required, but populated by Sales Audit.
THEAD	Record descriptor	Char(5)	THEAD	Identifies the file record type.
	File Line Id	Char(10)	N/A	Sequential file line number.
	Transaction date	Char(14)	N/A	Transaction date in YYYYMMDDHHMMSS format. Corresponds to the date that the sale/return transaction was processed at the POS.
	Item Type	Char(3)	REF or ITM	Can be REF or ITM.
	Item	Char(25)	N/A	ID number of the ITM or REF.
	Dept	Number(4)	N/A	Department of item sold or returned.
	Class	Number(4)	N/A	Class of item sold or returned.
	Sub Class	Number(4)	N/A	Subclass of item sold or returned.
	Pack Ind	Char(1)	N/A	Pack indicator of item sold or returned.
	Item Level	Number(1)	N/A	Item level of item sold or returned.
	Tran level	Number(1)	N/A	Transaction level of item sold or returned.
	Wastage Type	Char(6)	N/A	Wastage type of item sold or returned.
	Wastage pct	Number(12)	N/A	Waste pct (4 implied decimal places).
	Tran type	Char(1)	N/A	Transaction type code to specify whether transaction is a sale or a return.
	Drop Shipment indicator	Char(1)	N/A	Indicates whether the transaction is a drop shipment or not.

Table 6-25 (Cont.) File Layout

Record Name	Field Name	Field Type	Default Value	Description
	Total sales qty	Number(12)	N/A	Total sales quantity (4 implied decimal places).
	Selling UOM	Char(4)	N/A	Selling Unit of Measure for the item.
	Sales sign	Char(1)	N/A	Determines if the Total Sales Quantity and Total Sales Value are positive or negative.
	Total Sales Value	Number(20)	N/A	Total sales value of goods sold/ returned (4 implied decimal places).
	Last Date time modified	Char(14)	N/A	Date and time of last modification in YYYYMMDDHHMMSS format.
	Catchweight indicator	Char(1)	N/A	Indicates if item is a catchweight item.
	Total weight	Number(12)	N/A	The actual weight of the item, only populated if catchweight_ind = Y.
	Sub Tran type indicator	Char(1)	N/A	Transaction type for Sales Audit. Valid values are A, D, and NULL.
	Total IG TAX Value	Number(20)	N/A	This indicates total of all IG TAX amount for the item.
	Sales Type	Char(1)	N/A	This column indicates whether the line item is a Regular Sale, a customer order serviced by OMS (External CO), or a customer order serviced by a store (In Store CO).
	No Inventory Return Indicator	Char(1)	N/A	This column contains an indicator that identifies a return without inventory. This is generally a non-required column, but in the case of Returns, this is required.
	Return Disposition	Char(10)	N/A	This column contains the disposition code published by Oracle Retail Warehouse Management System (RWMS0 as part of the Returns upload to OMS.
	Return Warehouse	Char(10)	N/A	This column contains the physical warehouse ID for the warehouse identifier where the item was returned.
	Customer Order No	Char(48)	N/A	This column contains the customer order number ID.
	Fulfillment Order No	Char(48)	N/A	This column contains the fulfillment order number ID.

Table 6-25 (Cont.) File Layout

Record Name	Field Name	Field Type	Default Value	Description
	Fulfillment Loc Type	Char(2)	N/A	This column contains the fulfillment location type. Code for the fulfillment loc type from code_detail where code_type = 'FLTP'
	Fulfillment Loc	Number(10)	N/A	This column contains the fulfillment loc ID.
	Orig Store	Number(10)	N/A	This column contains the original store value for a Return transaction.
	POS Tran Id	Number(20)		This column contains the unique identifier for a sale transaction. This is an Optional field.
	Posting Store	Number(10)		This column contains the store at which the item sale/return should be accounted for in case of cross-store sales happening at co-located stores. It is expected that this field will be populated only for items that are checked out at a different store from the one at which they are originally managed.
TTAX	Record descriptor	Char(5)	TTAX	Identifies the file record type.
	File Line Id	Char(10)	N/A	Sequential file line number.
	Tax Code	Char(6)	N/A	The Tax Code of the item.
	Tax Rate	Number(20)	N/A	The tax rate of the item (10 implied decimal places).
	Total Tax Amount	Number(20)	N/A	The item level tax or prorated transaction level tax of the item (4 implied decimal places).
TDETL	Record descriptor	Char(5)	TDETL	Identifies the file record type.
	File Line Id	Char(10)	N/A	Sequential file line number.
	Promo Tran Type	Char(6)	N/A	Code for the promotional type from code_detail where code_type equals PRMT.
	Promotion Number	Number(10)	N/A	Promotion number from Merchandising.
	Sales quantity	Number(12)	N/A	Sales quantity sold for this promotion type (4 implied decimal places).
	Sales value	Number(20)	N/A	Sales value for this promotion type (4 implied decimal places).
	Discount value	Number(20)	N/A	Discount value for this promotion type (4 implied decimal places).

Table 6-25 (Cont.) File Layout

Record Name	Field Name	Field Type	Default Value	Description
	Promotion component	Number(10)	N/A	Links the promotion to additional pricing attributes. Contains the offer ID from Pricing.
TTAIL	Record descriptor	Char(5)	TTAIL	Identifies the file record type.
	File Line Id	Char(10)	N/A	Sequential file line number.
	Tran Record Counter	Number(6)	N/A	Number of TDETL records in this transaction set.
FTAIL	Record descriptor	Char(5)	FTAIL	Identifies the file record type.
	File Line Id	Number(10)	N/A	Sequential file line number.
	File Record counter	Number(10)	N/A	Number of records/transactions processed in the current file (only records between head and tail).

Fields expected in POSU format based on changes adopted:

	V16	V16 with Customer Order Changes	V19
Fulfillment Order No	No	Yes	Yes
Fulfillment Loc Type	No	Yes	Yes
Fulfillment Loc	No	Yes	Yes
Orig Store	No	Yes	Yes
POS Tran Id	No	No	Yes

Design Assumptions

- Tax can be sent either in TTAX or IGTAX regardless of default_tax_type of SVAT, GTAX, SALES or GTS. Prorated tax in TTAX will only be sent to Merchandising in all configuration.
- POS can send either transactional level tax details in TTAX lines or item-level tax details in IGTAX lines through the RTLOG file to Sales Audit. These tax details will be passed on to Merchandising in the TTAX lines of the POSU file. Even though POS can pass multiple IGTAX/TTAX lines to Sales Audit and from Sales Audit to Merchandising, Merchandising only supports one tax code per item. If multiple taxes for an item are sent from POS to Sales Audit, they will be summed to a single tax in Merchandising sales upload process and assigned one of the applicable tax codes when writing tran_data 88.

Export of Revised Sale/Return Transactions from ReSA to SIM/SIOCS (saexpsim)

Module Name	Saexpsim.pc
Description	Export of Revised Sale/Return Transactions from Sales Audit to SIM
Functional Area	Oracle Retail Sales Audit

Module Type	Integration
Module Technology	ProC
Catalog ID	RSA14
Wrapper Script	rmswrap_multi_dnlid_in.ksh

Schedule

Oracle Retail Merchandising Batch Schedule

Design Overview

The purpose of this batch module is to fetch all revised sale and return transactions that do not have SIM errors from the Sales Audit database tables for transmission to SIM. It retrieves all quantity revision transaction data for SALES, RETURN, EEXCH, VOID, and SPLORD transaction types.

If sa_system_options.unit_of_work is S, the whole store/day is skipped if any SIM error is found. If this value is T, then only transactions with SIM errors are skipped.

The batch will only export transactions whose quantity has been revised. The batch will write these revised transactions to the output file along with a reversal of the quantity.

A file of type SIMT is generated for each store/day.

Restart/Recovery

The logical unit of work for this module is defined as a unique store/day combination. Records will be fetched, updated, and inserted in batches of pl_commit_max_ctr. Only two commits will be done, one to establish the store/day lock and another at the end, to release the lock after a store/day has been completely processed. The SIMT formatted output file will be created with a temporary name and renamed just before the end of store/day commit.

In case of failure, all work done will be rolled back to the point right after the call to get_lock() and the lock released. Thus, the rollback segment should be large enough to hold all inserts into SA_EXPORTED for one store/day.

I/O Specification

Integration Type	Download from Sales Audit
File Name	SIMT_ appended by store number, business date, and system date
Integration Contract	IntCon000045

Output File Layout

Table 6-26 Output File

Record Name	Field Name	Field Type	Default Value	Description
FHEAD	Record descriptor	Char(5)	FHEAD	Identifies the file record type.
	File Line Id	Char(10)	0000000001	Sequential file line number.
	File type definition	Char(4)	SIMT	Identifies the file type.
	Store	Number(10)	N/A	Store location.
	Business Date	Char(8)	N/A	Business Date in YYYYMMDD format.
	File Create Date	Char(14)	N/A	File Create Date in YYYYMMDDHHMMSS format.
THEAD	Record descriptor	Char(5)	THEAD	Identifies the file record type.
	File Line Id	Char(10)	N/A	Sequential file line number.
	Transaction Number	Number(10)	N/A	Transaction Identifier.
	Revision Number	Number(3)	N/A	Revision Number of the transaction.
	Transaction date	Char(14)	N/A	Transaction date in YYYYMMDDHHMMSS format. Corresponds to the date that the transaction occurred.
	Transaction Type	Char(6)	N/A	Transaction Type.
	POS Transaction Indicator	Char(1)	N/A	Indicates if the transaction was received from POS or manually created. Valid values: Y - POS N - Manual
TDETL	Record descriptor	Char(5)	TDETL	Identifies the file record type.
	File Line Id	Char(10)	N/A	Sequential file line number.
	Item Sequence Number	Number(4)	N/A	Item sequence number.
	Item	Char(25)	N/A	Identifies the merchandise item.
	Item number type	Char(6)	N/A	Identifies the type of item number if the item type is ITEM or REF.

Table 6-26 (Cont.) Output File

Record Name	Field Name	Field Type	Default Value	Description
	Item Status	Char(6)	N/A	Status of the item within the transaction, V for item void, S for sold item, R for returned item. ORI - Order Initiate ORC - Order Cancel ORD - Order Complete LIN - Layaway Initiate LCA - Layaway Cancel LCO - Layaway Complete
	Serial Number	Char(128)	N/A	Unique ID.
	Pack Indicator	Char(1)	N/A	Pack Indicator.
	Catchweight Indicator	Char(1)	N/A	Catchweight Indicator.
	Quantity Sign	Char(1)	N/A	Sign of the quantity.
	Quantity Value	Number(12)	N/A	Number of items, with 4 implied decimal places.
	Standard Unit of Measure	Char(4)	N/A	Standard Unit of Measure of the item.
	Selling Unit of Measure	Char(4)	N/A	Unit of Measure of the quantity value.
	Waste Type	Char(6)	N/A	Waste Type.
	Waste Percent	Number(12)	N/A	Waste Percent.
	Drop Ship Indicator	Char(1)	N/A	Indicates whether the item is part of a drop shipment.
	Actual Weight	Number(12)	N/A	Contains the weight of the item sold, with 4 implied decimal places.
	Actual Weight Sign	Char(1)	N/A	Sign of the actual weight.
	Reason Code	Char(6)	N/A	Reason entered by the cashier for some transaction types.
	Sales Value	Number(20)	N/A	Transaction value, with 4 implied decimal places
	Sales Value Sign	Char(1)	N/A	Transaction value sign.
	Unit Retail	Number(20)	N/A	Unit retail, with 4 implied decimal places.
	Sales Type	Char(1)	N/A	Indicates if the transaction is an In Store Customer Order, External Customer Order, or Regular Sale.
	Customer Order Number	Char(48)	N/A	Contains the customer order ID.
	Customer Order Type	Char(6)	N/A	Customer order type.

Table 6-26 (Cont.) Output File

Record Name	Field Name	Field Type	Default Value	Description
	Fulfillment Order Number	Char(48)	N/A	Contains the order ID of the fulfillment order.
	Customer Order Line Number	Number(6)		Contains customer order line number.
TTAIL	Record descriptor	Char(5)	TTAIL	Identifies the file record type.
	File Line Id	Char(10)	N/A	Sequential file line number.
	Tran Record Counter	Number(6)	N/A	Number of TDETL records in this transaction set.
FTAIL	Record descriptor	Char(5)	FTAIL	Identifies the file record type.
	File Line Id	Number(10)	N/A	Sequential file line number.
	File Record counter	Number(10)	N/A	Number of records/transactions processed in the current file (only records between head and tail).

Design Assumptions

N/A

Export to Universal Account Reconciliation System from Sales Audit (saexpuar)

Module Name	saexpuar.pc
Description	Export to Universal Account Reconciliation System from Sales Audit
Functional Area	Oracle Retail Sales Audit
Module Type	Integration
Module Technology	ProC
Catalog ID	RSA06
Wrapper Script	N/A

Schedule

Oracle Retail Merchandising Batch Schedule

Design Overview

The SAEXPUAR program is used to select the lottery, bank deposit, money order, and credit card totals and write them to output files for export to an external account clearing house application. For each store day, saexpuar posts specified totals to their appropriate output files.

Restart/Recovery

The logical unit of work for this module is defined as a unique store/day combination. Records will be fetched, updated, and inserted in batches of commit_max_ctr. Only two commits will be done. One to establish the store/day lock (this will be done by the package) and the other is done at the end, after a store/day has been completely processed.

I/O Specification

Integration Type Download from Sales Audit
File Name UAR usage type appended with system date.
Integration Contract IntCon000046

Output File Layout

The output file will contain one line for each store/day detail record in a comma-delimited format. The fields are surrounded by double quotes. For example, a record for store 1000 on May 20, 2001 with an amount of 19.99 will look something like this:

```
"1", "1000", "1999", "20010520", "2", "", "1", "", "", "", "", "", "", "", "", "MN", "RET"
```

Table 6-27 Output File

Field Name	Field Type	Description
Detail Flag	Char	"1" for detail record.
Store	Number	Store number.
Amount	Number	Total Value * 100 (with 2 implied decimal places).
TranDate	Char	Transaction Date in YYYYMMDD format.
UAR TranCode	Char	Transaction Code. "1" for negative amount, "2" for positive amount.
User Defined Value 1	Char	Ref Number 1 on SA_TOTAL.
User Defined Value 2	Char	Total Seq Number on SA_TOTAL.
User Defined Value 3	Char	Ref Number 2 on SA_TOTAL.
User Defined Value 4	Char	Ref Number 3 on SA_TOTAL.
User Defined Value 5	Char	Not used.
User Defined Value 6	Char	Not used.
User Defined Value 7	Char	Not used.
User Defined Value 8	Char	Not used.
User Defined Value 9	Char	Not used.
User Defined Value 10	Char	Not used.
State	Char	State.
Account	Char	Total Identification on SA_TOTAL.

Design Assumptions

N/A

Extract of POS Transactions by Store/Date from Sales Audit for Web Search (ang_saplgen)

Module Name	ang_saplgen.pc
Description	Extract of POS Transactions by Store/Date from Sales Audit for Web Search
Functional Area	Sales Audit
Module Type	Integration
Module Technology	ProC
Integration Catalog ID	RMS162
Wrapper Script	N/A

Schedule

Oracle Retail Merchandising Batch Schedule

Design Overview

The purpose of this batch module is to fetch all corrected sale and return transactions that do not have Merchandising errors from the Sales Audit database tables for transmission to an external web search engine. If the transaction has a status of Deleted or Post Voided and has previously been transmitted, a reversal of the transaction will be sent. A file of type POSLOG is generated for each store/day.

Restart/Recovery

The logical unit of work for this module is defined as a unique store/day combination. Records will be fetched, in batches of pl_commit_max_ctr. The POSLOG formatted output file will be created with a completion of store/day looping.

I/O Specification

Integration Type	Download from Sales Audit
File Name	POSLOG_<store>_<business date>_<system date>.xml
Integration Contract	IntCon000018

Output File Layout

Table 6-28 Output File Layout

Field Name	Field Type	Description
BatchID	CHAR(18)	A concatenation of store number and business date for a store.

Table 6-28 (Cont.) Output File Layout

Field Name	Field Type	Description
RetailStoreID	CHAR(10)	The store number for which the POSLog file has to be extracted.
WorkStationID	CHAR(5)	RegistryID for the store.
TillID	CHAR(5)	RegistryID for the store.
SequenceNumber	CHAR(10)	Point of Sale system defined transaction number associated with a transaction.
BeginDate	CHAR(8)	Starting date time of the transaction.
EndDate	CHAR(8)	End date time of the transaction.
CurrencyCode	CHAR(3)	Code of the currency used during the transaction.
VoidFlag	CHAR(5)	Indicates if the item in the transaction is voided or not. Valid values are TRUE and FALSE.
Item_Status	CHAR(40)	Status of the item is required for voided, exchanged, or returned item.
MerchandisingHierarchy	CHAR(4)	Department number to which the item belongs
Description	CHAR(250)	Item description that has been sold.
Item	CHAR(25)	Item number.
TaxIncludedInPrice	CHAR(5)	Indicates if the item is being taxed or not. Valid values are TRUE and FALSE.
RegularSalesUnitPrice	CHAR(20)	Field holds the unit retail in the standard unit of retail for the item/location combination.
ActualSalesUnitPrice	CHAR(20)	Retail price for the item.
ExtendedAmount	CHAR(20)	Total sales for the item in the detail level.
Qty	CHAR(21)	Unit sold of the item.

Design Assumptions

N/A

Post User Defined Totals from Sales Audit to General Ledger (saexpgl)

Module Name	saexpgl.pc
Description	Post User Defined Totals from Sales Audit to General Ledger
Functional Area	Oracle Retail Sales Audit
Module Type	Integration
Module Technology	ProC
Integration Catalog ID	RSA09
Wrapper Script	rmswrap.ksh

Schedule

Oracle Retail Merchandising Batch Schedule

Design Overview

The purpose of this module is to post all the properly configured user-defined Sales Audit totals to a general ledger application (Oracle or PeopleSoft). Totals without errors will be posted to the appropriate accounting ledger, as defined in the Sales Audit GL cross-reference module. Depending on the unit of work system parameter, the data will be sent at either the store/day or individual total level. Newly revised totals, that have already been posted to the ledger, will have their previous revision reversed, and the new total posted to the appropriate accounts.

When this module encounters a total that is not mapped to the General Ledger (GL), it will write the same into the IF_ERRORS table and raise a notification that unmapped total/store combinations exist. The IF_ERRORS table is available through the Data Access Schema, which will enable you to query for the error and create the missing mappings. It will also look for records written into IF_ERRORS on previous runs and attempt to reprocess the posting if GL mappings have been created.

'Late posted totals' that are received within a duration specified in the Close Month After Days system option after the end of the fiscal period will be processed by the module and posted to the intended month. All late posted totals received after this specified number of days has elapsed will be recorded against the first day of the subsequent month.

Restart/Recovery

The logical unit of work for this module is defined as a unique store/day combination. Records will be fetched, updated, and inserted in batches the size of commit max counter. Only one commit will be performed after a store/day has been completely processed. A call to the release lock functions performs a commit.

I/O Specification

Integration Type	Download from Sales Audit
File Name	N/A
Integration Contract	IntCon000019 TG_FIF_GL_DATA

Design Assumptions

N/A

Inbound Scheduled Integration

This section provides a summary of integrations that are scheduled either to be run once per day or periodically throughout the day to retrieve data from another solution to Merchandising or Sales Audit. It includes both file-based and BDI-based integrations.

Item, Cost, and Price

Merchandising subscribes to data related to items, costs, and competitive prices from external sources, such as suppliers, PIM solutions, and so on.

The following scheduled inbound integrations are included in this functional area:

- [Upload Competitor's Prices \(cmpupld\)](#)
- [Upload Items and Cost Changes \(iindbatch.ksh\)](#)

Upload Competitor's Prices (cmpupld)

Module Name	cmpupld.pc
Description	Upload Competitor's Prices
Functional Area	Competitive Pricing
Module Type	Integration
Module Technology	ProC
Catalog ID	RMS61
Wrapper Script	rmswrap_in_rej.ksh

Schedule

Oracle Retail Merchandising Batch Schedule

Design Overview

This program is used to upload and process competitor item prices from an external source. The flat file being uploaded can contain pricing data for a completed shopping list or data for a new list of items to be shopped. The module processes data for both features.

Restart/Recovery

This is a file based upload, and file based restart/recovery logic is applied. The `commit_max_ctr` field should be set to prevent excessive rollback space usage and to reduce the overhead of file I/O. The recommended commit counter setting is 10000 records (subject to change based on experimentation).

I/O Specification

Integration Type	Upload to Merchandising
File Name	Determined by runtime parameter
Integration Contract	IntCon000007

Input File Layout

Table 6-29 Input File Layout

Record Name	Field Name	Field Type	Default Value	Description
File Header	File Type Record Descriptor	CHAR (5)	FHEAD	Value that identifies the record type.
	File Line Identifier	NUMBER (10)	0000000001	Sequential file line number.

Table 6-29 (Cont.) Input File Layout

Record Name	Field Name	Field Type	Default Value	Description
	File Type Definition	CHAR(4)	CMPU	Value that identifies the file as that for this program.
	File Create Date	CHAR (14)	N/A	Date when the file was written by the external system. It should be in the YYYYMMDDH H24MISS format.
File Detail	File Type Record Descriptor	CHAR (5)	FDETL	Value that identifies the record type.
	File Line Identifier	NUMBER (10)		Sequential file line number.
	Shopper ID	NUMBER (4)		Numeric value that uniquely identifies the shopper to which the competitive shopping list is assigned.
	Shop Date	CHAR (14)		Date when the competitive shop was performed. It should be in the YYYYMMDDH H24MISS format.
	Item	CHAR (25)		Alphanumeric value that uniquely identifies the transaction level or below transaction level item that was competitively shopped.
	Competitor ID	NUMBER(10)		Numeric value that uniquely identifies a competitor.

Table 6-29 (Cont.) Input File Layout

Record Name	Field Name	Field Type	Default Value	Description
	Competitor Store ID	NUMBER(10)		Numeric value that uniquely identifies a competitor's store.
	Recorded Date	CHAR (14)		Date when the item's retail price was recorded at the competitor's store. It should be in the YYYYMMDD24 MISS format.
	Competitive Retail Price	NUMBER(20, 4)		Numeric value that represents the retail price at the competitor's store. Format for this value should include four implied decimal places.
	Competitive Retail Type	CHAR(6)	R, P, C	Value that represents the retail type ('R' is for regular; 'P', promotional; and 'C', clearance) that was recorded.
	Promotion Start Date	CHAR (14)		Effective start date of the competitor's price. It should be in the YYYYMMDDH H24MISS format.
	Promotion End Date	CHAR (14)		Effective end date of the competitor's price. It should be in the YYYYMMDDH H24MISS format.

Table 6-29 (Cont.) Input File Layout

Record Name	Field Name	Field Type	Default Value	Description
	Offer Type Code	CHAR(6)		Alphanumeric value that corresponds to a valid offer type (such as, Coupon, Bonus Card, Pre-priced). Valid values are defined on CODE_DETAIL table with CODE_TYPE 'OFTP'.
	Multi-Units	NUMBER(12, 4)		Numeric value that represents the number of units that must be purchased to qualify for a multi-unit price. An example of a multi-unit price would be 2 for \$3.00. There are four implied decimal places.
	Multi-Units Retail	NUMBER(20, 4)		Numeric value that represents the price for a multi-unit item that was competitively shopped. There should be four implied decimal places.
File Trailer	File Type Record Descriptor	CHAR(5)	FTAIL	Value that identifies the record type.
	File Line Identifier	NUMBER (10)	N/A	Sequential file line number.
	File Record Counter	NUMBER (10)	N/A	Numeric value that represents the number of FDETL records in the file.

Design Assumptions

- Items included in the file must be defined as transaction level items in Merchandising.

Upload Items and Cost Changes (iindbatch.ksh)

Module Name	iindbatch.ksh
Description	Upload items and cost changes from an external system
Functional Area	Item and Cost Maintenance
Module Type	Integration
Module Technology	ksh
Catalog ID	RMS474
Wrapper Script	rmswrap_shell_in.ksh

Schedule

Oracle Retail Merchandising Batch Schedule

Design Overview

This batch program is used to bulk upload XML data files from template files to the Merchandising templates table. It supports two types of templates - those for items and those for cost changes. The templates used in this upload are the same as those used for spreadsheet upload of items and cost changes.

See also *Oracle Retail Merchandising Induction CSV to XML File Transformer Usage* on My Oracle Support (Doc ID 2730273.1), for more details on formatting XML files for this upload.

This batch will be responsible for validating the input parameters, below are the list of validations.

- The Input file should exist.
- The Input file's extension must be ".xml".
- The template name should be valid.
- Destination (Optional Parameter) determines whether data will be loaded into the main Merchandising tables (RMS) or staging tables for further enrichment (STG). If a destination is not included, then it will be defaulted to STG.

Once XML data is loaded into the staging table, the script will do the following:

- Initializes a row in the process tracker table for asynchronous processing.
- Call the main induction process that uploads data into the staging tables, validates and inserts data into the base Merchandising item or cost change tables.

Note:

The base templates used by this batch are loaded through a script on provisioning (ITEM_MASTER_DATA and COST_CHANGE). Additional templates can be configured using the Data Loading Template Configuration in the Merchandising task list under Application Administration for type Item or Cost Change.

Restart/Recovery

N/A

Design Assumptions

N/A

Ordering and Inventory

Merchandising subscribes to purchasing and inventory data via scheduled integration from external sources, such as stores, warehouses, order management solutions, and import partners.

This section has been broken into the following sub-sections:

- [Purchasing](#)
- [Import Management](#)
- [Stock Counts](#)
- [Franchise](#)
- [Other Inventory](#)

Purchasing

Merchandising subscribes to data related to purchase orders from external sources, such as suppliers, planning solutions, and so on.

The following scheduled inbound integrations are included in this functional area:

- [Upload of Deals from 3rd Party Systems \(dealupld\)](#)
- [Upload Order Data \(poindbatch.ksh\)](#)
- [Upload OTB Budget from Planning Systems \(otbupld\)](#)
- [Upload Purchase Order and Purchase Order Change Acknowledgements from Suppliers to RMS \(ediupack\)](#)
- [Upload Replenishment Data \(replindbatch.ksh\)](#)

For more on purchase order processing, see *Merchandising Operations Guide - Volume 1*.

Upload of Deals from 3rd Party Systems (dealupld)

Module Name	dealupld.pc
Description	Upload of Deals from 3rd Party Systems
Functional Area	Deals
Module Type	Integration
Module Technology	ProC
Catalog ID	RMS42
Wrapper Script	rmswrap_multi_in_rej.ksh

Design Overview

This process uploads deals from external systems into Merchandising. Generally, deals are uploaded from merchandise suppliers and other trading partners. This program uses a proprietary file format (not any EDI standard).

The deals that are uploaded through the batch are created in the worksheet (W) status by default, but can be created in submitted (S) or approved (A) statuses, based on the Deal Upload Status configuration for the supplier. If any validation error occurs during the deal submission or approval, the deal will be created in the worksheet status and a user needs to manually rectify the error through the Deal UI in order to approve it. Please note that this functionality is limited to Supplier based deals. Deals created for other Partners can only be created in the worksheet status.

Assumptions

1. This upload supports two format versions. The fields noted below can be omitted from the file format if not creating deals with a billing type of Clearance Consignment Rate (CCR) or Promotional Consignment Rate (PCR).
 - Consignment Rate in the TDETL record for Transaction Detail Record Type DCDTL.
 - The CPDTL section of the file, including THEAD, TDETL, and TTAIL.

However, if either of these types of deals are being created, the DCDTL TDETL Consignment Rate must be included in the upload; CPDTL section is required for PCR types only.

Restart/Recovery

The program uses File based restart recovery process. The logical unit of work is a single deal head detail record and its associated component records in the input file.

I/O Specification

Integration Type	Upload to Merchandising
File Name	Determined by runtime parameter
Integration Contract	IntCon000008

Input File Layout

Table 6-30 dealupld.pc - Input File Layout

Record Name	Field Name	Field Type	Default Value	Description/Constraints
FHEAD	File Type Record Descriptor	Char(5)	FHEAD	Identifies file record type (the beginning of the input file).
	File Line Identifier	Numeric ID(10)	Sequential number Created by program.	ID of current line being read from input file.

Table 6-30 (Cont.) dealupld.pc - Input File Layout

Record Name	Field Name	Field Type	Default Value	Description/Constraints
THEAD	File Type Definition	Char(5)	EDIDU	Identifies file as 'EDI Deals Upload'
	File Create Date	Char(14)	Create date	Current date, formatted to 'YYYYMMDDHH24MISS'.
	File Type Record Descriptor	Char(5)	THEAD	Identifies file record type to upload a new deal header.
	File Line Identifier	Numeric ID(10)	Sequential number Created by program.	ID of current line being read from input file.
TDETL	Transaction Detail Record Type	Char(5)	DHDTL	Identifies file record type Deal Header. This record MUST BE FOLLOWED BY ONE AND ONLY ONE REQUIRED TDETL RECORD that holds the deal head information.
	File Type Record Descriptor	Char(5)	TDETL	Identifies file record type to upload a new deal.
	File Line Identifier	Numeric ID(10)	Sequential number Created by program.	ID of current line being read from input file.
	Partner Type	Char(6)	REQUIRED	Type of the partner the deal applies to. Valid values are 'S' for a supplier, 'S1' for supplier hierarchy level 1 (for example, the manufacturer), 'S2' for supplier hierarchy level 2 (for example, the distributor) and 'S3' for supplier hierarchy level 3 (that is, the wholesaler). Descriptions of these codes will be held on the codes table under a code_type of 'SUHL'. Information pertaining to a single deal has to belong to the same supplier, since a deal may have only one supplier hierarchy associated with it. Only items with the same supplier hierarchy can be on the same deal. Supplier hierarchy is stored at an item / supplier / country / location level.

Table 6-30 (Cont.) dealupld.pc - Input File Layout

Record Name	Field Name	Field Type	Default Value	Description/Constraints
	Partner Id	Char(10)	Blank (space character string)	<p>Level of supplier hierarchy (for example, manufacturer, distributor or wholesaler), set up as a partner in the PARTNER table, used for assigning rebates by a level other than supplier. Rebates at this level will include all eligible supplier/item/country records assigned to this supplier hierarchy level.</p> <p>This field is required if the Partner Type field was set to 'S1', 'S2' or 'S3'. This field must be blank if the Partner Type field was set to 'S'.</p>
	Supplier	Number (10)	Blank (space character string)	<p>Deal supplier's number. This supplier can be at any level of supplier hierarchy.</p> <p>This field is required if the Partner Type field was set to 'S'. This field must be blank if the Partner Type field was set to 'S1', 'S2' or 'S3'.</p> <p>Deals for items with an ownership of Consignment can only be set up for the primary supplier for the given item/location combination.</p>
	Type	Char(6)	REQUIRED	<p>Type of the deal. Valid values are A for annual deal, P for promotional deal, O for PO-specific deal or M for vendor-funded markdown. Deal types will be held on the codes table under a code type of 'DLHT'.</p>
	Currency Code	Char(3)	Blank (space character string)	<p>Currency code of the deal's currency. All costs on the deal will be held in this currency.</p> <p>If Type is 'O', 'P' or 'A', then Currency Code may not be blank. Currency Code has to be blank if Type is 'M'.</p>
	Active Date	Char(14)	REQUIRED	<p>Date the deal will become active. This date will determine when deal components begin to be factored into item costs. For a PO-specific deal, the active date will be the order's written date.</p>

Table 6-30 (Cont.) dealupld.pc - Input File Layout

Record Name	Field Name	Field Type	Default Value	Description/Constraints
	Close Date	Char(14)	Blank (space character string)	Date the deal will/did end. This date determines when deal components are no longer factored into item costs. It is optional for annual deals, required for promotional deals. It will be left NULL for PO-specific deals. Close Date must not be blank if Type is 'P' or 'M'. Close Date has to be blank if Type is 'O'.
	External Reference Number	Char(30)	Blank (space character string)	Any given external reference number that is associated with the deal.
	Order Number	Number (12)	Blank (space character string)	Order the deal applies to, if the deal is PO-specific.
	Recalculate Approved Orders	Char(1)	REQUIRED	Indicates if approved orders should be recalculated based on this deal once the deal is approved. Valid values are Y for yes or N for no. Valid values are 'Y' and 'N'.
	Comments	Char (2000)	Blank (space character string)	Free-form comments entered with the deal.
	Billing Type	Char(6)	REQUIRED	Billing type of the deal component. Valid values are 'OI' for off-invoice, 'BB' for bill-back, 'VFP' for vendor funded promotion, 'VFM' for vendor funded markdown, 'CCR' for clearance consignment rate and 'PCR' for promotional consignment rate. Billing types are held in the codes table under a code type of 'DLBT'.
	Bill Back Period	Char(6)	Blank (space character string)	Code that identifies the bill-back period for the deal component. This field will only be populated for billing types of 'BB' or 'VFP' or 'VFM'. Valid bill back period codes are 'W', 'M', 'Q', 'H', 'A'. If Billing Type is 'BB', then Bill Back Period must not be blank; if Billing Type is 'OI' (off invoice), 'CCR' (clearance consignment rate), 'PCR' (promotional consignment rate), then Bill back Period has to be blank.

Table 6-30 (Cont.) dealupld.pc - Input File Layout

Record Name	Field Name	Field Type	Default Value	Description/Constraints
	Deal Application Timing	Char(6)	Blank (space character string)	Indicates when the deal component should be applied - at PO approval or time of receiving. Valid values are 'O' for PO approval, 'R' for receiving. These values will be held on the codes tables under a code type of 'AALC'. It must be NULL for an M-type deal (vendor funded markdown).
	Threshold Limit Type	Char(6)	Blank (space character string)	Identifies whether thresholds will be set up as quantity values, currency amount values or percentages (growth rebates only). Valid values are 'Q' for quantity, 'A' for currency amount. Threshold limit types will be held on the codes table under a code type of 'DLLT'. It must be NULL for an M-type deal (vendor funded markdown) or if the threshold value type is 'Q' (buy/get deals) or a 'CCR'/PCR' deal. If Growth Rebate Indicator is 'Y', then the Threshold Limit Type has to be 'Q', 'A' or NULL.
	Threshold Limit Unit of Measure	Char(4)	Blank (space character string)	Unit of measure of the threshold limits, if the limit type is quantity. Only Unit of Measures with a UOM class of 'VOL' (volume), 'MASS' or 'QTY' (quantity) can be used in this field. Valid Unit of Measures can be found on the UOM_CLASS table. If the Threshold Limit Type is 'A', then Threshold Limit Unit of Measure has to be blank. If the Threshold Limit Type is 'Q', Threshold Limit Unit of Measure must not be blank. If Threshold Limit Type is blank, Threshold Limit Unit of Measure must be blank.
	Rebate Indicator	Char(1)	REQUIRED	Indicates if the deal component is a rebate. Deal components can only be rebates for bill-back billing types. Valid values are 'Y' for yes or 'N' for no. If Billing Type is 'OI', 'CCR' or 'PCR', then Rebate Indicator must be 'N'.

Table 6-30 (Cont.) dealupld.pc - Input File Layout

Record Name	Field Name	Field Type	Default Value	Description/Constraints
	Rebate Calculation Type	Char(6)	Blank (space character string)	Indicates if the rebate should be calculated using linear or scalar calculation methods. Valid values are 'L' for linear or 'S' for scalar. This field will be required if the rebate indicator is 'Y'. Rebate calculation types will be held on the codes table under a code type of 'DLCT'. If Rebate Indicator is 'Y', then Rebate Calculation Type must not be blank. Otherwise it has to be blank.
	Growth Rebate Indicator	Char(1)	REQUIRED	Indicates if the rebate is a growth rebate, meaning it is calculated and applied based on an increase in purchases or sales over a specified period of time. Valid values are 'Y' for yes or 'N' for no. If Rebate Indicator is 'N', then Growth Rebate Indicator must be 'N'.
	Historical Comparison Start Date	Char(14)	Blank (space character string)	The first date of the historical period against which growth will be measured in this growth rebate. Note performance and the rebate amount are not calculated - this field is for informational/reporting purposes only. If Growth Rebate Indicator is 'Y', then Historical Comparison Start Date must not be blank. Otherwise it must be blank.
	Historical Comparison End Date	Char(14)	Blank (space character string)	The last date of the historical period against which growth will be measured in this growth rebate. Note performance and the rebate amount are not calculated - this field is for informational/reporting purposes only. If Growth Rebate Indicator is 'Y', then Historical Comparison End Date must not be blank. Otherwise it must be blank.

Table 6-30 (Cont.) dealupld.pc - Input File Layout

Record Name	Field Name	Field Type	Default Value	Description/Constraints
	Rebate Purchases or Sales Application Indicator	Char(6)	Blank (space character string)	Indicates if the rebate should be applied to purchases or sales. Valid values are 'P' for purchases or 'S' for sales. It will be required if the rebate indicator is 'Y'. Rebate purchase/sales indicators will be held on the codes table under a code type of 'DLRP'. If the Rebate Indicator is 'Y', then the Rebate Purchases or Sales Application Indicator must not be blank. Otherwise it has to be blank.
	Security Indicator	Char	Y	Security Indicator
TTAIL	File Line Identifier	Char(5)	TTAIL	Identifies file record type (the end of the transaction detail).
	File Line Identifier	Numeric ID(10)	Sequential number Created by program.	ID of current line being read from input file.
	Transaction Record Counter	Numeric ID(6)	Sequential number Created by program.	Number of records/transactions in current transaction set (only records between head and ttail). For DHDTL TDETL records this will always be 1!
THEAD	File Type Record Descriptor	Char(5)	THEAD	Identifies file record type to upload a new deal sub loop.
	File Line Identifier	Numeric ID(10)	Sequential number Created by program.	ID of current line being read from input file.
	Transaction Detail Record Type	Char(5)	DCDTL	Identifies file record type of sub loop as Deal Component Detail.
TDETL	File Type Record Descriptor	Char(5)	TDETL	Identifies file record type to upload deal components.
	File Line Identifier	Numeric ID(10)	Sequential number Created by program.	ID of current line being read from input file.
	Deal Component Type	Char(6)	REQUIRED	Type of the deal component, user-defined and stored on the DEAL_COMP_TYPE table.

Table 6-30 (Cont.) dealupld.pc - Input File Layout

Record Name	Field Name	Field Type	Default Value	Description/Constraints
	Application Order	Number (10)	Blank (space character string)	Number indicating the order in which the deal component should be applied with respect to any other deal components applicable to the item within the deal. This number will be unique across all deal components within the deal. It must be NULL for an M-type deal (vendor funded markdown).
	Collect Start Date	Char(14)	Blank (space character string)	Date that collection of the bill-back should begin. If Billing Type is 'BB' then Collect Start Date must not be blank, otherwise it has to be blank.
	Collect End Date	Char(14)	Blank (space character string)	Date that collection of the bill-back should end. If Billing Type is 'BB' then Collect End Date must not be blank, otherwise it has to be blank.
	Cost Application Level Indicator	Char(6)	Blank (space character string)	Indicates what cost bucket the deal component should affect. Valid values are 'N' for net cost, 'NN' for net cost and 'DNN' for dead net cost. These values will be held on the codes tables under a code type of 'DLCA'. It must be NULL for an M-type deal (vendor funded markdown), 'CCR' or 'PCR' deals
	Pricing Cost Indicator	Char(1)	REQUIRED	Identifies deal components that should be included when calculating a pricing cost. Valid values are 'Y'es and 'N'o.
	Deal Class	Char(6)	Blank (space character string)	Identifies the calculation class of the deal component. Valid values are 'CU' for cumulative (discounts are added together and taken off as one lump sum), 'CS' for cascade (discounts are taken one at a time with subsequent discounts taken off the result of the previous discount) and 'EX' for exclusive (overrides all other discounts). 'EX' type deal components are only valid for promotional deals. Deal classes will be held on the codes table under a code type of 'DLCL'. It must be NULL for an M-type deal (vendor funded markdown), 'CCR' and 'PCR' deals.

Table 6-30 (Cont.) dealupld.pc - Input File Layout

Record Name	Field Name	Field Type	Default Value	Description/Constraints
	Threshold Value Type	Char(6)	Blank (space character string)	<p>Identifies whether the discount values associated with the thresholds will be set up as qty values, currency amount values, percentages or fixed amounts. Valid values are 'Q' for qty, 'A' for currency amount, 'P' for percentage or 'F' for fixed amount. Qty threshold value (buy/get) deals are only allowed on off-invoice discounts. Deal threshold value types will be held on the codes table under a code type of 'DLL2'. It must be NULL for an M-type deal (vendor funded markdown), 'CCR' and 'PCR' deals.</p> <p>If Billing Type is 'BB', then the Threshold Value Type must be 'A' or 'P'.</p>
	Buy Item	Char(25)	Blank (space character string)	<p>Identifies the item that must be purchased for a quantity threshold-type discount. This value is required for quantity threshold value type discounts. Otherwise it has to be blank.</p>
	Get Type	Char(6)	Blank (space character string)	<p>Identifies the type of the 'get' discount for a quantity threshold-type (buy/get) discount. Valid values include 'X' (free), 'P' (percent), 'A' (amount) and 'F' (fixed amount). They are held on the codes table under a code type of 'DQGT'. This value is required for quantity threshold value deals. Otherwise it has to be blank.</p>
	Get Value	Number(20,4)	All 0s.	<p>Identifies the value of the 'get' discount for a quantity threshold-type (buy/get) discount that is not a 'free goods' deal. The Get Type above identifies the type of this value. This value is required for quantity threshold value type deals that are not a Get Type of free. Otherwise it has to be 0.</p> <p>If Get Type is 'P', 'A' or 'F', then Get Value must not be blank. If the Get Type is 'X' or blank, then Get Value has to be blank.</p>

Table 6-30 (Cont.) dealupld.pc - Input File Layout

Record Name	Field Name	Field Type	Default Value	Description/Constraints
	Buy Item Quantity	Number(12,4)	All 0s.	Identifies the quantity of the threshold 'buy' item that must be ordered to qualify for the 'free' item. This value is required for quantity threshold value type discounts. Otherwise it has to be 0.
	Recursive Indicator	Char(1)	REQUIRED	For 'buy/get free' discounts, indicates if the quantity threshold discount is only for the first 'buy amt.' purchased (such as, for the first 10 purchased, get 1 free), or if a free item will be given for every multiple of the 'buy amt' purchased on the order (such as, for each 10 purchased, get 1 free). Valid values are 'Y' for yes or 'N' for no. If the Get Type is blank, then Recursive Indicator has to be 'N'.
	Buy Item Order Target Quantity	Number(12,4)	All 0s.	Indicates the targeted purchase level for all locations on a purchase order. This is the target level that will be used for future calculation of net cost. This value is required for quantity threshold value type deals. Otherwise it has to be 0.
	Average Buy Item Order Target Quantity Per Location	Number(12,4)	All 0s.	Indicates the average targeted purchase level per location on the deal. This value will be used in future cost calculations. This value is required for quantity threshold value type deals. Otherwise it has to be 0.
	Get Item	Char(25)	Blank (space character string)	Identifies the 'get' item for a quantity threshold-type (buy/get) discount. This value is required for quantity threshold value deals. Otherwise it has to be blank. If Get Type is 'P', 'A', 'F' or 'X', then Get Item must not be blank. If the Get Type is blank, then Get Item has to be blank.

Table 6-30 (Cont.) dealupld.pc - Input File Layout

Record Name	Field Name	Field Type	Default Value	Description/Constraints
	Get Quantity	Number(12,4)	All 0s.	<p>Identifies the quantity of the identified 'get' item that will be given at the specified 'get' discount if the 'buy amt' of the buy item is purchased. This value is required for quantity threshold value type discounts. Otherwise it has to be 0.</p> <p>If Get Type is 'P', 'A', 'F' or 'X', then Get Quantity must not be 0. If the Get Type is blank, then Get Quantity has to be 0.</p>
	Free Item Unit Cost	Number(20,4)	All 0s.	<p>For 'buy/get free' discounts, identifies the unit cost of the threshold 'free' item that will be used in calculating the prorated qty. discount. It will default to the item/supplier cost, but can be modified based on the agreement with the supplier. It must be greater than zero as this is the cost that would normally be charged for the goods if no deal applied.</p> <p>If Get Type is 'P', 'A', 'F' or blank, then Free Item Unit Cost must be 0. If the Get Type is 'X', then Free Item Unit Cost must not be 0.</p>
	Transaction Level Discount Indicator	Char(1)	REQUIRED	<p>Indicates if the discount is a transaction-level discount (for example, 10% across an entire PO).</p> <p>Valid Values are 'Y' or 'N'. If set to 'Y', Deal Class has to be 'CU' and Billing Type has to be 'OI'.</p> <p>For 'CCR', and 'PCR' deals, the valid value is 'N'. No DIDTL or PPDTL records may be present for a Transaction Level Discount DCDTL record.</p>
	Comments	Char(2000)	Blank (space character string)	Free-form comments entered with the deal component.

Table 6-30 (Cont.) dealupld.pc - Input File Layout

Record Name	Field Name	Field Type	Default Value	Description/Constraints
	Get Free Discount	Number(12,4)	All 0s.	This specifies how much percentage of the total discount should be apportioned from the get items unit cost for off invoice deals where buy item is not same as the get item and QTY_THRESH_GET_TYPE is X. The remaining will be apportioned from the buy item unit cost.
	Consignment Rate	Number(12,4)	All 0s.	Rate used to capture the deal consignment rate applicable for the set of item/location combinations that are included in the deal during the deal timeframe, instead of the regular consignment rate.
TTAIL	File Line Identifier	Char(5)	TTAIL	Identifies file record type (the end of the transaction detail).
	File Line Identifier	Numeric ID(10)	Sequential number Created by program.	ID of current line being read from input file.
	Transaction Record Counter	Numeric ID(6)	Sequential number Created by program.	Number of records/transactions in current transaction set (only records between thead and ttail).
THEAD	File Type Record Descriptor	Char(5)	THEAD	Identifies file record type to upload a new deal sub loop.
	File Line Identifier	Numeric ID(10)	Sequential number Created by program.	ID of current line being read from input file.
	Transaction Detail Record Type	Char(5)	CPDTL	Identifies file record type of sub loop as Deal Component Prom.
TDETL	File Type Record Descriptor	Char(5)	TDETL	Identifies file record type to upload deal proof of performance details.
	File Line Identifier	Numeric ID(10)	Sequential number Created by program.	ID of current line being read from input file.
	Prom ID	Number (10)	All 0s.	Promotion identification number.
	Promo Comp Id	Number (10)	All 0s.	Promotion offer identification number.

Table 6-30 (Cont.) dealupld.pc - Input File Layout

Record Name	Field Name	Field Type	Default Value	Description/Constraints
	Consignment Rate	Number(12,4)	All 0s.	Rate used to capture the deal consignment rate applicable for the set of item/location combinations that are included in the deal during the deal timeframe, instead of the regular consignment rate.
	Reference Line	Number (10)	REQUIRED	This value determines which line in the input file this deal component-promotional record belongs to.
TTAIL	File Line Identifier	Char(5)	TTAIL	Identifies file record type (the end of the transaction detail).
	File Line Identifier	Numeric ID(10)	Sequential number Created by program.	ID of current line being read from input file.
	Transaction Record Counter	Numeric ID(6)	Sequential number Created by program.	Number of records/transactions in current transaction set (only records between thead and ttail).
THEAD	File Type Record Descriptor	Char(5)	THEAD	Identifies file record type to upload a new deal sub loop.
	File Line Identifier	Numeric ID(10)	Sequential number Created by program.	ID of current line being read from input file.
	Transaction Detail Record Type	Char(5)	DIDTL	Identifies file record type of sub loop as Deal Component Item-location Detail.
TDETL	File Type Record Descriptor	Char(5)	TDETL	Identifies file record type to upload deal item-location details.
	File Line Identifier	Numeric ID(10)	Sequential number Created by program.	ID of current line being read from input file.

Table 6-30 (Cont.) dealupld.pc - Input File Layout

Record Name	Field Name	Field Type	Default Value	Description/Constraints
	Merchandise Level	Char(6)	REQUIRED	Indicates what level of the merchandise hierarchy the record is at. Valid values include '1' for company-wide (all items), '2' for division, '3' for group, '4' for dept, '5' for class, '6' for subclass, '7' for line, '8' for line/differentiator 1, '9' for line/differentiator 2, '10' for line/differentiator 3, '11' for line/differentiator 4 and '12' for. These level types will be held on the codes table under a code type of 'DIML'.
	Company Indicator	Char(1)	REQUIRED	Indicates if the deal component is applied company-wide (that is, whether all items in the system will be included in the discount or rebate). Valid values are 'Y' for yes and 'N' for no.
	Division	Number (4)	Blank (space character string)	ID of the division included in or excluded from the deal component. Valid values are on the DIVISION table. If Group is not blank, then Division must not be blank. If Merchandise Level is 2, then Division must not be blank and Group, Department, Class and Subclass must be blank.
	Group	Number (4)	Blank (space character string).	ID of the group included in or excluded from the deal component. Valid values are on the GROUPS table. If Department is not blank, then Group must not be blank. If Merchandise Level is 3, then Group must not be blank and Department, Class and Subclass must be blank.
	Department	Number (4)	Blank (space character string).	ID of the department included in or excluded from the deal component. Valid values are on the DEPS table. If Class is not blank, then Department must not be blank. If Merchandise Level is 4, then Department must not be blank and Class and Subclass must be blank.

Table 6-30 (Cont.) dealupld.pc - Input File Layout

Record Name	Field Name	Field Type	Default Value	Description/Constraints
	Class	Number (4)	Blank (space character string).	ID of the class included in or excluded from the deal component. Valid values are on the CLASS table. If Subclass is not blank, then Class must not be blank. If Merchandise Level is 5, then Class must not be blank and Subclass must be blank.
	Subclass	Number (4)	Blank (space character string).	ID of the subclass included in or excluded from the deal component. Valid values are on the SUBCLASS table. If Merchandise Level is 6 or more than 6, then Subclass must not be blank.
	Item Parent	Char(25)	Blank (space character string)	Alphanumeric value that uniquely identifies the item/group at the level above the item. This value must exist as an item in another row on the ITEM_MASTER table. If Merchandise Level is 7, then Item Parent or Item Grandparent must not be blank (at least one of them has to be given).
	Item Grandparent	Char(25)	Blank (space character string)	Alphanumeric value that uniquely identifies the item/group two levels above the item. This value must exist as both an item and an item parent in another row on the ITEM_MASTER table. If Merchandise Level is 7, then Item Parent or Item Grandparent must not be blank (at least one of them has to be given).
	Differentiator 1	Char(10)	Blank (space character string)	Diff_group or diff_id that differentiates the current item from its item_parent. If Item Grandparent, Item Parent and Differentiator 2 are blank, then Differentiator 1 must be blank. If Merchandise Level is 8, then Differentiator 1 must not be blank.

Table 6-30 (Cont.) dealupld.pc - Input File Layout

Record Name	Field Name	Field Type	Default Value	Description/Constraints
	Differentiator 2	Char(10)	Blank (space character string)	Diff_group or diff_id that differentiates the current item from its item_parent. If Item Grandparent, Item Parent and Differentiator 1 are blank, then Differentiator 2 must be blank. If Merchandise Level is 9, then Differentiator 2 must not be blank.
	Differentiator 3	Char(10)	Blank (space character string)	Diff_group or diff_id that differentiates the current item from its item_parent. If Item Grandparent, Item Parent and Differentiator 1 and 2 are blank, then Differentiator 3 must be blank. If Merchandise Level is 10, then Differentiator 3 must not be blank.
	Differentiator 4	Char(10)	Blank (space character string)	Diff_group or diff_id that differentiates the current item from its item_parent. If Item Grandparent, Item Parent and Differentiator 1, 2 and 3 are blank, then Differentiator 4 must be blank. If Merchandise Level is 10, then Differentiator 4 must not be blank.
	Organizational Level	Char(6)	Blank (space character string)	Indicates what level of the organizational hierarchy the record is at. Valid values include '1' for chain, '2' for area, '3' for region, '4' for district and '5' for location. These level types will be held on the codes table under a code type of 'DIOL'. If company indicator is N, this must not be blank. If location type is warehouse or location list, this must be 5.
	Chain	Number (10)	Blank (space character string).	ID of the chain included in or excluded from the deal component. Valid values are on the CHAIN table. If org. level is 1, this field must not be blank.
	Area	Number (10)	Blank (space character string).	ID of the area included in or excluded from the deal component. Valid values are on the AREA table. If org. level is 2, this field and chain must not be blank.

Table 6-30 (Cont.) dealupld.pc - Input File Layout

Record Name	Field Name	Field Type	Default Value	Description/Constraints
	Region	Number (10)	Blank (space character string).	ID of the region included in or excluded from the deal component. Valid values are on the REGION table. If org. level is 3, this field, area, and chain must not be blank.
	District	Number (10)	Blank (space character string).	ID of the district included in or excluded from the deal component. Valid values are on the DISTRICT table. If org. level is 4, then this field, region, area, and chain must not be blank.
	Location	Number (10)	Blank (space character string).	ID of the location included in or excluded from the deal component. Valid values are on the STORE, WH, or LOC_LIST_HEAD table. If org. level is 5, this field must not be blank. Chain, area, region, and district should be blank if the loc_type is L or W. If the loc_type is S, then they all must not be blank. If Location Type is not blank, then Location must not be blank. Otherwise it has to be blank.
	Origin Country Identifier	Char(3)	Blank (space character string)	Origin country of the item that the deal component should apply to.
	Location Type	Char(1)	Blank (space character string)	Type of the location referenced in the location field. Valid values are 'S' and 'W'. Location types will be held on the codes table under the code type 'LOC3'. If location is blank then this field has to be blank also.
	Item	Char(25)	Blank (space character string)	Unique alphanumeric value that identifies the item. If Merchandise Level is 10, then Item must not be blank.
	Exclusion Indicator	Char(1)	REQUIRED	Indicates if the deal component item/location line is included in the deal component or excluded from it. Valid values are 'Y' for yes or 'N' for no.
	Reference Line	Number (10)	REQUIRED	This value determines which line in the input file this item-loc record belongs to.

Table 6-30 (Cont.) dealupld.pc - Input File Layout

Record Name	Field Name	Field Type	Default Value	Description/Constraints
TTAIL	File Line Identifier	Char(5)	TTAIL	Identifies file record type (the end of the transaction detail).
	File Line Identifier	Numeric ID(10)	Sequential number Created by program.	ID of current line being read from input file.
	Transaction Record Counter	Numeric ID(6)	Sequential number Created by program.	Number of records/transactions in current transaction set (only records between thead and ttail).
THEAD	File Type Record Descriptor	Char(5)	THEAD	Identifies file record type to upload a new deal sub loop.
	File Line Identifier	Numeric ID(10)	Sequential number Created by program.	ID of current line being read from input file.
	Transaction Detail Record Type	Char(5)	PPDTL	Identifies file record type of sub loop as Proof of Performance Detail.
TDETL	File Type Record Descriptor	Char(5)	TDETL	Identifies file record type to upload deal proof of performance details.
	File Line Identifier	Numeric ID(10)	Sequential number Created by program.	ID of current line being read from input file.
	Deal Sub Item	Char(25)	No data	Specific transaction level (or below) item that's proof of performance is being measured. This can be populated when the deal itself is on a case UPC but the proof of performance is on an individual selling unit.
	Proof of Performance Type	Char(6)	REQUIRED	Code that identifies the proof of performance type (that is, the term is that the item must be displayed on an end cap for 28 days - the pop_type is code 'ECD' for end cap display). Valid values for this field are stored in the code_type = 'PPT'. This field is required by the database.

Table 6-30 (Cont.) dealupld.pc - Input File Layout

Record Name	Field Name	Field Type	Default Value	Description/Constraints
	Proof of Performance Value	Number (20,4)	All 0s.	Value that describes the term of the proof of performance type (that is, the term is that the item must be displayed on an end cap for 28 days - the pop_value is 28). This field is required by the database if the record has a pop_value_type. If Proof of Performance Value is not blank, then Proof of Performance Value Type must not be blank. If Proof of Performance Value is blank, then Proof of Performance Value Type must be blank.
	Proof of Performance Value Type	Char(6)	Blank (space character string)	Value that describes the type of the pop_value (that is, the term is that the item must be displayed on an end cap for 28 days - the pop_value_type is the code 'DAYS' for days). Valid values for this field are stored in the code_type = 'PPVT'. This field is required by the database if the record has a pop_value. If Proof of Performance Value is not blank, then Proof of Performance Value Type must not be blank. If Proof of Performance Value is blank, then Proof of Performance Value Type must be blank.
	Vendor Recommended Start Date	Char(14)	Blank (space character string)	This column holds the date that the vendor recommends that the POP begin.
	Vendor Recommended End Date	Char(14)	Blank (space character string)	This column holds the date that the vendor recommends that the POP end.
	Planned Start Date	Char(14)	Blank (space character string)	This column holds the date that the merchandiser/category manager plans to begin the POP.
	Planned End Date	Char(14)	Blank (space character string)	This column holds the date that the merchandiser/category manager plans to end the POP.
	Comment	Char(255)	Blank (space character string)	Free-form comments.
	Reference Line	Number (10)	REQUIRED	This value determines which line in the input file this Proof of Performance record belongs to.

Table 6-30 (Cont.) dealupld.pc - Input File Layout

Record Name	Field Name	Field Type	Default Value	Description/Constraints
TTAIL	File Line Identifier	Char(5)	TTAIL	Identifies file record type (the end of the transaction detail).
	File Line Identifier	Numeric ID(10)	Sequential number Created by program.	ID of current line being read from input file.
	Transaction Record Counter	Numeric ID(6)	Sequential number Created by program.	Number of records/transactions in current transaction set (only records between thead and ttail).
THEAD	File Type Record Descriptor	Char(5)	THEAD	Identifies file record type to upload a new deal sub loop.
	File Line Identifier	Numeric ID(10)	Sequential number Created by program.	ID of current line being read from input file.
	Transaction Detail Record Type	Char(5)	DTDTL	Identifies file record type of sub loop as Deal Component Threshold Detail.
TDETL	File Type Record Descriptor	Char(5)	TDETL	Identifies file record type to upload deal threshold details.
	File Line Identifier	Numeric ID(10)	Sequential number Created by program.	ID of current line being read from input file.
	Lower Limit	Number (20,4)	REQUIRED	Lower limit of the deal component. This is the minimum value that must be met in order to get the specified discount. This value will be either a currency amount or quantity value, depending on the value in the deal_detail.threshold_limit_type field of this deal component (Threshold Value Type field of the DCDTL record that this DTDTL record belongs to as specified in the reference line field).

Table 6-30 (Cont.) dealupld.pc - Input File Layout

Record Name	Field Name	Field Type	Default Value	Description/Constraints
	Upper Limit	Number (20,4)	REQUIRED	Upper limit of the deal component. This is the maximum value for which the specified discount will apply. This value will be either a currency amount or quantity value, depending on the value in the deal_detail.threshold_limit_type field of this deal component (Threshold Value Type field of the DCDTL record that this DTDTL record belongs to as specified in the reference line field).
	Value	Number (20,4)	REQUIRED	Value of the discount that will be given for meeting the specified thresholds for this deal component. This value will be either a currency amount or quantity value, depending on the value in the deal_detail.threshold_value_type field of this deal component (Threshold Value Type field of the DCDTL record that this DTDTL record belongs to as specified in the reference line field).
	Target Level Indicator	Char(1)	REQUIRED	Indicates if a threshold level is the targeted purchase or sales level for a deal component. This indicator will be used for cost calculations. Valid values are 'Y' for yes and 'N' for no.
	Reference Line	Number (10)	REQUIRED	This value determines which line in the input file this Threshold record belongs to.
TTAIL	File Line Identifier	Char(5)	TTAIL	Identifies file record type (the end of the transaction detail).
	File Line Identifier	Numeric ID(10)	Sequential number Created by program.	ID of current line being read from input file.
	Transaction Record Counter	Numeric ID(6)	Sequential number Created by program.	Number of records/transactions in current transaction set (only records between thead and ttail).
FTAIL	File Line Identifier	Char(5)	FTAIL	Identifies file record type (the end of the input file).
	File Line Identifier	Numeric ID(10)	Sequential number Created by program.	ID of current line being read from input file.

Table 6-30 (Cont.) dealupld.pc - Input File Layout

Record Name	Field Name	Field Type	Default Value	Description/Constraints
	File Record Counter	Numeric ID(10)	Sequential number Created by program.	Number of records/transactions in current file (only records between head and tail).

The input file structure should be as below:

```

FHEAD
{
THEAD of DHDTL   REQUIRED   for deal head record
  TDETL          REQUIRED   1 deal head record
  TTAIL          REQUIRED   end of deal head record
  THEAD of DCDTL REQUIRED   for deal component records
  [
    TDETL          OPTIONAL for deal component records
  ]
  TTAIL          REQUIRED   end of deal component records
  THEAD of CPDTL OPTIONAL  for deal component promotion records
  [
    TDETL          OPTIONAL for deal component promotion records
  ]
  TTAIL          OPTIONAL  end of deal component promotion records
  THEAD of DIDTL REQUIRED   for item-loc records
  [
    TDETL          OPTIONAL for item-loc records
  ]
  TTAIL          REQUIRED   end of item-loc records
  THEAD of PPDTL REQUIRED   for proof of performance records
  [
    TDETL          OPTIONAL for proof of performance records
  ]
  TTAIL          REQUIRED   end of proof of performance records
  THEAD of DTDTL REQUIRED   for threshold records
  [
    TDETL          OPTIONAL for threshold records
  ]
  TTAIL          REQUIRED   end of threshold records
}
FTAIL
  THEAD of DIDTL REQUIRED   for item-loc records
  [
    TDETL          OPTIONAL for item-loc records
  ]
  TTAIL          REQUIRED   end of item-loc records
  THEAD of PPDTL REQUIRED   for proof of performance records
  [
    TDETL          OPTIONAL for proof of performance records
  ]
  TTAIL          REQUIRED   end of proof of performance records
  THEAD of DTDTL REQUIRED   for threshold records
  [
    TDETL          OPTIONAL for threshold records
  ]
  TTAIL          REQUIRED   end of threshold records

```

```
}  
FTAIL
```

Upload Order Data (poindbatch.ksh)

Module Name	poindbatch.ksh
Description	Upload Order Data
Functional Area	Purchase Order Maintenance
Module Type	Integration
Module Technology	Ksh
Catalog ID	RMS234
Wrapper Script	rmswrap_shell_in.ksh

Schedule

Oracle Retail Merchandising Batch Schedule

Design Overview

This batch program is used to Bulk upload xml file data from template files to S9T_FOLDER table (into content_xml column).

This batch will be responsible for validating the input parameters, below are the list of validations.

- The Input file should exist.
- The Input file's extension must be ".xml".
- The template_name should be valid. Function S9T_PKG.CHECK_TEMPLATE is called for validation.
- Destination (Optional Parameter) should be STG or Merchandising. If destination is not passed then default it to STG.

Once XML data is loaded into S9T_FOLDER table, the script will do post processing by calling the packages listed below:

- PO_INDUCT_SQL.INIT_PROCESS - This initialize a row in svc_process_tracker for asynchronous processing.
- PO_INDUCT_SQL.EXEC_ASYNC - This function calls the main induction process that uploads data into the staging tables, validates and inserts data into the base Merchandising purchase order tables.

Note:

The base templates used by this batch are loaded through a script on provisioning (PURCHASE_ORDER_DATA). Additional templates can be configured using the Data Loading Template Configuration in the Merchandising task list under Application Administration for type Purchase Orders.

Restart/Recovery

N/A

Design Assumptions

N/A

Upload OTB Budget from Planning Systems (otbupld)

Module Name	otbupld.pc
Description	Upload OTB Budget from Planning Systems
Functional Area	Open To Buy
Module Type	Integration
Module Technology	ProC
Catalog ID	RMS132
Wrapper Script	rmswrap_multi_in_rej.ksh

Schedule

Oracle Retail Merchandising Batch Schedule

Design Overview

The purpose of this batch module is to accept new and updated open to buy (OTB) budget data from an external planning system. Merchandising supports three types of OTB budgets – those associated with Non-Basic (N/B), Buyer Replenished Basic (BRB) and Auto-Replenished Basic (ARB) orders, as defined by the Order type on Merchandising purchase orders. OTB budgets are created by subclass/end of week date in Merchandising.

Restart/Recovery

Processing of each row is independent and thus if an erroneous record is found during processing; only that record needs to be corrected and reprocessed.

If a record fails validation, it will be written to a rejected record file. This file will facilitate easy reprocessing once the error is fixed by writing the record exactly as it was in the source file.

I/O Specification

Integration Type	Upload to Merchandising
File Name	Determined by runtime parameter
Integration Contract	IntCon000033

Input File Layout

Table 6-31 otbupld - Input File

Record Name	Field Name	Field Type	Default Value	Description
FHEAD	File head descriptor	Char(5)	FHEAD	Describes file line type
	Line id	Number(10)	0000000001	Sequential file line number
	File Type Definition	Char(4)	'OTBI'	Identifies file as 'OTB Import'
	File Create Date	Char(14)	N/A	The date on which the file was written by external system. The Date is in YYYYMMDDH H24MISS format
	Subclass	Number(4)		The ID number of a subclass within the class given
FDETL	Eow Date	Char(14)		The end of week date for the budgeted week in YYYYMMDDH H24MISS format
	File record descriptor	Char(5)	FDETL	Describes file line type
	Line ID	Number(10)	N/A	Sequential file line number
	Transaction Set Control Number	Number(14)	N/A	Sequence number used to force unique transaction check
	Order Type	Char(1)	N/A	Order type budgeted for: specified as A for ARB, B for BRB, and N for N/B
	Department	Number(4)	N/A	The ID number of a department

Table 6-31 (Cont.) otbupld - Input File

Record Name	Field Name	Field Type	Default Value	Description
	Class	Number(4)	N/A	The ID number of a class within the department given
	Subclass	Number(4)	N/A	The ID number of a subclass within the class given
	Eow Date	Char(14)	N/A	The end of week date for the budgeted week in YYYYMMDDH H24MISS forma
	Budget Amount	Number(20)	N/A	Budgeted amount for the specified order type/ week; value includes 4 implied decimal places
FTAIL	File record descriptor	Char(5)	N/A	Marks end of file
	Line ID	Number(10)	Line number in file	Sequential file line number
	Number of lines	Number(10)	Total detail lines	Number of lines in file not counting FHEAD and FTAIL

Design Assumptions

- POs with an Order Type of DSD and Customer Order do not impact open to buy.

Upload Purchase Order and Purchase Order Change Acknowledgements from Suppliers to RMS (ediupack)

Module Name	ediupack.pc
Description	Upload Purchase Order and Purchase Order Change Acknowledgements from Suppliers to Merchandising
Functional Area	Purchase Orders

Module Type	Integration
Module Technology	ProC
Catalog ID	RMS48
Wrapper Script	rmswrap_in_rej.ksh

Schedule

Oracle Retail Merchandising Batch Schedule

Design Overview

This program has four functions:

1. to acknowledge vendor receipt of a buyer-generated order without changes (acknowledge type AK)
2. to acknowledge vendor receipt of a buyer-generated order with date, cost or quantity modifications (acknowledge type AK)
3. to notify buyer of a new or updated vendor-generated order (acknowledge type AP)
4. to acknowledge order cancellations (acknowledge type CA)

All acknowledgements update the ORDHEAD table with acknowledgement information.

When the supplier sends the acknowledgement of a buyer order with modifications, they can send the entire purchase order or only the changes. The file details are matched to the current order. If the Not Before Date, Not After Date, Quantity, Price, and item all match the current order, then no changes were submitted. If one of the variables is blank, for example the price, assume that no pricing changes were made. As soon as one of the variables does not match, the order has been changed. These changes will not be written directly to the order; they will be written to the revision tables. Revisions will be accepted in the on-line ordering screens and changed orders will be resubmitted via EDIDLORD.

Vendor generated orders will create new orders by inserting new records on the EDI temporary order tables that are picked up by a subsequent process (VRPLBLD). For revisions to a vendor-generated order, updates will be made to the order automatically without requiring user acceptance. If the update is to add a new item/location to the order, this will generate a new purchase order using the same vendor reference number.

For Customer Order POs created through an external Order Management System (OMS) and Franchise Order POs, the modifications to the dates, quantity and cost are applied automatically (and will not need to be accepted online). Also, changes to Franchise POs through this program will not affect their associated Franchise orders.

Restart/Recovery

The files will not have enough volume to warrant the implementation of restart recovery for commit/rollback considerations but minimal file-based restart/recovery capability will be added. The logical unit of work is a complete transaction represented by detail lines between the transaction header and transaction tail.

A savepoint will be issued before each transaction header record is successfully processed. If a non-fatal error occurs, a rollback to the last savepoint will be issued so that the rejected records are not posted to the database. If a fatal error occurs and restart is necessary, processing will restart at the last commit point.

I/O Specification

Integration Type	Upload to Merchandising
File Name	Determined by runtime parameter
Integration Contract	IntCon000014

Input File Layout

Table 6-32 ediupack - Input File

Record Name	Field Name	Field Type	Default Value	Description
FHEAD	File head descriptor	Char(5)	FHEAD	Describes file line type
	Line id	Number(10)	0000000001	Sequential file line number
	File Type Definition	Char(4)	ORAK	Identifies file as 'Order Acknowledgment Import'
THEAD	File record descriptor	Char(5)	THEAD	Describes file line type
	Line id	Number(10)	Line number in file	Sequential file line number
	Transaction number	Number(10)	N/A	Sequential transaction number
	Acknowledge type	Char(2)	N/A	AP-product replenishment (VMI orders and updates) AK- Acknowledge or change CA-cancel order (no detail)
	Order number	Char(15)	N/A	May be external order number (vendor order number) OR Oracle Retail order number
	Written_date	Char(8)	N/A	Written date in YYYYMMDD format
	Supplier number	Number(10)	N/A	Supplier number
	Not before date	Char(8)	N/A	Not_before_date YYYYMMDD
	Not after date	Char(8)	N/A	Not_after_date YYYYMMDD
	Purchase type	Char(6)	N/A	Specifies type of purchase – may be blank
	Pickup date	Char(8)	N/A	Pickup_date YYYYMMDD – may be blank
TITEM	File record descriptor	Char(5)	TITEM	Describes file line type
	Line id	Number(10)	Line number in file	Sequential file line number

Table 6-32 (Cont.) ediupack - Input File

Record Name	Field Name	Field Type	Default Value	Description
	Transaction number	Number(10)	N/A	Sequential transaction number
	ITEM	Char(25)	N/A	Item (either item or ref_item must be defined)
	Ref_item	Char(25)	N/A	Reference item (either item or ref_item must be defined)
	Vendor catalog number	Char(30)	N/A	VPN (Vendor Product Number)
	Unit cost value	Number(20)	N/A	Unit_cost * 10000 (4 implied decimal places)
	Loc_type	Char(2)	N/A	'ST' for store, 'WH' for warehouse
	Location	Number(10)	N/A	If NULL, apply to all locations for this item
	Pickup location	Char(250)	N/A	Location to pick up item – may be blank
TSHIP	File record descriptor	Char(5)	TSHIP	Describes file line type
	Line id	Number(10)	Line number in file	Sequential file line number
	Transaction number	Number(10)	N/A	Sequential transaction number
	Store/wh indicator	Char(2)	N/A	'ST' for store, 'WH' for warehouse
	Ship to location	Number(10)	N/A	Store or warehouse number
	Quantity	Number(12)	N/A	Quantity ordered * 10000 (4 implied decimal places)
TTAIL	File record descriptor	Char(5)	TTAIL	Describes file line type
	Line id	Number(10)	Line number in file	Sequential file line number
	Transaction number	Number(10)	N/A	Sequential transaction number
	Lines in transaction	Number(6)	N/A	Total number of lines in this transaction
FTAIL	File record descriptor	Char(5)	FTAIL	Marks end of file
	Line id	Number(10)	Line number in file	Sequential file line number
	Number of transactions	Number(10)	'NA	Number of lines between FHEAD and FTAIL

Design Assumptions

N/A

Upload Replenishment Data (replindbatch.ksh)

Module Name	replindbatch.ksh
Description	Upload replenishment schedule
Functional Area	Inventory Movement
Module Type	Integration
Module Technology	Ksh
Catalog ID	RMS475
Wrapper Script	rmswrap_shell_in.ksh

Schedule

Oracle Retail Merchandising Batch Schedule

Design Overview

This batch program is used to Bulk upload xml file data from template files to a staging table (into the content XML column).

This batch will be responsible for validating the input parameters, below are the list of validations.

- The input file should exist.
- The input file's extension must be ".xml".
- The template_name should be valid. A package function will be called for validation.

Once xml data is loaded into the staging table, the script will do the following:

- Initialize a row in the process tracker table for asynchronous processing.
- Call the main induction process that uploads data into the staging tables, validates and inserts data into the base Merchandising replenishment schedule tables.

Note:

The base templates used by this batch are loaded through a script on provisioning (REPLENISHMENT_DATA). Additional templates can be configured using the Data Loading Template Configuration in the Merchandising task list under Application Administration for type Replenishment.

Restart/Recovery

N/A

Design Assumptions

N/A

Import Management

When using the Import Management features in Merchandising, there are several inbound integration processes that are available for harmonized tariff schedules (HTS), transportation, and letter of credit functions. If you are using Simplified Import Management (based on your system options configurations), then only the HTS upload is supported.

For additional information about import management, including detailed flow diagrams, see the *RTM Overview* white paper in the Merchandising Documentation Library (Doc ID: 1585843.1).

The following integrations are included in this section:

- [Harmonized Tariff Schedule Upload \(htsupld\)](#)
- [Letter of Credit Confirmation Upload \(lcupld\)](#)
 - [SWIFT File Conversion - Letter of Credit Confirmation \(lcmt730\)](#)
- [Letter of Credit Drawdowns and Charges Upload \(lcup798\)](#)
 - [SWIFT File Conversion - Letter of Credit Drawdowns and Charges \(lcmt798\)](#)
- [Transportation Upload \(tranupld\)](#)

Harmonized Tariff Schedule Upload (htsupld)

Module Name	htsupld.pc
Description	Harmonized Tariff Schedule Upload
Functional Area	Oracle Retail Trade Management
Module Type	Integration
Module Technology	ProC
Catalog ID	RMS41
Wrapper Script	rmswrap_multi_in_rej.ksh

Schedule

Oracle Retail Merchandising Batch Schedule

Design Overview

The harmonized tariff schedule batch module processes a file containing the most recent tariff schedule, by country of import, into Merchandising tables. The module uploads both the initial entry of the schedule and all the updates, as they become available.

Restart/Recovery

Recommended commit counter is 2000. Input file names must end in a ".1" for the restart mechanism to properly parse the file name. Because there is only 1 input file to be uploaded, only 1 thread is used.

A reject file is used to hold records that have failed processing. You can fix the rejected records and process the reject file again.

I/O Specification

Integration Type	Upload to Merchandising
File Name	Determined by runtime parameter
Integration Contract	IntCon000051

Input File Layout

Table 6-33 Input File Layout

Record Name	Field Name	Field Type	Default Value	Description
FHEAD	Record Descriptor	Char(5)	FHEAD	Describes file line type
	Line number	Number(10)	0000000001	Sequential file line number
	File ID	Char(5)	HTSUP	Describes file type
THEAD	Record Descriptor	Char(5)	THEAD	Describes file line type
	Line number	Number(10)	N/A	Sequential file line number
	Transaction id	Number(14)	N/A	Unique transaction id
	HTS Line	Char(358)	N/A	V1 through V4 records from the customs HTS file concatenated together
TDETL	Record Descriptor	Char(5)	TDETL	Describes file line type
	Line number	Number(10)	N/A	Sequential file line number
	Transaction id	Number(10)	N/A	Unique transaction id
	Tax/fee line	Char(80)	N/A	V5 through VC records from the customs HTS file, each on a separate TDETL line
TTAIL	Record Descriptor	Char(5)	TTAIL	Describes file line type
	Line number	Number(10)	N/A	Sequential file line number

Table 6-33 (Cont.) Input File Layout

Record Name	Field Name	Field Type	Default Value	Description
	Detail lines	Number(6)	N/A	Number of lines between THEAD and TTAIL
FTAIL	Record Descriptor	Char(5)	FTAIL	Describes file line type
	Line number	Number(10)	N/A	Sequential file line number
	Transaction Lines	Number(10)	N/A	Number of lines between FHEAD and FTAIL

Original Input File

 **Note:**

The input file contains lines of 2400 characters (that is, the newline character occurs only after every 2400 characters). Each 2400-character line consists of thirty 80-character records. Each 80-character record starts with 'V1' or 'V2' ... or 'VD' or blank if the record is completely empty. For each tariff, records V1 and V2 are mandatory; records V3 through VD are optional, which means they can be all blank. Record V4 is not currently used in Merchandising/Trade Management. Records V5 through VC contain the tax/fee information for the tariff, and all have the same structure.

Record Name	Field Name	Field Type	Default Value	Description
V1 a	Control identifier	Char(1)	V	Identifies start of record
b	Record type	Char(1)	1	Identifies record type
c	Tariff number	Number(25)		Harmonized Tariff Schedule (HTS) code or tariff number used to classify merchandise for import. If this number is less than 25 positions, it is left justified.
d	Transaction code	Char(1)	A, D, R	A code representing the type of transaction. Valid Transaction Codes are: A = Add D = Delete R = Replace

Record Name	Field Name	Field Type	Default Value	Description
e	Begin effective date	char(6)		A numeric date in MMDDYY (month, day, year) format representing the record begin effective date. This date indicates when the record becomes effective.
f	End effective date	char(6)		A numeric date in MMDDYY (month, day, year) format representing the record end effective date. This date indicates the last date the record is effective.
g	number of reporting units	number(1)	0,1,or 2 or 3	The number of reporting units required by the Bureau of the Census. In a few instances, units not required by Census may be required to compute duty. In these cases, the Census reporting units are always first, followed by any additional units required to compute the duty.
h	1 st reporting unit of measure	char(4)		A code representing the first unit of measure. Valid unit of measure codes are found on the Units of Measure Class table (uom_class) in Merchandising.
l	2 nd reporting unit of measure	char(4)		A code representing the second unit of measure. Valid unit of measure codes are found on the Units of Measure Class table (uom_class) in Merchandising.
j	3 rd reporting unit of measure	char(4)		A code representing the third unit of measure. Valid unit of measure codes are found on the Units of Measure Class table (uom_class) in Merchandising.
k	duty computation code	char(1)		A code indicating the formula to be used to compute the duty. Valid Duty Computation Codes are found under the Duty Computation Codes (DCMP) code type.
l	commodity description	char(30)		A condensed version of the commodity description that appears in the HTS.
m	column 1 specific rate of duty	Number(12)		The specific rate of duty (monetary amount per unit of measure) applied for imports in general when no conditional tariff treatments are applicable. Within Merchandising this rate is stored against the Column 1 (C1) tariff treatment. Eight decimal places are implied.
n	base rate indicator	char(1)	'B' or blank	A code indicating if the rate contains a base rate. If the base rate indicator is <i>B</i> , the duty rate is a base rate; otherwise, space fill. Not Used in RMS.

Record Name	Field Name	Field Type	Default Value	Description
o	space fill	char(1)	blank	Space fill. Not used in RMS.
V2 a	Control identifier	char(1)	V	Identifies start of record
b	Record type	char(1)	2	Identifies record type
c	tariff number	Number (25)		Harmonized Tariff Schedule (HTS) code or tariff number used to classify merchandise for import. If this number is less than 25 positions, it is left justified. This number is the same as that in Record Identifier V1.
d	general column 1 ad valorem percentage	Number (12)		The ad valorem rate of duty applied for imports in general when no conditional tariff treatments are applicable. Within Merchandising this rate is stored against the Column 1 (C1) tariff treatment. Eight decimal places are implied.
e	column 1 other	Number (12)		The other rate of duty applied for imports in general when no conditional tariff treatments are applicable. Within Merchandising this rate is stored against the Column 1 (C1) tariff treatment. Eight decimal places are implied.
f	Column 2 specific rate	Num(12)		The specific rate of duty (monetary amount per unit of measure) applied for imports in general when the country of origin is not in good standing with the country of import and higher rates of duty are assessed to deter trade. Within Merchandising this rate is stored against the Column 2 (C2) tariff treatment. Eight decimal places are implied.
g	Column 2 ad valorem percentage	Num(12)		The ad valorem rate of duty applied for imports in general when the country of origin is not in good standing with the country of import and higher rates of duty are assessed to deter trade. Within Merchandising this rate is stored against the Column 2 (C2) tariff treatment. Eight decimal places are implied.
h	Column 2 other rate	Num(12)		The other rate of duty applied for imports in general when the country of origin is not in good standing with the country of import and higher rates of duty are assessed to deter trade. Within Merchandising this rate is stored against the Column 2 (C2) tariff treatment. Eight decimal places are implied.

Record Name	Field Name	Field Type	Default Value	Description
i	countervailing duty flag	char(1)	blank or 1	A code of 1 indicating the tariff number is subject to countervailing duty; otherwise, space fill.
j	additional tariff indicator	char(1)	blank or 'R'	A code indicating if an additional HTS code or tariff number may be required to fully classify the item. This indicator is R when an additional tariff number may be required; otherwise, space fill.
k	Miscellaneous Permit/License Indicator	char(2)		A code indicating if a tariff number may be subject to a miscellaneous permit/license number.
l	space fill	char(4)	blanks	Not used in RMS.
V3 a	Control identifier	char(1)	V	identifies start of record
b	Record type	char(1)	3	identifies record type
c	tariff number	Number(25)		Harmonized Tariff Schedule (HTS) code or tariff number used to classify merchandise for import. If this number is less than 25 positions, it is left justified. This number is the same as that in Record Identifier V1.
d	GSP excluded countries	char(20)		The International Organization for Standardization (ISO) country code that indicates countries not eligible for preferential treatment under GSP. Valid country codes are found on the country table (country).
e	OGA codes	char(15)		Codes that indicate special requirements by Other Government Agencies (OGA) may apply. Up to five 3 position OGA codes can be provided.
f	anti-dumping flag	char(1)	1 or blank	A code of 1 indicating the tariff number is subject to an antidumping duty; otherwise, space fill.
g	quota indicator	char(1)	1 or blank	A code of 1 indicating the tariff number may be subject to quota. If the tariff number is not subject to quota, space fill.
h	category number	char(6)		A code located in the HTS indicating the textile category assigned to the tariff number. If there is no textile category number, space fill.

Record Name	Field Name	Field Type	Default Value	Description
I	special program indicators Special Program Indicator (SPI) / Tariff Treatment	char(60)		Special Program Indicator or Tariff Treatment codes that indicate if a tariff number is subject to a special program with preferential rates of duty. Up to fourteen 2 position codes can be reported. Left justify. The tariff treatment codes are not reported in any particular sequence. If more than fourteen 2-position codes are required, they are reported on the VD record.
	NEWLINE		\n	
V4 a	Control identifier	char(1)	V	identifies start of record. Entire V4 record not used in RMS.
b	Record type	char(1)	4	identifies record type
c	tariff number	Number (25)		Harmonized Tariff Schedule (HTS) code or tariff number used to classify merchandise for import. If this number is less than 25 positions, it is left justified. This number is the same as that in Record Identifier V1.
d	value edit code	char(3)		A code representing the value edit.
e	value low bounds	Number (10)		A value representing the minimum value edit. Five decimal places are implied. If this record contains date edits (positions 36-53), space fill.
f	value high bounds	Number (10)		A value representing the maximum value edit. Five decimal places are implied. If this record contains date edits (positions 36-53), space fill.
g	entry date restriction	Number (1)	0,1, or 2	A code representing the first entry date restriction code.
h	beginning restriction date	char(4)		A numeric date in MMDD (month and day) format representing the first begin restriction date used in the edit. If this record contains a value edit (positions 13-35), space fill.
I	end restriction date	char(4)		A numeric date in MMDD (month and day) format representing the first end restriction date used in the edit. If this record contains a value edit (positions 13-35), space fill.
j	entry date restriction 2	number(1)	0,1, or 2	A code representing the second entry date restriction code.
k	beginning restriction date 2	char(4)		A numeric date in MMDD (month and day) format representing the second begin restriction date used in the edit. If this record contains a value edit (positions 13-35), space fill.

Record Name	Field Name	Field Type	Default Value	Description
l	end restriction date 2	char(4)		A numeric date in MMDD (month and day) format representing the second end restriction date used in the edit. If this record contains a value edit (positions 13-35), space fill.
m	country of origin	char(2)		A code representing the value edit.
n	space filler	char(2)	blanks	A value representing the minimum value edit. Five decimal places are implied. If this record contains date edits (positions 36-53), space fill.
o	quantity edit code	char(3)		A value representing the maximum value edit. Five decimal places are implied. If this record contains date edits (positions 36-53), space fill.
p	low quantity	Number (10)		A code representing the first entry date restriction code.
q	high quantity	Number (10)		A numeric date in MMDD (month and day) format representing the first begin restriction date used in the edit. If this record contains a value edit (positions 13-35), space fill.
V5 a	Control identifier	char(1)	V	Identifies start of record
b	Record type	char(1)	5,6,7,8,9,A,B,C	Identifies record type
c	tariff number	Number (25)		Harmonized Tariff Schedule (HTS) code or tariff number used to classify merchandise for import. If this number is less than 25 positions, it is left justified. This number is the same as that in Record Identifier V1.
d	Country code Special Program Indicator (SPI) / Tariff Treatment / Country Code	char(2)		A code representing a tariff treatment/special program indicator which may or may not be the same as a valid ISO country code. Single character tariff treatment codes should include a space after the code to fill the 2-character space per code.
e	specific rate	Number (12)		The specific rate of duty listed in the Special column of the HTS. Eight decimal places are implied.
f	ad valorem rate	Number (12)		The ad valorem rate of duty listed in the Special column of the HTS. Eight decimal places are implied.
g	Other rate	Number (12)		The rate of duty listed in the Special column of the HTS that is not a specific or ad valorem rate. Eight decimal places are implied.

Record Name	Field Name	Field Type	Default Value	Description
h	tax/fee class code	char(3)		A code representing the tax/fee class or type. The system assumes that any class/type code which is less than 23 (i.e. 016, 017, 018, 022, etc.) is a tax, and any value greater than or equal to 23 is a fee (i.e. 023, 024, 045, 056, etc.).
l	tax/fee computation code	char(1)		A code indicating the first tax/fee computation formula. Valid Computation formulas for taxes and fees are found under the Duty Computation Codes (DCMP) code type, but computation codes 0, J and K are only valid for duty, not for tax and fee computations.
j	tax/fee flag	number(1)		A code indicating a tax/fee is required. Valid Tax/Fee Flag Codes are: 1 = Tax/fee required 2 = Tax/fee may be required. Not used in RMS.
k	tax/fee specific rate	Number (12)	blank if no value	The specific rate of duty required to compute taxes and/or fees. Eight decimal places are implied.
l	tax/fee ad valorem	Number (12)	blank if no value	The ad valorem rate of duty required to compute taxes and/or fees. Eight decimal places are implied.
m	space fill	char(1)	blank	Space fill.
Note: V6 through VC records have the same fields as the V5 record.				
VD a	Control identifier	char(1)	V	identifies start of record
b	Record type	char(1)	D	identifies record type
c	tariff number	Number (25)		unique tariff number
d	Special Program Indicator (SPI) Code / Tariff Treatment	char(32)		Special Program Indicator or Tariff Treatment codes that indicate if a tariff number is subject to a special program with preferential rates of duty. Up to sixteen 2-position codes can be reported. Left justify. The codes are not reported in any particular sequence.
e	Filler	char(36)		Space fill.

Design Assumptions

N/A

Letter of Credit Confirmation Upload (Icupld)

Module Name Icupld.pc

Description	Letter of Credit Confirmation Upload
Functional Area	Oracle Retail Trade Management
Module Type	Integration
Module Technology	ProC
Catalog ID	RMS55
Wrapper Script	Rmswrap_in_rej.ksh

Schedule

Oracle Retail Merchandising Batch Schedule

Design Overview

The LCUPLD program is used to upload LC (Letter of Credit) confirmations from bank partners.

After this program has processed a confirmation, the appropriate tables will be updated; a confirmation will update the LC to confirm status and it will write the appropriate records to the LC_ACTIVITY table.

Restart/Recovery

Restart/recovery for this program is set up at the individual FDETL record. Although there may be more than one FDETL record for a given LC, they will each be processed as a separate entity.

File based restart/recovery must be used. The commit_max_ctr field should be set to prevent excessive rollback space usage, and to reduce the overhead of file I/O. The recommended commit counter setting is 10000 records.

I/O Specification

Integration Type	Upload to Merchandising
File Name	Determined by runtime parameter
Integratin Contract	IntCon000054

Input File Layout

Table 6-34 Input File Layout

Record Name	Field Name	Field Type	Default Value	Description
File Header	File Type Record Descriptor	Char(5)	FHEAD	Identifies file record type
	File Line Sequence Number	Number(10)	0000000001	Line number of the current file
	File Type Definition	Char(4)	LCUP	Identifies file as 'Letter of Credit Upload'

Table 6-34 (Cont.) Input File Layout

Record Name	Field Name	Field Type	Default Value	Description
File Detail	File Create Date	Char (14)	vdate	Date file was written by external system 'YYYYMMDDHH24MISS' format
	File Type Record Descriptor	Char(5)	FDETL	Identifies file record type
	File Line Sequence Number	Number(10)		Line number of the current file
	Sender's Reference	Char(16)	lc_head.bank_l c_id	The LC number that the bank assigns to a Letter of Credit
	Receiver's Reference	Number(8)	lc_activity.lc_ref _id	The LC number that Trade Management assigned to the Letter of Credit
	Date of Message Being Acknowledged	Char(14)	lc_activity.acti vity_date	YYYYMMDDHH24MISS format
File Trailer	Comments	Char(2000)	lc_activity.com ments	This field is a concatenation of the following SWIFT fields: 71B – Charges, 72 – Sender information
	File Type Record Descriptor	Char(5)	FTAIL	Identifies file record type
	File Line Sequence	Number(10)	N/A	Line number of the current file
	Total number lines	Number(10)	N/A	Total number of lines in file not including FHEAD and FTAIL

Letter of Credit Drawdowns and Charges Upload (Icup798)

Module Name	Icup798.pc
Description	Letter of Credit Drawdowns and Charges
Functional Area	Oracle Retail Trade Management
Module Type	Integration
Module Technology	ProC

Catalog ID RMS54
Wrapper Script rmswrap_in_rej.ksh

Schedule

Oracle Retail Merchandising Batch Schedule

Design Overview

This program reads data from an input file containing letter of credit charges and drawings (in standard Oracle Retail format, modified from the SWIFT 798 format by the lcmt798 Perl script), validates it, and inserts it into the LC_ACTIVITY table. If a record fails validation, it will be written to a reject file. These rejected records can be reprocessed by lcup798 after errors have been corrected.

Restart/Recovery

This program will be restartable but not threadable.

Restart/recovery logic for file-based processing is used. Records will be committed to the database when commit_max_ctr defined in the RESTART_CONTROL table is reached.

I/O Specification

Integration Type Upload to Merchandising
File Name Determined by runtime parameter
Integratin Contract IntCon000055

The input file for this batch program is the output from the lcmt798 Perl script.

Input File Layout

Table 6-35 Input File Layout

Record Name	Field Name	Field Type	Default Value	Description
FHEAD	File head descriptor	Char(5)	FHEAD	Describes file line type
	Line id	Number (10)	0000000001	Sequential file line number
	File Type Definition	Char(4)	'LCCH'	Identifies as an LC 798 file-Letter of Credit Charges
	Current date	Date	N/A	File date in YYYYMMDDHH 24MISS format
FDETL	File record descriptor	Char(5)	FDETL	Describes file line type
	Line id	Number (10)		Sequential file line number

Table 6-35 (Cont.) Input File Layout

Record Name	Field Name	Field Type	Default Value	Description
	Bank letter of credit reference ID	Char (16)	SWIFT tag 20	Bank's LC ref ID
	Order number	Number(8)	SWIFT tag 21	Order number attached to LC. May be blank
	Invoice number	Number (15)	SWIFT tag 23	NOT a Merchandising invoice number, just a reference invoice number from the issuing bank. May be blank
	Transaction number	Number (10)	N/A	Amendment number or transaction number assigned by bank. May be null
	Transaction code	Char(6)	B or D	'B'ank charge or 'D'rawdown
	Amount	Number(21)	SWIFT tag 33A,71A	(This is a 20-digit number with a leading – sign or blank and 4 implied decimal places.) Amount of charge or drawdown
	Currency code	Char(3)	SWIFT 33A,71A	Currency that the amount is in
	Activity date	Date	SWIFT 33A,32C,32D	Activity date(formatted as 'YYYYMMDD')
	Comments	Char(2000)	SWIFT tag 72	Any comments associated with activity. May be null
FTAIL	File record descriptor	Char(5)	FTAIL	Marks end of file
	Line id	Char(10)	N/A	Sequential file line number
	Number of lines	Number(10)	N/A	Number of lines in file not counting FHEAD and FTAIL

SWIFT File Conversion - Letter of Credit Confirmation (lcmt730)

Module Name	lcmt730
Description	SWIFT File Conversion – Letter of Credit Confirmation
Functional Area	Oracle Retail Trade Management
Module Type	Integration
Module Technology	Perl
Catalog ID	RMS138
Wrapper Script	batch_lcmt730.ksh

Schedule

Oracle Retail Merchandising Batch Schedule

Design Overview

The lcmt730 Perl script converts letter of credit confirmations from a S.W.I.F.T. format (MT730) to a Merchandising flat file format. The output file from this script will be the input file for the lcupld.pc.

I/O Specification

Integration Type	Upload to Merchandising
File Name	Determined by runtime parameter
Integratin Contract	IntCon000054 (output) IntCon000139 (input)

Input File Layout

Table 6-36 Input File Layout

SWIFT I.D. and Description	Data Type	Description	How MT 730 fields are put into the Merchandising standard file format and what should be the size of Merchandising to be dealt with	Comments
20 - Sender's Reference	16x	LC number. The one assigned by the Sender (issuing bank)	FDETL - Sender's reference, Char(16)	This field maps to Trade Management's Bank LC Ref ID.

Table 6-36 (Cont.) Input File Layout

SWIFT I.D. and Description	Data Type	Description	How MT 730 fields are put into the Merchandising standard file format and what should be the size of Merchandising to be dealt with	Comments
21 - Receiver's Reference	16x	LC number assigned by the Receiver (retailer)	FDETL - Receiver's reference, Number(8) (NOREF used if unknown)	This field maps to Trade Management's LC Ref ID. If this field has 'NOREF', the record must be rejected since this field is used to indicate the LC within Trade Management to which this record applies.
25 - Account Identification	35x	Identifies the number of the account, which has been used for the settlement of charges, on the books of the Sender.	N/A	Trade Management currently does not have fields that map directly to this. Current position - will be included in the input file. However, it will be ignored during the upload process.
30 - Date of Message Being Acknowledged	6!n	When a message is acknowledging a MT700, this field specifies the date of issue. In all other cases, this field specifies the date on which the message being acknowledged was sent.	FDETL - Date of message Being Acknowledged, Date	This field maps to the LC activity date. As well, if this in confirming an LC application, it will be mapped to the LC's confirmation date. Year interpretation: If YY>79 then YYMMDD = 19YYMMDD Else YYMMDD = 20YYMMDD.

Table 6-36 (Cont.) Input File Layout

SWIFT I.D. and Description	Data Type	Description	How MT 730 fields are put into the Merchandising standard file format and what should be the size of Merchandising to be dealt with	Comments
32a - Amount of Charges	Option B - 3!a15d Option D - 6!n3!a15d	Contains the currency code and total amount of charges claimed by the sender of the message. When charges have been debited, D is used (:32D) and when reimbursement for charges is needed, B is used (:32B).	FDETL -Upload_type = 'C'onfirmation	Current position - Because the 730 will only be used for confirmations, this field will not contain any values. The upload type should be set equal to 'C'onfirmation.
57a - Account With Bank	Option A - [/1!a]/[34x] 4!a2!a2!c] 3!c] Option D - [/1!a]/[34x] 4*35x	This field specifies the bank to which the amount of charges is to be remitted in favor of the Sender.	FDETL - Account With Bank, Char(10)	Current position - will be added to the input file however will be ignored in the upload process. Because Trade Management has no facilities to maintain BICs or party identifiers, option D will always be used for this field (that is, 57D) without [/1!a]/[34x] party identifier.

Table 6-36 (Cont.) Input File Layout

SWIFT I.D. and Description	Data Type	Description	How MT 730 fields are put into the Merchandising standard file format and what should be the size of Merchandising to be dealt with	Comments
71B - Charges	6*35x	Specification of the charges claimed.	FDETL - Comments, Char(2000)	This field maps to Trade Management's activity comments field. Sender to Receiver information (72) will be concatenated to this.
72 - Sender to Receiver Information	6*35x	Text explanation if wanted.	FDETL - Comments, Char(2000)	This field maps to Trade Management's activity comments field. Charges (71B) will be concatenated to this.

Output File Layout

Table 6-37 Output File Layout

Record Name	Field Name	Field Type	Default Value	Description
File Header	File Type Record Descriptor	Char(5)	FHEAD	Identifies file record type
	File Line Sequence Number	Number(10)	specified by external system	Line number of the current file
	File Type Definition	Char(4)	LCUP	Identifies file as 'Letter of Credit Upload'
	File Create Date	Char (14)	vdate	date file was written by external system 'YYYYMMDD HH24MISS' format

Table 6-37 (Cont.) Output File Layout

Record Name	Field Name	Field Type	Default Value	Description
File Detail	File Type Record Descriptor	Char(5)	FDETL	Identifies file record type
	File Line Sequence Number	Number(10)	specified by external system	Line number of the current file
	Sender's Reference	Char(16)	lc_head.bank_id_id	The LC number that the bank assigns to a Letter of Credit
	Receiver's Reference	Number(8)	lc_activity.lc_ref_id	The LC number that Merchandising assigned to the Letter of Credit
	Date of Message Being Acknowledged	Date (char 8)	lc_activity.activity_date	If the upload type is 'L' then this date will match the date MT 700 date of issue (which we have not resolved between being the vdate or the lc_head.application_date) 'YYYYMMDD' format
	Comments	Char(2000)	lc_activity.comments	Need to truncate? This field will probably be a concatenation of the following SWIFT fields: 71B – Charges, 72 – Sender information
File Trailer	File Type Record Descriptor	Char(5)	FTAIL	Identifies file record type
	File Line Sequence	Number(10)	Specified by external system	Line number of the current file
	Total number of lines	Number(10)	Specified by external system	Total number lines in file

Design Assumptions

N/A

SWIFT File Conversion - Letter of Credit Drawdowns and Charges (lcmt798)

Module Name	lcmt798
Description	SWIFT File Conversion – Letter of Credit Drawdowns and Charges
Functional Area	Retail Trade Management - Letter of Credit Interfaces
Module Type	Integration
Module Technology	Perl
Catalog ID	RMS139
Wrapper Script	batch_lcmt798.ksh

Schedule

Oracle Retail Merchandising Batch Schedule

Design Overview

This Perl script converts letter of credit (L/C) activity data for charges and drawdowns from a S.W.I.F.T. format input file to a Merchandising format file.

I/O Specification

Integration Type	Upload to Merchandising
File Name	Determined by runtime parameter
Integration Contract	IntCon000139 (input)

Input File Layout

Table 6-38 Input File Layout

Swift Tag	Description	Regd?	Datatype	Merchandising Field
20 - Transaction Reference Number	The sender's unambiguous identification of the transaction. Its detailed form and content are at the discretion of the sender.	Yes	16x - Transaction Reference Number	Bank L/C ID Lc_head.bank_l c_id Varchar2(16)

Table 6-38 (Cont.) Input File Layout

Swift Tag	Description	Regd?	Datatype	Merchandising Field
12 - Type of Financial Instrument	This field classifies the financial instrument by a description or proprietary code.	Yes	Option A- :4!c/[8c]/30x :4!c - Qualifier / - Delimiter [8c] - Issuer Code / - Delimiter 30x - Type	This field will contain a constant identifier - '798'
77E - Proprietary Message	This field contains the proprietary message in a format agreed to by the Sender and the Receiver.	Yes	Option E- 73x [n*78x]	This field will contain the information below (fields 21, 23, 32C, 32D, 71A, 33A, 72) Carriage return, Line feed, Colon 'CrLf:' will be used to separate fields included in this 77E For example: :77E:'CrLf' :21:10004321:CrLf' :32C:990121USD1045 and so on. There may be multiple 77Es in one file

Table 6-38 (Cont.) Input File Layout

Swift Tag	Description	Regd?	Datatype	Merchandising Field
21 - Related Reference	This field specifies, in an unambiguous way, a message or transaction identifier which is normally included as part of the information supplied with the message or transaction itself, and can subsequently be used to distinguish the message or transaction identified from other messages or transactions.	No	16x	P/O Number Lc_activity.order_no Number(8)
23 - Further identification	This field specifies the type of transaction being confirmed, as well as the settlement method used.	No	16x	Invoice Number Lc_activity.invoice_no Varchar2(15)

Table 6-38 (Cont.) Input File Layout

Swift Tag	Description	Regd?	Datatype	Merchandising Field
32C - Date and Amount	This field specifies the currency code and amount in a transaction and a corresponding date.	No	Option A- :4!c/[8c]/30x :4!c - Qualifier / - Delimiter [8c] - Issuer Code / - Delimiter 30x - Type	Charges Credited (this is interpreted as a positive amount) Date will be in format YYMMDD The integer part of the Amount must contain at least one digit. A decimal comma ',' is mandatory and is included in the maximum length Lc_activity.amount Number(20,4) Lc_activity.currency_code Varchar2(3) Lc_activity.activity_date Date

Table 6-38 (Cont.) Input File Layout

Swift Tag	Description	Regd?	Datatype	Merchandising Field
32D - Date and Amount	This field specifies the currency code and amount in a transaction and a corresponding date.	No	Option D- 6!n3!a15d 6!n - Date 3!a - Currency 15d - Amount	Charges Debited (this is interpreted as a negative amount) Date will be in format YYMMDD The integer part of the Amount must contain at least one digit. A decimal comma ',' is mandatory and is included in the maximum length Lc_activity.amount Number(20,4) Lc_activity.currency_code Varchar2(3) Lc_activity.activity_date Date

Table 6-38 (Cont.) Input File Layout

Swift Tag	Description	Regd?	Datatype	Merchandising Field
33A - Date and Amount	This field specifies the currency code and amount in a transaction and a corresponding date.	No	Option A- 6!n3!a15d 6!n - Date 3!a - Currency 15d - Amoun	Date, currency, amount of drawing (this is interpreted as a positive amount) Date will be in format YYMMDD The integer part of the Amount must contain at least one digit. A decimal comma ',' is mandatory and is included in the maximum length Lc_activity.amount Number(20,4) Lc_activity.currency_code Varchar2(3) Lc_activity.activity_date Date

Table 6-38 (Cont.) Input File Layout

Swift Tag	Description	Regd?	Datatype	Merchandising Field
33C - Date and Amount	This field specifies the currency code and amount in a transaction and a corresponding date.	No	Option A- 6!n3!a15d 6!n - Date 3!a - Currency 15d - Amount	Date, currency, amount of drawing (this is interpreted as a negative amount) Date will be in format YYMMDD The integer part of the Amount must contain at least one digit. A decimal comma ',' is mandatory and is included in the maximum length. Lc_activity.amount Number(20,4) Lc_activity.currency_code Varchar2(3) Lc_activity.activity_date Date
72 - Sender to Receiver Information	This field specifies instructions or additional information for the Receiver, Intermediary, Account with Institution or Beneficiary Institution.	No	6*35x	Comments Lc_activity.comment Varchar2(2000)
18A - Number of Repetitive Parts	This field specifies the number of times the repetitive part(s)/ sequence(s) directly before or after this field appears in the message.	No	Option A- 5n - Number of Repetitive Parts.	Number of 77E's contained within the file.

I/O Specification

Integration Type	Upload to Merchandising
File Name	Determined by runtime parameter
Integratin Contract	IntCon000055 (input)

Output File Layout

Table 6-39 Output File Layout

Record Name	Field Name	Field Type	Default Value	Description
File Header	File Type Record Descriptor	Char(5)	FHEAD	Identifies file record type
	File Line Identifier	Number (10)	Line number in file	ID of current line being created for output file
	File Type Definition	Char(4)	LCCH	Identifies file as 'Letter of Credit Changes'
	File Create Date	Char(14)	Create date	Current date, formatted to 'YYYYMMDDHH24MISS'
File Detail	File Type Record Descriptor	Char(5)	FDETL	Identifies file record type
	File Line Sequence Number	Number (10)	Line number in file	ID of current line being created for output file
	Bank Letter of Credit Reference ID	Char(16)	SWIFT tag 20	Bank L/C ID
	Order Number	Number (8)	SWIFT tag 21	Contains the order number that is attached to the letter of credit
	Invoice Number	Char (15)	SWIFT tag 23	Identifies the Issuing Bank's invoice number to which the drawdown refers. This field does not correspond to a Merchandising invoice number
	Transaction Number	Char (10)	Null	Identifies the amendment number or actual transaction number assigned by the bank

Table 6-39 (Cont.) Output File Layout

Record Name	Field Name	Field Type	Default Value	Description
	Transaction Code	Char (6)	If the transaction is a Bank Charge – 'B' If the transaction is a Drawdown – 'D'	Identifies the type of transaction that occurred The type is determined by what detail fields are received for the record. If the record contains a 33A this field will get a 'D'. If the record contains either a 32C or 32D this field will get a 'B'
	Amount Sign	Char (1)	SWIFT 33A, 33C SWIFT 32C, 32D	If the record contains a 33A field leave a blank space in this field If the record contains a 33C field this field should contain a '-' If the record contains a 32C field leave a blank space in this field If the record contains a 32D field this field should contain a '-'
	Amount	Number (20)	SWIFT 33A, 33C SWIFT 32C, 32D	Holds the amount of the activity. This field will have 4 implied decimal places If SWIFT 32C or 32D (Bank Charge) contains a value, use the amount from this field If SWIFT 33A or 33C (Drawdown) contains a value, use the amount from this field
	Currency Code	Char (3)	SWIFT 33A, 32C, 32D	Contains the activity's currency code If SWIFT 32C or 32D (Bank Charge) contains a value, use the currency from this field If SWIFT 33A (Drawdown) contains a value, use the currency from this field

Table 6-39 (Cont.) Output File Layout

Record Name	Field Name	Field Type	Default Value	Description
File Trailer	Activity Date	Char (8)	SWIFT 33A, SWIFT 32C, 32D	Holds the date that the activity took place. Formatted to 'YYYYMMDD' If SWIFT 32C or 32D (Bank Charge) contains a value, use the date from this field If SWIFT 33A (Drawdown) contains a value, use the date from this field
	Comments	Char (2000)	SWIFT tag 72	Holds any comments for the activity
	File Type Record Descriptor	Char(5)	FTAIL	Identifies file record type
	File Line Identifier	Number (10)	Sequential number Created by program.	ID of current line being created for output file
	File Record Counter	Number (10)	N/A	This will contain the number of FDETL lines processed

Transportation Upload (tranupld)

Module Name	tranupld.pc
Description	Transportation Upload
Functional Area	Oracle Retail Trade Management
Module Type	Integration
Module Technology	ProC
Catalog ID	RMS140
Wrapper Script	rmswrap_multi_in.ksh

Schedule

Oracle Retail Merchandising Batch Schedule

Design Overview

This program uploads data from trading partners about the transportation of merchandise from the manufacturing site through customs clearance.

Restart/Recovery

The logical unit of work is a valid DTRAN record. The program reads each DTRAN record from the upload file, validates it and processes it. The recommended commit max counter value for this program is 1000 (this value depends on the implementation).

I/O Specification

Integration Type Upload to Merchandising
File Name Determined by runtime parameter
Integration Contract IntCon000177

Input File Layout

Table 6-40 Input File Layout

Record Name	Field Name	Field Type	Default Value	Description
FTRAN	Record descriptor	Char(5)	FTRAN	File head marker
	Line id	Number(10)	0000000001	Unique line id
	File type definition	Char(4)	TRUP	Identifies program as tranupld
	File create date	Char(14)	Current date	YYYYMMDDH HMISS format
DTRAN	Record descriptor	Char(5)	DTRAN	Vessel, Voyage, ETD, Container, BL, Invoice File head
	Line id	Number(10)	N/A	Unique line id
	Partner Type	Char(6)	N/A	Identifies the partner type
	Partner ID	Char(10)	N/A	Identifies the partner id
	Vessel ID	Char(20)	N/A	Identifies the Vessel
	Voyage ID	Char(10)	N/A	Identifies the Voyage or Flight ID
	Estimated Depart Date	Char(8)	N/A	YYYYMMDD format
	Shipment Number	Char (20)	N/A	Identifies an outside Shipment number
Actual Arrival Date	Char(8)	N/A	YYYYMMDD format	

Table 6-40 (Cont.) Input File Layout

Record Name	Field Name	Field Type	Default Value	Description
	Trans Mode	Char(6)	N/A	Identifies the type of transportation being used. Valid values are found in the TRMO Code Type on the CODE_DETAIL table
	Vessel SCAC Code	Char(6)	N/A	Customs defined ID for the Vessel. Validated against SCAC table
	Estimated Arrival Date	Char(8)	N/A	YYYYMMDD format
	Lading Port	Char(5)	N/A	Identifies the Lading Port. Validated against OUTLOC with type = 'LP'
	Discharge Port	Char(5)	N/A	Identifies the Discharge Port. Validated against OUTLOC with type = 'DP'
	Service Contract Number	Char(15)	N/A	Identifies the outside Service Contract Number
	Container id	Char(20)	N/A	Identifies the Container
	Container SCAC code	Char(6)	N/A	Customs defined id for the container. Validated against SCAC table
	Delivery Date	Char(8)	N/A	YYYYMMDD format
	Seal id	Char(15)	N/A	Customs defined id for the container's seal

Table 6-40 (Cont.) Input File Layout

Record Name	Field Name	Field Type	Default Value	Description
	Freight Type	Char(6)	N/A	Code that identifies the container type. Validated against the FREIGHT_TYP E table
	Freight Size	Char(6)	N/A	Code that identifies the container size. Validated against the FREIGHT_SIZ E table
	In Transit No.	Char(15)	N/A	External transit number
	In Transit Date	Char(8)	N/A	YYYYMMDD format
	BL/AWB id	Char(30)	N/A	Identifies the Bill of Lading or Air Way Bill
	Candidate Ind	Char(1)	Defaulted to 'N'	Identifies a complete Transportation record. Valid values are 'Y' and 'N'
DPOIT	Record descriptor	Char(5)	DPOIT	Order/Item detail info
	Line id	Number(10)	N/A	Unique file line id
	ACD_Code	Char(1)	N/A	Determines which process to perform 'A'dd, 'C'hange, 'D'elete.
	Rush Ind	Char(1)	Defaulted to 'N'	Identifies whether or not the item should be on a 'Rush' delivery. Valid values are 'Y' and 'N'
	Order number	Number(8)	N/A	Merchandising order number
	Item	Char(25)	N/A	Merchandising Item number
	Invoice id	Char(30)	N/A	Identifies the Commercial Invoice

Table 6-40 (Cont.) Input File Layout

Record Name	Field Name	Field Type	Default Value	Description
	Invoice date	Char(8)	N/A	YYYYMMDD format
	Currency Code	Char(3)	N/A	Currency that the Currency Amount is reported in. Validated against CURRENCIES table.
	Exchange Rate	Char (20)	N/A	The exchange rate back to the primary currency (10 implied decimals)
	Invoice amt	Char (20)	N/A	Invoice amt*10000 (with 4 implied decimal places), amount charged by supplier for the PO/Item
	Origin Country id	Char(3)	N/A	Identifies where the PO/Item was made
	Consolidation Country id	Char(3)	N/A	Identifies where the PO/Items were consolidated
	Export Country id	Char(3)	N/A	Identifies where the PO/Items where shipped from
	Status	Char(6)	N/A	Identifies the PO/Item status. Valid values are found in the TRCO Code Type on CODE_DETAIL
	Receipt ID	Char(30)	N/A	Identifies the external receipt number
	FCR id	Char(15)	N/A	Identifies the Freight Cargo Receipt id
	FCR date	Char(8)	N/A	YYYYMMDD format

Table 6-40 (Cont.) Input File Layout

Record Name	Field Name	Field Type	Default Value	Description
	Packing Method	Char(6)	N/A	Identifies the Packing Type (Hanging or Flat). Valid values are 'HANG' or 'FLAT'
	Lot Number	Char(15)	N/A	Identifies the Lot Number of the PO/Item
	Item Qty	Number(12)	N/A	Item Qty*10000(with 4 implied decimals), qty of Items
	Item QTY UOM	Char(4)	N/A	Identifies the UOM associated with the item quantity
	Carton QTY	Number(12)	N/A	Carton QTY*10000 (with 4 implied decimals), qty of Cartons
	Carton QTY UOM	Char(4)	N/A	Identifies the UOM associated with the carton quantity
	Gross WT	Number(12)	N/A	Gross WT*10000 (with 4 implied decimals), Gross weight
	Gross WT UOM	Char(4)	N/A	Identifies the UOM associated with the gross weight
	Net WT	Number(12)	N/A	Net WT*10000 (with 4 implied decimals), Net Weight
	Net WT UOM	Char(4)	N/A	Identifies the UOM associated with the net weight

Table 6-40 (Cont.) Input File Layout

Record Name	Field Name	Field Type	Default Value	Description
	Cubic	Number(12)	N/A	Cubic*10000 (with 4 implied decimals), cubic size
	Cubic UOM	Char(4)	N/A	Identifies the UOM associated with the cubic size
	Comments	Char(256)	N/A	User Comments
FTAIL	Record type	Char(5)	FTAIL	N/A
	Line id	Number(10)	N/A	Unique file line id
	No. of lines	Number(10)	N/A	Total number of transaction lines in file (not including FHEAD and FTAIL)

Design Assumptions

N/A

Stock Counts

Merchandising subscribes to data related to stock counts from stores, warehouses, and third-party counters.

The following scheduled inbound integrations are included in this functional area:

- [Conversion of Warehouse Stock Count Results File \(lifstkup\)](#)
- [Upload Stock Count Results from Stores/Warehouses \(stockcountupload.ksh\)](#)

For more on stock count processing, see *Merchandising Operations Guide – Volume 1*.

Conversion of Warehouse Stock Count Results File (lifstkup)

Module Name	lifstkup.pc
Description	Conversion of WMS Stock Count Results File
Functional Area	Stock Counts
Module Type	Integration
Module Technology	ProC
Catalog ID	RMS150
Wrapper Script	batch_lifstkup.ksh

Schedule

Oracle Retail Merchandising Batch Schedule

Design Overview

The Stock Upload Conversion batch is used when WMS sends count information to Merchandising. This batch converts the inventory balance upload file into the format supported by the Stock Count Upload process.

Restart/Recovery

Oracle Retail standard file-based restart/recovery is used. The commit max counter field should be set to prevent excessive rollback space usage, and to reduce the overhead of file I/O. The recommended commit counter setting is 1000 records (subject to change based on implementation).

I/O Specification

Integration Type	Upload to Merchandising
File Name	Determined by runtime parameter
Integration Contract	IntCon000172 (input from WMS) IntCon000102 (output for Merchandising stockcountupload)

Input File Layout

Table 6-41 Input File Layout

Field Name	Field Type	Description
DC_DEST_ID	11 – Number (10) + 1 for trailing space	Unique identifier for the warehouse
TRANSACTION_DATE	15 – Date (14) + 1 for trailing space	Date on which the transaction occurred
ITEM_ID	26 - Varchar2 (25) + 1 for trailing space	Uniquely identifies the item on the count
AVAILABLE_QTY	15 – Number (12) + 1 for leading sign and + 1 for decimal and + 1 for trailing space	Units available for distribution
DISTRIBUTED_QTY	14 – Number (12) + 1 for decimal and + 1 for trailing space	Units distributed include: Units distributed but not yet picked, units picked but not yet manifested, units manifested but not yet shipped
RECEIVED_QTY	15 - Number (12) + 1 for leading sign and + 1 for decimal and + 1 for trailing space	Units received but not put away
TOTAL_QTY	14 – Number (12,4) + 1 for decimal and + 1 for trailing space	Sum of all units that physically exist: container status of: I, D, M, R, T, X

Table 6-41 (Cont.) Input File Layout

Field Name	Field Type	Description
AVAILABLE_WEIGHT	15 – Number (12,4) + 1 for leading sign + 1 for decimal + 1 for trailing space	Weight available for distribution of catch weight items
RECEIVED_WEIGHT	14 – Number (12,4) + 1 for decimal + 1 for trailing space	Weight received but not put away for catch weight items
DISTRIBUTED_WEIGHT	14 – Number (12,4) + 1 for decimal + 1 for trailing space	Weight distributed includes: weight distributed but not yet picked, weight picked but not yet manifested, weight manifested but not yet shipped (value only catch weight items)
TOTAL_WEIGHT	13 – Number (12,4) + 1 for decimal	Sum of all weight that physically exist: container status of: I, D, M, R, T, X. For catch weight items

Output File Layout

Table 6-42 Output File Layout

Record Name	Field Name	Field Type	Default Value	Description
FHEAD	file type record descriptor	Char (5)	FHEAD	Describes the file line type
	file line identifier	Number (10)	0000000001	ID of current line being processed
	file type	Char (4)	'STKU'	Identifies the file type
	stocktake_date	Date (14)	N/A	The date on which the count occurred, formatted as YYYYMMDDHH24MISS
	file create date	Date (14)	N/A	Date on which the file was created, formatted as YYYYMMDDHH24MISS
	cycle count	Number (8)	N/A	stake_head.cycle_count
	Location type	Char (1)	'W'	Will always be 'W', as this process is only executed for warehouse locations

Table 6-42 (Cont.) Output File Layout

Record Name	Field Name	Field Type	Default Value	Description
	location	Number(10)	N/A	Indicates the number of the physical warehouse where the count occurred
FDETL	file type record descriptor	Char(5)	FDETL	Identifies the file line type
	file line identifier	Number(10)	N/A	ID of current line being processed, internally incremented
	Item type	Char(3)	'ITM'	Indicates the type of item that was counted. This will always be 'ITM', indicating a transaction level item
	item value	Char(25)	N/A	The ID of the item that was counted
	inventory quantity	Number(12)	N/A	The total quantity or weight of product counted; includes four implied decimal places
	location description	Char(150)	N/A	Used by Merchandising to determine the location where the item was counted. This program will always leave as NULL
FTAIL	file type record descriptor	Char(5)	FTAIL	Identifies the file line type
	file line identifier	Number(10)	N/A	ID of current line being processed, internally incremented
	file record count	Number(10)	N/A	Indicates the number of detail records

Design Assumptions

N/A

Upload Stock Count Results from Stores/Warehouses (stockcountupload.ksh)

Module Name	stockcountupload.ksh
Description	Upload Stock Count Results from Stores/Warehouses
Functional Area	Stock Count
Module Type	Integration
Module Technology	ksh
Catalog ID	RMS153
Wrapper Script	batch_stockcountupload.ksh

Schedule

Oracle Retail Merchandising Batch Schedule

Design Overview

The purpose of this module is to upload the contents of the stock count file, which contains the results of a count that occurred in a store or warehouse, to staging tables for further processing.

Input/Out Specification

Integration Type	Upload in Merchandising
File Name	Determined by runtime parameter
Integratin Contract	IntCon000102

Input File Layout

Table 6-43 Input File Layout

Record Name	Field Name	Field Type	Default Value	Description
File Header	File head descriptor	Char(5)	FHEAD	Describes file line type
	File line identifier	Number(10)	0000000001	ID of current line being processed
	File Type	Char(4)	STKU	Identifies the file type
	File create date	Char(14)	N/A	Indicates the date the file was created in YYYYMMDDHH 24MISS format
	Stock take date	Char(14)	N/A	Date on which stock count will take place in YYYYMMDDHH MISS format

Table 6-43 (Cont.) Input File Layout

Record Name	Field Name	Field Type	Default Value	Description
	Cycle count	Number (8)	N/A	Unique number to identify the stock count
	Location Type	Char(1)	N/A	Indicates the type of location where the count occurred. Valid values are 'S','W','E'.
	Location	Number(10)	N/A	The location where the stock count occurred
Transaction Record	File record descriptor	Char(5)	FDETL	Describes file line type
	Line Number	Number(10)	N/A	Sequential file line number
	Item type	Char(3)	N/A	Indicates the type of item counted – either transaction level (ITM) or reference item (REF)
	Item value	Char(25)	N/A	Unique identifier for item that was counted
	Inventory quantity	Number(12)	N/A	Total quantity counted for the item at the location formatted with 4 implied decimal places
	Location description	Char(150)	N/A	Description of inventory location (such as,. sales floor, backroom)
FTAIL	File record descriptor	Char(5)	FTAIL	Marks end of file
	File line identifier	Number(10)	N/A	ID of current line being processed, internally incremented
	File record count	Number(10)	N/A	Number of detail records

Design Assumptions

This program uses grep to search log files for errors. The GREP function should point to the /usr/xpg4/bin/ directory instead of /usr/bin directory to utilize the "-E" option. Otherwise, it will fail with an "illegal option" error message.

Franchise

Merchandising subscribes to data related to franchise customers, orders, and returns from order management solutions and other external franchise customer management solutions.

The following scheduled inbound integrations are included in this functional area:

- [Franchise Customer Upload \(fcustomerupload\)](#)
- [Franchise Order Upload \(wfordupld.ksh\)](#)
- [Franchise Return Upload \(wfretupld.ksh\)](#)
- [Upload Cost Buildup Template \(fcosttmplupld\)](#)
- [Upload of Franchise Sales \(wflsupld.ksh\)](#)

For more on franchise processing, see *Merchandising Operations Guide - Volume 1*.

Franchise Customer Upload (fcustomerupload)

Module Name	fcustomerupload.ksh
Description	Franchise Customers Upload
Functional Area	Franchise Management
Module Type	Integration
Module Technology	ksh
Integration Catalog ID	RMS126
Wrapper Script	rmswrap_shell_in.ksh

Schedule

Oracle Retail Merchandising Batch Schedule

Design Overview

This module uploads franchise customers and customer group details from an external system into Merchandising staging tables. It also performs both technical and business validation of the data sent in the file; for example, it validates that a customer cannot be deleted if a franchise store is associated with it.

Restart/Recovery

The restart recovery is different from the conventional Merchandising batch. There are three points on the batch upload process where you can evaluate the successful load of the data.

- SQL load - SQL load dumps invalid records that do not meet certain technical requirements (for example: data type inconsistencies, and so on.). The rejected record is written either to a bad file or to a discard file. The discard file contains records that do not

satisfy conditions such as missing or invalid record types. Records with other technical issues are written to the bad file.

 **Note:**

A non-fatal code is returned by the program and a message will be written to the log file if reject files are created.

Action Required: When such conditions exist, you may update either the bad or discard file and attempt to reload using the same files.

- File-Based Validations - the data from the files are loaded into the staging tables for validation. PL/SQL functions will validate the data in the staging tables to determine if there are any issues with the FHEAD and FTAIL in the file. These kinds of errors are FATAL errors and the batch ends the file processing immediately with return code 255.

Action Required: When this condition exists, you can fix the data upload file and try to reload.

- Business Validation Level - PL/SQL functions determine if the transactions loaded are valid enough to modify the actual Merchandising tables. Records that do not meet certain technical or business validations are rejected and the information is updated back into the staging table with an appropriate error message and the batch issues a NON-FATAL return code 1.

Action Required: When this condition exists, you can fix the data upload file and try to reload.

I/O Specification

Integration Type	Upload to Merchandising
File Name	Determined by runtime parameter
Integration Contract	IntCon000022

Input File Layout

Table 6-44 File Layout

Record Name	Field Name	Field Type	Default Value	Description
File Header	File Record Descriptor	Char(5)	N/A	Identifies file record type. It should be FHEAD
	File Line ID	Number(10)	N/A	ID of current line being processed by input file
	File Type	Char(5)	FCUST	Identifies file as 'Franchise customer upload'

Table 6-44 (Cont.) File Layout

Record Name	Field Name	Field Type	Default Value	Description
	File Create Date	Date	SYSDATE	Date file was written by external system
Transaction Header	File Record Descriptor	Char(5)	N/A	Identifies transaction record type. It should be THEAD
	File Line ID	Number(10)	N/A	ID of current line being processed by input file
	Message Type	Char(30)	N/A	Identifies the action that will be performed on the franchise customer transaction header record. It can be either create (fcustgrpcre) or update (fcustgrpupd) or delete (fcustgrpdel) a franchise customer group
	Franchise Customer group ID	Number(10)	N/A	Customer group ID
	Franchise Customer group Name	Char(120)	N/A	Customer group name. This field is optional for delete
Transaction Detail	File Record Descriptor	Char(5)	N/A	Identifies transaction record type. It should be TDETL
	File Line ID	Number(10)	N/A	ID of current line being processed by input file

Table 6-44 (Cont.) File Layout

Record Name	Field Name	Field Type	Default Value	Description
	Message Type	Char(30)	N/A	Identifies the action that will be performed on the franchise customer transaction detail record. It can be either create (fcustcre) or update (fcustupd) or delete (fcustdel) a franchise customer .
	Franchise Customer ID	Number(10)	N/A	Customer ID to be processed
	Franchise Customer Name	Char(120)	N/A	Customer Name
	Credit Ind	Char(1)	N	This field will determine if the franchise customer has good credit. Valid values are Y and N
	Auto approve Ind	Char(1)	N	To auto approve the externally uploaded orders and returns. Valid values are Y and N
Transaction Trailer	File Record Descriptor	Char(5)	N/A	Identifies file record type. It should be TTAIL
	File Line ID	Number(10)	N/A	ID of current line being processed by input file
	Transaction Record Count	Number(10)	N/A	Number of TDETL records in this transaction set. (total records between THEAD & TTAIL)

Table 6-44 (Cont.) File Layout

Record Name	Field Name	Field Type	Default Value	Description
File Trailer	File Record Descriptor	Char(5)	N/A	Identifies file record type. It should be FTAIL
	File Line ID	Number(10)	N/A	ID of current line being processed by input file.
	File Record Counter	Number(10)	N/A	Number of records/ transactions processed in current file (total records between FHEAD & FTAIL)

Design Assumptions

N/A

Franchise Order Upload (wfordupld.ksh)

Module Name	wfordupld.ksh
Description	Franchise Order Upload
Functional Area	Franchise Management
Module Type	Integration
Module Technology	ksh
Catalog ID	RMS60
Wrapper Script	batch_wfupload.ksh

Schedule

Oracle Retail Merchandising Batch Schedule

Design Overview

This batch program is used to upload franchisee orders from an external source. These orders will be created with an order type of 'EDI' and will be created for the source type specified in the upload file. If source type is not specified, then the costing location for the item/franchise store will be used. Orders will be created in approved status if the customer is setup for auto approval, assuming that the customer has valid credit.

If the customer fails credit check or if available inventory at the source location is insufficient to fulfill the order, the order will be generated in input status.

Franchise orders from customers that are not identified for 'Auto Approval' are uploaded into Merchandising in input status. These orders will need to be manually approved in Merchandising in order to be considered active.

Restart/Recovery

The restart recovery is different from the conventional Merchandising batch. There are two points on the batch upload process where users can evaluate the successful load of the data.

- **SQL load** - At this point, SQL load dumps invalid records that do not meet certain technical requirements (for example: file layout issues, data type inconsistencies, and so on.). The rejected record is written to a bad file or to a discard file. The discard file contains records that do not satisfy conditions, such as missing or invalid record types. Records with other technical issues are written to the bad file.

Note:

A non-fatal code is returned by the program and a message will be written to the log file if reject files are created.

Action Required: When such conditions exist, you may update either the bad or discard file and attempt to reload using the same files.

- **Business Validation** - At this point data from the file(s) are loaded into the staging table(s). PL/SQL functions determine if this loaded data is valid enough to be inserted into the actual Merchandising tables. For records that do not meet certain technical or business validations, the error message will be updated in staging table.

Action Required: When this condition exists, you can fix the data upload file and try to reload the file with valid data.

I/O Specification

Integration Type Download from Merchandising
File Name wford*.dat
Integration Contract IntCon000108

SQL Loader Input File Layout

Table 6-45 SQL Loder Input File Layout

Record Name	Field Name	Field Type	Null allowed?	Default Value	Description
FHEAD	File head descriptor	Char(5)	No	FHEAD	Describes file line type.
	Line Number	Number(10)	No	N/A	Id of the current line being processed.

Table 6-45 (Cont.) SQL Loder Input File Layout

Record Name	Field Name	Field Type	Null allowed?	Default Value	Description
	Customer Id	Number(10)	No	N/A	Customer ID of the customer requesting the order.
	Customer Order Reference number	Char(20)	No	N/A	A reference field used by the customer for their tracking purposes.
	Currency Code	Char(3)	No	N/A	This is the currency on which the order was transacted.
	Default Billing location	Number(10)	No	N/A	A customer's location where the billing for the entire order is sent. If blank, each location is billed.
	Comments	Char(2000)	Yes	N/A	Any other miscellaneous information relating to the order.
FDETL	File record descriptor	Char(5)	No	FDETL	Describes file line type.
	Line Number	Number(10)	No	N/A	Id of the current line being processed.
	Item	Char(25)	No	N/A	The item on the franchise order.
	Customer Location	Number(10)	No	N/A	The franchise store requesting the order.
	Source Loc Type	Char(2)	Yes	N/A	Source location type for which the franchise order has been created. Valid values are ST - Store, WH - warehouse, or SU - Supplier

Table 6-45 (Cont.) SQL Loder Input File Layout

Record Name	Field Name	Field Type	Null allowed?	Default Value	Description
	Source Location	Number(10)	Yes	N/A	Source location for which the franchise order has been created. If the source location is a warehouse then both physical and virtual warehouses are allowed.
	Requested Quantity	Number (12,4)	No	N/A	Number of item units being ordered, includes 4 implied decimal places
	Unit of Purchase	Char(3)	No	N/A	Unit of purchase can be the item's standard unit of measure, case, inners or pallets.
	Fixed Cost	Number (20,4)	Yes	N/A	This is cost which will be charged to the customer for the item on the franchise order; value includes 4 implied decimal places.
	Need Date	Char(11)	No	N/A	Date on which the item is needed in the franchise store, with the following format "DD-MON-YYYY".
	Not After Date	Char(11)	No	N/A	Date after which the item may no longer be accepted for a franchise store, with the following format "DD-MON-YYYY".

Table 6-45 (Cont.) SQL Loder Input File Layout

Record Name	Field Name	Field Type	Null allowed?	Default Value	Description
FTAIL	File record descriptor	Char(5)		FTAIL	Marks end of file.
	Line Number	Number(10)		N/A	Id of current line being processed.
	File record count	Number(10)		N/A	Number of detail records.

Design Assumptions

N/A

Franchise Return Upload (wfretupld.ksh)

Module Name	wfretupld.ksh
Description	Franchise Return Upload
Functional Area	Franchise Management
Module Type	Integration
Module Technology	Ksh
Catalog ID	RMS154
Wrapper Script	batch_wfupload.ksh

Schedule

Oracle Retail Merchandising Batch Schedule

Design Overview

This batch program is used for uploading franchise returns sent from an external source, such as an external order management application. When returns are uploaded in this manner, the data will be validated and the return will be created in Merchandising. Additionally, an associated franchise return transfer will also be created.

Restart/Recovery

The restart recovery is different from the conventional Merchandising batch. There are two points on the batch upload process where users can evaluate the successful load of the data.

- SQL load - At this point, SQL load dumps invalid records that do not meet certain technical requirements (for example: file layout issues, data type inconsistencies, and so on.). The rejected record is written either to a bad file or to a discard file. The discard file contains records that do not satisfy conditions, such as missing or invalid record types. Records with other technical issues are written to the bad file.

 **Note:**

A non-fatal code is returned by the program and a message will be written to the log file if reject files are created. When such conditions exist, you may either update the bad or discard file and attempt to reload using the same files.

- **Business Validation** - At this point data from the file(s) are loaded into the staging table(s). PL/SQL functions determine if this loaded data is valid enough to be inserted into the actual Merchandising tables. For all records that do not meet certain technical or business validations, the error message will be updated in staging table. When this condition exists, you can fix the data upload file and try to reload the file with valid data.

I/O Specification

Integration Type Upload to Merchandising
File Name wfreturn*.dat
Integration Contract Intcon000109

SQL Loader Input File Layout

The following is the file pattern for the upload file.

 **Note:**

The values are pipe "|" delimited and can optionally be enclosed by " ".

Table 6-46 SQL Loader Input File Layout

Record Name	Field Name	Field Type	Null Allowed?	Default Value	Description
FHEAD	File head descriptor	Char(5)	No	FHEAD	Describes file line type.
	Line Number	Number(10)	No		Id of the current line being processed.
	Customer ID	Number(10)	No		Franchise customer ID of the customer making the return.
	Customer Return Reference number	Char(20)	No		A reference field used by the franchise customer for their tracking purposes.

Table 6-46 (Cont.) SQL Loader Input File Layout

Record Name	Field Name	Field Type	Null Allowed?	Default Value	Description
	Currency Code	Char(3)	No		This is the return currency.
	Comments	Char(2000)	Yes		Any other miscellaneous information related to the return.
FDETL	File record descriptor	Char(5)	No	FDETL	Describes file line type.
	Line Number	Number(10)	No	N/A	Id of the current line being processed.
	Item	Char(25)	No	N/A	The item on the franchise return.
	Franchise Order Number	Number(10)	No	N/A	The franchise order number against which the return is made.
	Customer Location	Number(10)	No	N/A	The franchise location which is making the return.
	Return Loc Type	Char(1)	No	N/A	Return location type for the franchise return; valid values are S - store or W - warehouse.
	Return Location	Number(10)	No	N/A	Return location for the franchise return.
	Return Method	Char(1)	No	N/A	The type of return; valid values are: -R-Return to Store/ Warehouse -D-Destroy at site
	Unit of measure	Char(3)	No	N/A	The unit measure of the return quantity. This is assumed to be the items standard UOM.

Table 6-46 (Cont.) SQL Loader Input File Layout

Record Name	Field Name	Field Type	Null Allowed?	Default Value	Description
	Return qty	Number(12,4)	No	N/A	The quantity of item to be returned
	Return Reason	Char(6)	No	N/A	Return reason code; valid values are found on the CODE_DETAIL table where CODE_TYPE is 'RTVR'.
	Return unit cost	Number(20,4)	Yes	N/A	The per unit cost for the return.
	Restock Type	Char(1)	No	N/A	Indicates how the restocking fee will be calculated per item; valid values are S-specific or V-value.
	Restock Fee	Number(20,4)	No	N/A	Unit restocking fee.
FTAIL	File record descriptor	Char(5)	No	FTAIL	Marks end of file.
	Line Number	Number(10)	No	N/A	Id of current line being processed.
	File record count	Number(10)	No	N/A	Number of detail records.

Design Assumptions

N/A

Upload Cost Buildup Template (fcostmplupld)

Module Name	fcostmplupld.ksh
Description	Upload Cost Buildup Template
Functional Area	Franchise Management
Module Type	Integration
Module Technology	ksh
Catalog ID	RMS125
Wrapper Script	rmswrap_shell_in.ksh

Schedule

Oracle Retail Merchandising Batch Schedule

Design Overview

This module uploads cost buildup templates and franchise cost relationships used for franchise pricing from an external system into Merchandising staging tables. It also performs both technical and business validation of the data sent in the file; for example, it validates that start and end dates are included for new and updated templates.

 **Note:**

No date format is specified in the input file, as any valid PL/SQL date format can be used.

Restart/Recovery

The restart recovery is different from the conventional Merchandising batch. There are three points on the batch upload process where users can evaluate the successful load of the data.

- SQL load - SQL load dumps invalid records that do not meet certain technical requirements (for example: file layout issues, data type inconsistencies, and so on.). The rejected record is written either to a bad file or to a discard file. The discard file contains records that do not satisfy conditions such as missing or invalid record types. Records with other technical issues are written to the bad file.

 **Note:**

A non-fatal code is returned by the program and a message will be written to the log file if reject files are created

Action Required: When such conditions exist, you may update either the bad or discard file and attempt to reload using the same files.

1. Business Validation Level - the data from the files are loaded into the staging tables for validation. PL/SQL functions determine if this loaded data is valid enough to be inserted into the actual Merchandising tables. Records that do not meet certain technical or business validations are rejected and the information is updated back into the staging table with an appropriate error message and the batch issues a NON-FATAL return code.

Action Required: When this condition exists, you can fix the data upload file and try to reload.

2. Chunking validated data - At this point the data from staging tables that have passed business validation are chunked based on the number of valid transactions (cost templates) and max_chunk_size from RMS_PLSQL_BATCH_CONFIG table. If there are no valid transactions to be chunked, batch issues a FATAL return code.

Action Required: When this condition exists, you can fix the data upload file and try to reload.

I/O Specification

Integration Type Upload to Merchandising
File Name Determined by runtime parameter
Integration Contract IntCon000021

SQL Loader Input File Layout

Table 6-47 SQL Loader Input File Layout

Record Name	Field Name	Field Type	Default Value	Description
File Header	File Type Record Descriptor	Char(5)	N/A	Identifies file record type. Valid value is FHEAD.
	File Line Identifier	Number(10)	N/A	Sequential file line number
	File Type Definition	Char(5)	CTMPL	Identifies file as 'Cost Template Upload'
	File Create Date	Date	SYSDATE	Date on which the file was created by external system
Transaction Header	File Record Descriptor	Char(5)	N/A	Identifies transaction header record type. Valid value is THEAD
	File Line Identifier	Number(10)	N/A	Sequential file line number

Table 6-47 (Cont.) SQL Loader Input File Layout

Record Name	Field Name	Field Type	Default Value	Description
	Message Type	Char(30)	N/A	Identifies the action that will be performed on the franchise cost template header information that is provided as part of this record It can be either create or update or delete a franchise cost template. Valid message types are: costtmpadd (for additions), costtmpmod (for updates), costtmpdel (for deletions)
	Template ID	Number(10)	N/A	Template ID
	Template Description	Char(120)	N/A	Template Description
	Template Type	Char(1)	N/A	Indicates the type of the template. Valid values are M = Margin then Up-Charge, U = Up-charges, then Margin, R = % of Retail and C = Cost
	Percentage	Number(12,4)	N/A	Margin percent or % off Retail value; required if template type is M, U and R types of templates

Table 6-47 (Cont.) SQL Loader Input File Layout

Record Name	Field Name	Field Type	Default Value	Description
Transaction Detail	Cost	Number(20,4)	N/A	Indicates the franchise cost for an item when template type is 'C' This is mandatory and should only be populated if template type is 'C'
	Final Cost	Char(1)	N/A	Signifies if the cost is final or acquisition. Valid values are 'Y' or 'N'
	File Record Descriptor	Char(5)		Identifies transaction detail record type. Valid value is TDETL
	File Line Identifier	Number(10)		Sequential file line number
	Message Type	Char(30)		Identifies the action that will be performed on the franchise cost template relationship information that is provided as part of this record. It can be either create or update or delete a cost relationship. Valid values are: costtmpreladd (for additions), costtmprelmod (for updates), costtmpreldel (for deletions)
	Dept	Number(4)		Department associated with the cost template

Table 6-47 (Cont.) SQL Loader Input File Layout

Record Name	Field Name	Field Type	Default Value	Description
	Class	Number(4)		Class associated with the cost template
	Subclass	Number(4)		Subclass associated with the cost template
	Item	Char(25)		Unique number that identifies a valid item associated with the template. Used for template types of 'C' only
	Location	Number(10)		Franchise Store Number associated with the template
	Start Date	Date		Date on which a cost template will be effective for the subclass/item and franchise store (required for update and delete of a cost relationship)
	End Date	Date		Date on which a cost template will expire for a subclass/item and franchise store (required for update and delete of a cost relationship)
	New Start Date	Date		New Date on which a franchise cost relationship will be effective
	New End Date	Date		New Date on which a franchise cost relationship will expire

Table 6-47 (Cont.) SQL Loader Input File Layout

Record Name	Field Name	Field Type	Default Value	Description
	Cost Component ID	Char(10)		Unique code which signifies the up-charge cost component when First_Applied is 'U' This should only be populated if First Applied is 'U'
Transaction Trailer	File Record Descriptor	Char(5)	N/A	Identifies transaction trailer record type. Valid value is TTAIL
	File Line Identifier	Number(10)	N/A	Sequential file line number
	Transaction Record Counter	Number(10)	N/A	Number of TDETL records in this transaction set
File Trailer	File Record Descriptor	Char(5)	N/A	Identifies file trailer record type. Valid value is TTAIL
	File Line Identifier	Number(10)	N/A	Sequential file line number
	File Record Counter	Number(10)	N/A	Number of records/ transactions processed in current file (only records between FHEAD & FTAIL)

Design Assumptions

N/A

Upload of Franchise Sales (wflsupld.ksh)

Module Name	wflsupld.ksh
Description	Upload of Franchise Sales to Merchandising
Functional Area	Franchise Management

Module Type	Integration
Module Technology	ksh
Catalog ID	RMS156
Wrapper Script	batch_wfslsupld.ksh

Schedule

Oracle Retail Merchandising Batch Schedule

Design Overview

Non-stockholding franchise stores in Merchandising are used for retailers who have franchise or other business customers for whom they supply inventory, but don't manage it for them. However, even though inventory information will not be available for these locations in Merchandising, sales information will be able to be uploaded to Merchandising via this process to allow retailers to have better visibility to future demand from these customers. In addition to uploading sales information, this same batch script also purges old non-stockholding franchise store sales records from Merchandising. The script runs in 4 modes:

- **Load** - this mode will load the data from the file into a staging table in Merchandising for processing; any errors encountered in validating the data on the upload are also written to the staging table (WFSLSUPLD_STAGING).
- **Process** - this mode will process the records in the staging table that did not have errors during load, which includes both writing the data to the WF_NONSTOCKHOLDING_SALES table, as well as purging the processed records from the staging table.
- **Reject** - this mode will process the records on the staging table that had errors on initial load. It will create a reject file for each location/report date with the data in error for that location/date. The records will then be deleted from the staging table.
- **Purge** - this mode is used to purge old sales records from the WF_NON_STOCKHOLDING_SALES table. Records are deleted based on the system parameter Non-stockholding Franchise Sales History days (WF_NON_STOCK_SALES_HIST_DAYS).

Restart/Recovery

The program can be restarted by running the wfslsupld REJECT mode to create an input file of rejected records and wfslsupld LOAD/PROCESS mode to reprocess the rejected records.

I/O Specification

Integration Type	Upload to Merchandising
File Name	Input file name is a parameter during runtime
Integration Contract	IntCon000111

Input File Layout

Table 6-48 Input File Layout

Record Name	Field Name	Field Type	Default Value	Description
FHEAD	Record descriptor	Char(5)	FHEAD	Identifies the file record type
	File Line Id	Char(10)	N/A	Sequential file line number
	File type definition	Char(4)	WFSU	Identifies the file type
	Customer Location	Number(10)	N/A	Store number identifier for the customer location
	Report Date	Char(14)	N/A	Report date of the file in YYYYMMDDH HMMSS format
	File Create Date	Char(14)	N/A	File Create Date in YYYYMMDDH HMMSS format
FDETL	Record descriptor	Char(5)	FDETL	Identifies the file record type
	File Line Id	Char(10)	N/A	Sequential file line number
	Item	Char(25)	N/A	Item number identifier
	Net Sales Quantity	Number(12)	N/A	Sales Quantity with 4 implied decimal places
	Net Sales Quantity UOM	Char(4)	N/A	Unit of Measure for the Net Sales Quantity
	Total Retail Amount	Number(20)	N/A	Total Retail Amount with 4 implied decimal places
	Total Retail Amount Currency	Char(3)	N/A	Currency code for the Total Retail Amount
FTAIL	Record descriptor	Char(5)	FTAIL	Identifies the file record type
	File Line Id	Number(10)	N/A	Sequential file line number

Table 6-48 (Cont.) Input File Layout

Record Name	Field Name	Field Type	Default Value	Description
	File Record counter	Number(10)	N/A	Number of records/ transactions processed in current file (only records between head & tail)

Design Assumptions

N/A

Other Inventory

Other inventory related, scheduled inbound integrations include:

- [External Transaction Data Upload \(trandataload.ksh\)](#)
- [Upload and Process Inventory Reservations from Sales Audit \(ordinvupld\)](#)
- [Upload Item Availability for Type A & D Contracts from Suppliers \(ediupavl\)](#)

External Transaction Data Upload (trandataload.ksh)

Module Name	trandataload.ksh
Description	External Transaction Data Upload
Functional Area	Finance
Module Type	Integration
Module Technology	KSH
Catalog ID	RMS 376
Wrapper Script	batch_trandataload.ksh

Schedule

Oracle Retail Merchandising Batch Schedule

Design Overview

This process, along with trandataprocess.ksh, provides a mechanism to write records directly into the TRAN_DATA tables based on a file from an external system. The primary purpose of this functionality is to allow additional costs to be included in stock ledger valuation that cannot be included based on existing Merchandise functionality. Records written to the TRAN_DATA tables do not necessarily have a connection to any Merchandising transaction, and are based on a determination made outside of Merchandising. The records written through this mechanism function exactly the same as records written by normal Merchandising processes. For cost based transactions, the information must be passed at an

item/location level. For retail-based transactions, it can be at either an item/location or subclass/location level.

 **Note:**

There is no support for recalculating or impacting unit inventory in Merchandising based on the transactions passed in, and only cost or retail value in the stock ledger is impacted - although the weighted average cost (WAC) may also be impacted if that method of accounting is used in Merchandising

The trandataload script loads the staging table STAGE_EXT_TRAN_DATA table from a flat file using SQL Loader and divides the data into chunks to be processed in parallel threads based on the commit_max_counter and num_threads value on RESTART_CONTROL table.

This script accepts the following input parameters:

- Database Connect string
- File load indicator – This indicator is passed as Y if a flat file has to be loaded into the table STAGE_EXT_TRAN_DATA else its N
- Input file – This is the path of the input file. This is mandatory when File load indicator is Y.

The SQL loading from a flat file is optional in the script. If File load indicator is Y the program validates if the input file exists and logs an error in case the input file does not exist. The SQL Load (sqlldr) process loads the input file using control file - trandataload.ctl into the STAGE_EXT_TRAN_DATA table.

- A fatal error from sqlldr will halt the process.
- Rejected records are a non-fatal error and loader will continue processing and create bad file and discard files in case the input file does not match the expected format.

If you chose not to load data into the staging table (File load indicator 'N') then the batch assumes that data has been loaded on the staging table from a different source. After the loading process is complete, the batch divides the data into chunks. If the staging table is empty or all the records are in 'P'rocessed status then the batch logs an appropriate error.

Chunking Logic

- Dense rank the staged records over Subclass, item and location.
- Divide the rank value by the commit max counter.
- Rounding the divided value gives the Chunk ID to which the particular value belongs to.
- Item can be NULL on the staging table, when NULL consider item to be '-999'.
- This will make sure the records with same subclass value and having item as NULL and NOT NULL are not grouped together in a chunk.

Since records with item have to be processed differently, (WAC recalculation and Variance postings) the batch makes sure that they fall in a different chunk to those records which do not have item value.

The Chunk data is inserted into STAGE_EXT_TRAN_DATA_CHUNK table.

Restart/Recovery

N/A

I/O Specification - Input File Specification

This batch uses SQL Loader to populate the staging table. The input file should be in pipe delimited format. Sample record structure would look like:

```
<item>|<dept>|<class>|<subclass>|<location>|<loc_type>|<tran_date>|<tran_code>|
<adj_code>|<units>|<total_cost>|<total_retail>|<ref_no_1>|<ref_no_2>|<GL_ref_no>|
<Old_unit_retail>|<New_unit_retail>|<Sales_type>|<VAT_rate>|<av_cost>|<ref_pack_no>|
<total_cost_excl_elc>|<WAC_reclucate_ind>|<status>|<create_timestamp>|
```

File Layout

The table below specifies the detail of each field in the record.

Table 6-49 File Layout

Field Name	Field Type	Default Value	Description
Item	VARCHAR2(25)	N/A	Item is an optional field. Transactions can be uploaded at the Subclass level also.
Dept	NUMBER(4)	N/A	Mandatory Field
Class	NUMBER(4)	N/A	Mandatory Field
Subclass	NUMBER(4)	N/A	Mandatory Field
Location	NUMBER(10)	N/A	Mandatory Field
Loc_type	VARCHAR2(1)	N/A	Valid values - 'S', 'W', 'E'
Tran_data	DATE	N/A	Mandatory Field
Tran_code	NUMBER(2)	N/A	Mandatory Field
Adj_code	VARCHAR2(1)	N/A	Valid values - 'C', 'U', 'A'
Units	NUMBER(12, 4)	N/A	Mandatory Field
Total_cost	NUMBER(20, 4)	N/A	N/A
Total_retail	NUMBER(20, 4)	N/A	N/A
Ref_no_1	NUMBER(10)	N/A	N/A
Ref_no_2	NUMBER(10)	N/A	N/A
GL_ref_no	NUMBER(10)	N/A	N/A
Old_unit_retail	NUMBER(20, 4)	N/A	N/A
New_unit_retail	NUMBER(20, 4)	N/A	N/A
Pgm_name	VARCHAR(100)	N/A	N/A

Table 6-49 (Cont.) File Layout

Field Name	Field Type	Default Value	Description
Sales_type	VARCHAR2(1)	N/A	Valid values - 'C', 'R', 'P'
Vat_rate	NUMBER(12, 4)	N/A	N/A
Av_cost	NUMBER(20, 4)	N/A	N/A
Ref_pack_no	VARCHAR2(25)	N/A	N/A
Total_cost_excl_elc	NUMBER(20, 4)	N/A	N/A
Wac_recalculate_ind	VARCHAR2(1)	N/A	If Weighted Average Cost of the Item-Location should be recalculated after uploading this transaction then this value should be passed as 'Y'.
Status	VARCHAR2(1)	'N'	This value will be defaulted to 'N' by this program. It will be updated to 'P' once it has been processed else to 'E' in case of Error.
Create_timestamp	DATE	Sysdate	N/A

Design Assumptions

N/A

Upload and Process Inventory Reservations from Sales Audit (ordinvupld)

Module Name	ordinvupld.pc
Description	Upload and Process Inventory Reservations from Sales Audit
Functional Area	RMS
Module Type	Integration
Module Technology	ProC
Catalog ID	RMS113
Wrapper Script	batch_ordinvupld.ksh

Schedule

Oracle Retail Merchandising Batch Schedule

Design Overview

This batch program processes the input file generated by the Sales Audit Inventory Export batch, which is generated to reserve and un-reserve inventory based on in-store customer orders and layaway. An in-store customer order is one where the

customer is purchasing inventory present in the store, but will not take it home immediately. For example, with a large item like a sofa, the customer may pickup at a later time with a larger vehicle. Layaway is when a customer pays for an item over time and only takes the item home once it has been fully paid for. In processing this file, Merchandising updates the quantity of the item/location sent to either add or subtract from the quantity in the Customer Order inventory status type.

Restart/Recovery

The logical unit of work for this batch program is a valid item status transaction at a given store/location. The logical unit of work is defined as a group of these transaction records. The Oracle Retail standard file-based restart/recovery logic is used. Records are committed to the database when the maximum commit counter is reached.

I/O Specification

Integration Type	Upload to Merchandising
File Name	Determined by runtime parameter
Integration Contract	IntCon000049

Input File Layout

Table 6-50 ordinuapl.pc - Input File

Record Name	Field Name	Field Type	Default Value	Description
FHEAD	Record descriptor	Char(5)	FHEAD	Identifies the file record type
	File Line Id	Char(10)	0000000001	Sequential file line number
	File type definition	Char(4)	ORIN	Identifies the file type
	File Create Date	Char(14)	N/A	File Create Date in YYYYMMDDHHMMSS format
	Location	Number(10)	N/A	Store location number
THEAD	Record descriptor	Char(5)	THEAD	Identifies the file record type
	File Line Id	Char(10)	N/A	Sequential file line number
	Transaction Date & Time	Char(14)	Transaction Date	Date and time of the order processed
	Transaction Type	Char(6)	'SALE'	Transaction type code specifies whether the transaction is sale or Return
TDETL	Record descriptor	Char(5)	TDETL	Identifies the file record type

Table 6-50 (Cont.) ordinvupld.pc - Input File

Record Name	Field Name	Field Type	Default Value	Description
	File Line Id	Char(10)	N/A	Sequential file line number
	Item Type	Char(3)	REF or	Can be REF or ITM
	Item	Char(25)	ITM	Id number of the ITM or REF
	Item Status	Char(6)	LIN - Layaway Initiate LCA - Layaway Cancel LCO - Layaway Complete PVLCO - Post void of Layaway complete ORI - Pickup/delivery Initiate ORC - Pickup/delivery Cancel ORD - Pickup/delivery Complete PVORD - Post void of Pick-up/delivery complete	Type of transaction
	Dept	Number(4)	N/A	Department of item sold or returned
	Class	Number(4)	N/A	Class of item sold or returned.
	Sub class	Number(4)	N/A	Subclass of item sold or returned
	Pack Ind	Char(1)	N/A	Pack indicator of item sold or returned
	Quantity Sign	Char(1)	'P' or 'N'	Sign of the quantity.
	Quantity	Number(12)	N/A	Quantity * 10000 (4 implied decimal places), number of units for the given order (item) status
	Selling UOM	Char(4)	N/A	UOM at which this item was sold

Table 6-50 (Cont.) ordinvupld.pc - Input File

Record Name	Field Name	Field Type	Default Value	Description
	Catchweight Ind	Char(1)	N/A	Indicates if the item is a catchweight item. Valid values are Y or NULL
	Customer Order number	Char(48)	N/A	Customer Order number
	Posting Store	Number(10)		Contains the store at which the item reservation/ reservation/ reservation cancellation should occur in case of cross-store transactions happening at co-located stores. It is expected that this field will be populated only for items that are to be reserved (or have the reservation canceled) at a different store from the one at which the checkout happened.
TTAIL	File Type Record Descriptor	Char(5)	TTAIL	Identifies file record type
	File Line Identifier	Number(10)	Specified by Sales Audit	ID of current line being processed by input file.
	Transaction count	Number(6)	Specified by Sales Audit	Number of TDETL records in this transaction set
FTAIL	File Type Record Descriptor	Char(5)	FTAIL	Identifies file record type
	File Line Identifier	Number(10)	Specified by external system	ID of current line being processed by input file
	File Record Counter	Number(10)	N/A	Number of records/ transactions processed in current file (only records between FHEAD & FTAIL)

Design Assumptions

N/A

Upload Item Availability for Type A & D Contracts from Suppliers (ediupavl)

Module Name	ediupavl.pc
Description	Upload Item Availability for Type A & D Contracts from Suppliers
Functional Area	EDI - Contracts
Module Type	Integration
Module Technology	ProC
Catalog ID	RMS50
Wrapper Script	rmswrap_in_rej.ksh

Schedule

See Oracle Merchandising Batch Schedule.

Design Overview

This module runs to upload supplier availability information, which is a list of the items that a supplier has available. This information is used by Merchandising for type A and D contracts which require supplier availability information. The data uploaded is written to the SUP_AVAIL table.

Restart/Recovery

N/A

I/O Specification

Integration Type	Upload to Merchandising
File Name	Determined by runtime parameter
Integration Contract	IntCon000016

Input File Layout

Table 6-51 ediupavl.pc - File Layout

Record Name	Field Name	Field Type	Default Value	Description
FHEAD	Record descriptor	Char(5)	FHEAD	Describes file line type
	Line number	Number(10)	0000000001	Sequential file line number
	File type	Char(4)	SPAV	N/A

Table 6-51 (Cont.) ediupavl.pc - File Layout

Record Name	Field Name	Field Type	Default Value	Description
	Create date	Char(14)	N/A	File create date in YYYYMMDDH H24 MISS format
FDETL	Record descriptor	Char(5)	FDETL	Describes file line type
	Line number	Number(10)	N/A	Sequential file line number
	Transaction number	Number(14)	N/A	Sequential transaction number
	Supplier	Number(10)	N/A	Indicates the supplier for whom the data applies
	Item type	Char(3)	N/A	Indicates the type of item contained in the file. Valid types are 'ITM', 'UPC', or 'VPN'
	Item id	Char(25)	N/A	Unique ID for the item
	Item supplement	Char(5)	N/A	UPC supplement
	Available quantity	Number(12)	N/A	Available quantity including 4 implied decimal places
FTAIL	Record descriptor	Char(5)	FTAIL	Number(10)
	Line number	Number(10)	N/A	Sequential file line number (total # lines in file)
	Number of detail records	Number(10)	N/A	Number of FDETL lines in file

Design Assumptions

This module will only be run if contracting is turned on in the system.

Sales Processing

Merchandising and Sales Audit subscribe to data from point of sale (POS) and order management (OMS) solutions related to sales, returns, customer pick-ups, and so on. Generally, sales are first audited in Sales Audit and then sent to Merchandising for posting and inventory updates. However, customers can choose to bypass Sales Audit if using an external auditing solution or choosing not to audit sales data by sending data directly to Merchandising.

This section has been broken into the following sub-sections:

- [Sales Audit](#)
- [Sales Posting](#)

Sales Audit

The purpose of Sales Audit is to accept transaction data from point-of-sale (POS) and order management (OMS) solutions and move the data through a series of processes that culminate in "clean" data. Data that Sales Audit finds to be inaccurate is brought to the attention of the auditors who can use the features in Sales Audit to correct the exceptions.

For more information on Sales Audit processing see *Merchandising Operations Guide Volume 1*.

This chapter contains details about the following integration processes used to import data to Sales Audit:

- [Customer Engagement Promotion Import \(CePromoBatch.ksh\)](#)
- [Import of Unaudited Transaction Data from POS to Sales Audit \(saimptlog/saimptlogi\)](#)
- [Import Total Value Adjustments From External Systems \(saimpadj\)](#)
- [Sales Audit Voucher Upload \(savouch\)](#)

Customer Engagement Promotion Import (CePromoBatch.ksh)

Module Name	CePromoBatch.ksh
Description	Invokes the Customer Engagement Promotion web service to fetch the promotions and saves it to the CE promo tables that will be used by the sagetref module.
Functional Area	Sales Audit
Module Type	Integration
Module Technology	Unix Script, Java, PL/SQL
Catalog ID	N/A
Wrapper Script	N/A

Schedule

Oracle Retail Merchandising Batch Schedule

Design Overview

The purpose of this script is to call the batch client that will execute the Oracle Retail Customer Engagement (ORCE) Promotion web service. The values retrieved from the web service will populate the CE_PROMO and CE_PROMO_DEAL. These tables will be used by the Get Reference Data for Sales Audit Import Processing (sagetref) batch.

The list of valid promotions in ORCE will be extracted by using the retrieve promotions method under the Promotion Event Web Service offered by ORCE. This request will return promotion information including the promotion ID. Similarly the promotion component details in ORCE can be retrieved using the retrieve promotion deals method under the Promotion Event Web Service. The information returned through this request will include the deal ID. The data in the CE_PROMO and CE_PROMO_DEAL tables will be extracted into the promotion file format by the sagetref batch and will be used to validate the RTLOG files being imported into Sales Audit.

The following credentials are entered in the Oracle Enterprise Manager:

```
map="oracle.retail.apps.resa.restservices"
key="orce_promotions "
user="rel:sadiq" <user provided by ORCE for the promotion service>
password="oretail" <password provided by ORCE for the promotion service>
desc="orce_promotions password"
```

Restart/Recovery

N/A

Key Tables Affected

Table	Select	Insert	Update	Delete
CE_PROMO	No	Yes	No	Yes
CE_PROMO_DEAL	No	Yes	No	Yes
RETAIL_SERVICE_REPORT_URL	Yes	No	No	No

Design Assumptions

- The CE promotion web service URL will be different for each deployment.
- The URL update in the RETAIL_SERVICE_REPORT_URL table would be a direct update by AMS team as for RFI integration it is very customer specific and a one-time update (like before) at deployment time

Import of Unaudited Transaction Data from POS to Sales Audit (saimptlog/saimptlogi)

Module Name	saimptlog.c saimptlogi.c
Description	Import of Unaudited Transaction data from POS to Sales Audit
Functional Area	Oracle Retail Sales Audit
Module Type	Integration
Module Technology	ProC

Catalog ID	RSA11a RSA11b
Wrapper Script	batch_saimptlogi.ksh

Schedule

Oracle Retail Merchandising Batch Schedule

Design Overview

Importing POS and Order Management System (OMS) data to Sales Audit is a five or six-step process depending on whether `saimptlogi` or `saimptlog` is used. `Saimptlog` produces SQL*Loader files while `saimptlogi` does inserts directly into the database. `Saimptlogi` is meant for use in a trickle feed environment.

To import POS and OMS data, perform the following:

1. SAGETREF must be run to generate the current reference files:

- Items
- Wastage
- Sub-transaction level items
- Primary variant relationships
- Variable weight PLU
- Store business day
- Code types
- Error codes
- Store POS
- Tender type
- Merchant code types
- Partner vendor
- Supplier vendors
- Employee ids
- Banner ids
- Currency File
- Promotions File
- Warehouse File
- Inventory Status File

These files are all used as input to `SAIMPTLOG` and `SAIMPTLOGI`. Because `SAIMPTLOG` and `SAIMPTLOGI` can be threaded, this boosts performance by limiting interaction with the database.

2. Either `SAIMPTLOG` or `SAIMPTLOGI` must be run against each file. The files are the transaction log files in an Oracle Retail compatible format called RTLOG. The retailer is responsible for converting its transaction logs to RTLOGs. Both

SAIMPTLOG and SAIMPTLOGI create a write lock, depending on the locking level specified in the Sales Audit System Options. It will create a write lock for a store/day combination on Sales Audit tables if the locking level indicated is Store Day. Otherwise, it will create a write lock for a transaction on Sales Audit tables if the locking level indicated is transaction. It will then set the data_status to loading until SAIMPTLOGFIN is executed. SAIMPTLOG generates distinct SQL*Loader files for that store/day for the sa_tran_head, sa_tran_head_attrib, sa_tran_item, sa_tran_item_attrib, sa_tran_disc, sa_tran_disc_attrib, sa_tran_igtax (item Level Tax not VAT), sa_tran_igtax_attrib (item Level Tax Attribute not VAT Attribute), sa_tran_payment (Payment details), sa_tran_tax, sa_tran_tax_attrib, sa_tran_tender, sa_tran_tender_attrib, sa_error, sa_customer, sa_cust_attrib, sa_tran_write_lock and sa_missing_tran tables, whereas SAIMPTLOGI inserts data to the database directly. Both produce an Oracle Retail formatted voucher file for processing.

3. SQL*Loader is executed to load the transaction tables from the files created by SAIMPTLOG. The store/day SQL*Loader files can be concatenated into a single file per table to optimize load times. Alternatively, multiple SQL*Loader files can be used as input to SQL*Loader. SQL*Loader may not be run in parallel with itself when loading a table. Header data (primary keys) must be loaded before ancillary data (foreign keys). This means that the sa_tran_head table must be loaded first, sa_tran_item before sa_tran_disc, and sa_customer before sa_cust_attrib. The main tables of each attribute table must be loaded first, before its attributes. This means that the sa_tran_item must be loaded first, before sa_tran_item_attrib; the sa_tran_disc must be loaded first, before sa_tran_disc_attrib; the sa_tran_igtax must be loaded first, before sa_tran_igtax_attrib; the sa_tran_tender must be loaded first, before sa_tran_tender_attrib; the sa_tran_tax must be loaded first, before sa_tran_tax_attrib. The remaining tables may be loaded in parallel.
4. SAVOUCH is executed to load each of the voucher files in Oracle Retail standard formatted. SAVOUCH may not be multi-threaded.
5. SAIMPTLOGFIN is executed to populate the sa_balance_group table, cancel post voided transactions and vouchers, validate missing transactions, and to mark the import as either partially or fully complete loaded. SAIMPTLOGFIN may not be multi-threaded.

 **Note:**

This design covers only Steps 2 and 3.

Restart/Recovery

N/A

File Upload Error Handling

For each RTLOG file, a record is written to the FILE_UPLOAD_STATUS table. In cases where a non-fatal error occurs after validating a record in the file, the error is written to the error file and a corresponding record is also inserted to the FILE_UPLOAD_ERRORS table.

I/O Specification

Integration Type	Upload to Sales Audit
File Name	Determined by runtime parameter

Integration Contract

Inputs from sagetref.pc:

IntCon000113 (itemfile)
IntCon000114 (wastefile)
IntCon000115 (refitemfile)
IntCon000116 (primvariantfile)
IntCon000117 (varupcfile)
IntCon000118 (storedayfile)
IntCon000119 (promfile)
IntCon000120 (codesfile)
IntCon000121 (errorfile)
IntCon000122 (storeposfile)
IntCon000123 (tendertypefile)
IntCon000124 (merchcodesfile)
IntCon000125 (partnerfile)
IntCon000126 (supplierfile)
IntCon000127 (employeeefile)
IntCon000128 (bannerfile)
IntCon000129 (promfile)
IntCon000130 (whfile)
IntCon000131 (invstatusfile)

Inputs from POS:

IntCon000048 (RTLOG)

Outputs (if using saimptlog SQL Loader Option –
note that saimptlogi inserts directly to Sales Audit
tables and does not create these output files)

IntCon000160 (SAVO)
IntCon000161 (satdisc.ctl)
IntCon000162 (saigtax.ctl)
IntCon000163 (sacust.ctl)
IntCon000164 (sathead.ctl)
IntCon000165 (satitem.ctl)
IntCon000166 (sattend.ctl)
IntCon000167 (satypmt.ctl)
IntCon000168 (samisstr.ctl)
IntCon000169 (sattax.ctl)
IntCon000170 (sacustatt.ctl)
IntCon000171 (saerror.ctl)
IntCon000172 (sathatt.ctl)
IntCon000173 (saitatt.ctl)
IntCon000174 (saidatt.ctl)
IntCon000175 (saixatt.ctl)
IntCon000176 (satxatt.ctl)
IntCon000177 (sattatt.ctl)
(satwritelock.ctl)

The input files for this program are reference files generated by sagetref.pc and RTLOGs. Refer to the details for the sagetref.pc program for the input file specifications.

Output File Layout

Table 6-52 File Name: SAVO (Sales Audit Voucher File)

Record Name	Field Name	Field Type	Default Value	Description
FHEAD	File Type Record Descriptor	Char(5)	FHEAD	File type Record Descriptor
	SA File Line No	Char(10)		Sales Audit File Line No.
	Translator Id	Char(5)	SAVO	Identifies transaction type
	Sys Date	Char(14)		System date in YYYYMMDDHHMMSS format
	Is business date	Char(8)		Business date in YYYYMMDD format
FDETL	Record Descriptor	Char(5)	FDETL	File Type Record descriptor
	SA File Line No	Number(10)		Sales Audit File Line no.
	Voucher seq Number	Number(20)		Unique identifier for an entry to sa_voucher table
	Voucher No	Char(25)		Voucher No.
	Voucher Type	Number(6)		Voucher Type
	Assigned Business Date	Char(8)		Business date in YYYYMMDD format
	Assigned Store	Number(10)		Store to which the voucher is assigned
	Issuing Date	Char(8)		Date this document was issued
	Issuing store	Number(10)		Store this document was issued from
	Issuing POS Register	Char(5)		Issuing Point Of Sale Register
	Issuing Cashier	Char(10)		Issuing Cashier
	Issued Tran Seq No.	Number(20)		Transaction sequence no.
	Issued item seq number	Number(4)		Will hold the item sequence of the item when the voucher is sold as an item (gift voucher)
	Issued Tender Seq No.	Number(4)		Tender sequence no.
	Issued Amount	Number(20)		Issued Amount * 10000 (4 implied digits)
	Issued Cust Name	Char(120)		Issued Customer Name
	Issued Customer Addr1	Char(240)		Issued Customer Addr1
	Issued Customer Addr2	Char(240)		Issued Customer Addr 2
	Issued Customer City	Char(120)		City of the customer, the voucher is issued
	Issued Customer State	Char(3)		State of the customer

Table 6-52 (Cont.) File Name: SAVO (Sales Audit Voucher File)

Record Name	Field Name	Field Type	Default Value	Description
	Issued Customer Postal Code	Char(30)		Postal address of the customer.
	Issued Customer Country	Char(3)		Country of the customer the voucher was issued.
	Recipient Name	Char(120)		Name of the intended recipient
	Recipient State	Char(3)		The state of the intended recipient.
	Recipient Country	Char(3)		The country of the intended recipient.
	Redemption Date	Char(8)		Date the voucher was redeemed.
	Redemption Store	Number(10)		Store, the voucher was redeemed at.
	Redemption Register	Char(5)		Register, the document was redeemed at.
	Redemption cashier	Char(10)		Cashier redeeming the voucher
	Redemption tran seq number	Number(20)		Transaction Number when the document was redeemed
	Redemption Tender seq number	Number(4)		This column will hold the tender sequence of the tender within the transaction when a voucher is redeemed as tender
	Redemption Amount	Number(20)		Amount the document was redeemed for*10000 (4 implied decimal places)
	Expiry Date	Char(8)		Expiry Date
	Status	Char(1)		Indicator showing the document's status, issued or redeemed. Valid values = I – Issued, R – Redeemed.
	Comments	Char(2000)		Comments
FTAIL	Record Descriptor	Char(5)	FTAIL	File Type Record descriptor
	SA File Line No.	Number(10)		Sales Audit File Line No.
	#lines	Number(10)		Total number of transaction lines in file (not including FHEAD and FTAIL)

Control Files

Table 6-53 File Name: Satdisc.ctl

Table Name	Column Name	Field Type	Field Width	Position	Description
SA_TRAN_DISC	TRAN_SEQ_NO	INTEGER EXTERNAL	20	1:20	
	ITEM_SEQ_NO	INTEGER EXTERNAL	4	21:24	
	DISCOUNT_SEQ_NO	INTEGER EXTERNAL	4	25:28	
	RMS_PROMO_TYPE	CHAR	6	29:34	
	PROMOTION	INTEGER EXTERNAL	10	35:44	
	DISC_TYPE	CHAR	6	45:50	
	COUPON_NO	CHAR	40	51:90	
	COUPON_REF_NO	CHAR	16	91:106	
	QTY	DECIMAL EXTERNAL	14	107:120	
	UNIT_DISCOUNT_AMOUNT	DECIMAL EXTERNAL	21	121: 141	
	STANDARD_QTY	DECIMAL EXTERNAL	14	142:155	
	STANDARD_UNIT_DISC_AMOUNT	DECIMAL EXTERNAL	21	156:176	
	REF_NO13	CHAR	30	177:206	
	REF_NO14	CHAR	30	207:236	
	REF_NO15	CHAR	30	237:266	
	REF_NO16	CHAR	30	267:296	
	ERROR_IND	CHAR	1	297:297	
	CATCHWEIGHT_IND	CHAR	1	298:298	
	UOM_QUANTITY	INTEGER EXTERNAL	12	299:310	
	PROMO_COMP	INTEGER EXTERNAL	10	311:320	This field maps to the OFFER_ID field from Pricing.
STORE	INTEGER EXTERNAL	10	321:330		
DAY	INTEGER EXTERNAL	3	331:333		

Table 6-54 File Name: Saigtax.ctl

Table Name	Column Name	Field Type	Field Width	Position	Description
SA_TRAN_IG TAX	TRAN_SEQ_NO	INTEGER EXTERNAL	20	1:20	
	ITEM_SEQ_NO	INTEGER EXTERNAL	4	21:24	
	IGTAX_SEQ_NO	INTEGER EXTERNAL	4	25:28	
	TAX_AUTHORITY	CHAR	10	29:38	
	IGTAX_CODE	CHAR	6	39:44	
	IGTAX_RATE	DECIMAL EXTERNAL	11	45:65	
	TOTAL_IGTAX_AMT	DECIMAL EXTERNAL	22	66:87	
	STANDARD_QTY	DECIMAL EXTERNAL	14	88:101	
	STANDARD_UNIT_I GTAX_AMT	DECIMAL EXTERNAL	21	102:122	
	ERROR_IND	CHAR	1	123:123	
	REF_NO_21	CHAR	30	124:153	
	REF_NO_22	CHAR	30	154:183	
	REF_NO_23	CHAR	30	184:213	
	REF_NO_24	CHAR	30	214:243	
	STORE	INTEGER EXTERNAL	10	244:253	
	DAY	INTEGER EXTERNAL	3	254:256	
TAX_CALC_TYPE	CHAR	6	257:262		

Table 6-55 File Name: Sacust.ctl

Table Name	Column Name	Field Type	Field Width	Position	Description
SA_CUSTOM ER	TRAN_SEQ_NO	INTEGER EXTERNAL DATE	20	1 :20	
	CUST_ID	CHAR	16	21 :36	
	CUST_ID_TYPE	CHAR	6	37 :42	
	NAME	CHAR	240	43 :162	
	ADDR1	CHAR	240	163:402	
	ADDR2	CHAR	240	403:642	
	CITY	CHAR	240	643:762	
	STATE	CHAR	3	763:765	

Table 6-55 (Cont.) File Name: Sacust.ctl

Table Name	Column Name	Field Type	Field Width	Position	Description
	POSTAL_CODE	CHAR	30	766:795	
	COUNTRY	CHAR	3	796:798	
	HOME_PHONE	CHAR	20	799:818	
	WORK_PHONE	CHAR	20	819:838	
	E_MAIL	CHAR	100	839:938	
	BIRTHDATE	DATE	8	939:946	FORMAT IS "YYYYMMDD"
	STORE	INTEGER EXTERNAL	10	947:956	
	DAY	INTEGER EXTERNAL	3	957:959	

Table 6-56 File Name: Sathead.ctl

Table Name	Column Name	Field Type	Field Width	Position	Description
SA_TRAN_H EAD	TRAN_SEQ_NO	INTEGER EXTERNAL	20	1:20	
	REV_NO	INTEGER EXTERNAL	3	21:23	
	STORE_DAY_SEQ _NO	INTEGER EXTERNAL	20	24:43	
	TRAN_DATETIME	DATE	14	44:57	FORMAT IS YYYYMM DDHH24MI SS
	REGISTER	CHAR	5	58:62	
	TRAN_NO	INTEGER EXTERNAL	10	63:72	
	CASHIER	CHAR	10	73:82	
	SALESPERSON	CHAR	10	83:92	
	TRAN_TYPE	CHAR	6	93:98	
	SUB_TRAN_TYPE	CHAR	6	99:104	
	ORIG_TRAN_NO	INTEGER EXTERNAL	10	105:114	
	ORIG_REG_NO	CHAR	5	115:119	
	REF_NO1	CHAR	30	120:149	
	REF_NO2	CHAR	30	150:179	
	REF_NO3	CHAR	30	180:209	
	REF_NO4	CHAR	30	210:239	
	REASON_CODE	CHAR	6	240:245	

Table 6-56 (Cont.) File Name: Sathead.ctl

Table Name	Column Name	Field Type	Field Width	Position	Description
	VENDOR_NO	CHAR	10	246:255	
	VENDOR_INVC_NO	CHAR	30	256:285	
	PAYMENT_REF_NO	CHAR	16	286:301	
	PROOF_OF_DELIVERY_NO	CHAR	30	302:331	
	STATUS	CHAR	6	332:337	
	VALUE	CHAR	22	338:359	INCLUDES AN OPTIONAL NEGATIVE SIGN AND A DECIMAL POINT.
	POS_TRAN_IND	CHAR	1	360:360	
	UPDATE_ID	CHAR	30	361:390	
	UPDATE_DATE_TIME	DATE	14	391:404	FORMAT IS YYYYMM DDHH24MI SS
	ERROR_IND	CHAR	1	405:405	
	BANNER_NO	INTEGER EXTERNAL	4	406:409	
	ROUND_AMT	INTEGER EXTERNAL	22	410:431	
	ROUNDED_OFF_AMOUNT	INTEGER EXTERNAL	22	432:453	
	CREDIT_PROMOTION_ID	INTEGER EXTERNAL	10	454:463	
	REF_NO25	CHAR	30	464:493	
	REF_NO26	CHAR	30	494:523	
	REF_NO27	CHAR	30	524:553	
	STORE	INTEGER EXTERNAL	10	554:563	
	DAY	INTEGER EXTERNAL	3	564:566	
	RTLOG_ORIG_SYSTEMS	CHAR	3	567:569	
	TRAN_PROCESS_SYSTEMS	CHAR	3	570:572	
	TRAN_DATE	DATE	8	573:580	
	REF_NO28	CHAR	30	581:610	
	REF_NO29	CHAR	30	611:640	
	REF_NO30	CHAR	30	641:670	

Table 6-56 (Cont.) File Name: Sathead.ctl

Table Name	Column Name	Field Type	Field Width	Position	Description
	REF_NO31	CHAR	30	671:700	

Table 6-57 File Name: Satitem.ctl

Table Name	Column Name	Field Type	Field Width	Position	Description
SA_TRAN_ITEM	TRAN_SEQ_NO	INTEGER EXTERNAL	20	1:20	
	ITEM_SEQ_NO	INTEGER EXTERNAL	4	21:24	
	ITEM_STATUS	CHAR	6	25:30	
	ITEM_TYPE	CHAR	6	31:36	
	ITEM	CHAR	25	37:61	
	REF_ITEM	CHAR	25	62:86	
	NON_MERCH_ITEM	CHAR	25	87:111	
	VOUCHER_NO	CHAR	25	112:136	
	DEPT	INTEGER EXTERNAL	4	137:140	
	CLASS	INTEGER EXTERNAL	4	141:144	
	SUBCLASS	INTEGER EXTERNAL	4	145:148	
	QTY	DECIMAL EXTERNAL	14	149:162	INCLUDES AN OPTIONAL NEGATIVE SIGN AND A DECIMAL POINT.
	UNIT_RETAIL	DECIMAL EXTERNAL	21	163:183	INCLUDES A DECIMAL POINT.
	UNIT_RETAIL_VAT_INCL	CHAR	1	184:184	INDICATES WHETHWE UNIT RETAIL INCL/EXCL VAT.
	SELLING UOM	CHAR	4	185:188	
	OVERRIDE_REASON	CHAR	6	189:194	
	ORIG_UNIT_RETAIL	DECIMAL EXTERNAL	21	195:215	INCLUDES A DECIMAL POINT.
	STANDARD_ORIG_UNIT_RETAIL	DECIMAL EXTERNAL	21	216:236	
	TAX_IND	CHAR	1	237:237	

Table 6-57 (Cont.) File Name: Satitem.ctl

Table Name	Column Name	Field Type	Field Width	Position	Description
	ITEM_SWIPED_IND	CHAR	1	238:238	
	ERROR_IND	CHAR	1	239:239	
	DROP_SHIP_IND	CHAR	1	240:240	
	WASTE_TYPE	CHAR	6	241:246	
	WASTE_PCT	DECIMAL EXTERNAL	12	247:258	INCLUDES A DECIMAL POINT.
	PUMP	CHAR	8	259:266	
	RETURN_REASON_CODE	CHAR	6	267:272	
	SALESPERSON	CHAR	10	273:282	
	EXPIRATION_DATE	DATE	8	283:290	FORMAT IS YYYYMM DD
	STANDARD_QTY	DECIMAL EXTERNAL	14	291:304	INCLUDES AN OPTIONAL NEGATIVE SIGN AND A DECIMAL POINT.
	STANDARD_UNIT_RETAIL	DECIMAL EXTERNAL	21	305:325	INCLUDES A DECIMAL POINT.
	STANDARD_UOM	CHAR	4	326:329	
	REF_NO5	CHAR	30	330:359	
	REF_NO6	CHAR	30	360:389	
	REF_NO7	CHAR	30	390:419	
	REF_NO8	CHAR	30	420:449	
	CATCHWEIGHT_IND	CHAR	1	450:450	
	SELLING_ITEM	CHAR	25	451:475	
	CUSTOMER_ORDER_LINE_NO	INTEGER EXTERNAL	6	476:481	
	MEDIA_ID	INTEGER EXTERNAL	10	482:491	
	UOM_QUANTITY	INTEGER EXTERNAL	12	492:503	
	TOTAL_IGTAX_AMT	DECIMAL EXTERNAL		504:524	
	UNIQUE_ID	CHAR	25	525:652	

Table 6-57 (Cont.) File Name: Satitem.ctl

Table Name	Column Name	Field Type	Field Width	Position	Description
	STORE	INTEGER EXTERNAL	10	653:662	
	DAY	INTEGER EXTERNAL	3	663:665	
	CUST_ORDER_NO	CHAR	48	666:713	
	CUST_ORDER_DATE	DATE	14	714:727	FORMAT IS YYYYMMDDHH 24MISS
	FULFILL_ORDER_NO	CHAR	48	728:775	
	NO_INV_RET_IND	CHAR	1	776:776	
	SALES_TYPE	CHAR	1	777:777	
	RETURN_WH	INTEGER EXTERNAL	10	778:787	
	RETURN_DISPOSITION	CHAR	10	788:797	
	ORIG_STORE	INTEGER EXTERNAL	10	798:807	
	ORIG_TRAN_NO	INTEGER EXTERNAL	10	808:817	
	FULFILLMENT_LOC_TYP E	CHAR	2	818:819	
	FULFILLMENT_LOC	INTEGER EXTERNAL	10	820:829	
	POSTING_STORE	INTEGER EXTERNAL	10	830:839	

Table 6-58 File Name: Sattend.ctl

Table Name	Column Name	Field Type	Field Width	Position	Description
SA_TRAN_TEN DER	TRAN_SEQ_NO	INTEGER EXTERNAL	20	1:20	
	TENDER_SEQ_NO	INTEGER EXTERNAL	4	21:24	
	TENDER_TYPE_GRP OUP	CHAR	6	25:30	
	TENDER_TYPE_ID	INTEGER EXTERNAL	6	31:36	

Table 6-58 (Cont.) File Name: Sattend.ctf

Table Name	Column Name	Field Type	Field Width	Position	Description
	TENDER_AMT	DECIMAL EXTERNAL	22	37:58	INCLUDES AN OPTIONAL NEGATIVE SIGN AND A DECIMAL POINT.
	CC_NO	INTEGER EXTERNAL	40	59:98	
	CC_EXP_DATE	DATE	8	99:106	FORMAT IS YYYYMM DD
	CC_AUTH_NO	CHAR	16	107:122	
	CC_AUTH_SRC	CHAR	6	123:128	
	CC_ENTRY_MODE	CHAR	6	129:134	
	CC_CARDHOLDER_ VERF	CHAR	6	135:140	
	CC_TERM_ID	CHAR	5	141:145	
	CC_SPEC_COND	CHAR	6	146:151	
	CC_TOKEN	CHAR	40	152:191	
	VOUCHER_NO	CHAR	25	192:216	
	COUPON_NO	CHAR	40	217:256	
	COUPON_REF_NO	CHAR	16	257:272	
	CHECK_ACCT_NO	CHAR	30	273:302	
	CHECK_NO	INTEGER EXTERNAL	10	303:312	
	IDENTI_METHOD	CHAR	6	313:318	
	IDENTI_ID	CHAR	40	319:358	
	ORIG_CURRENCY	CHAR	3	359:361	
	ORIG_CURR_AMT	DECIMAL EXTERNAL	22	362:383	
	REF_NO9	CHAR	30	384:413	
	REF_NO10	CHAR	30	414:443	
	REF_NO11	CHAR	30	444:473	
	REF_NO12	CHAR	30	474:503	
	ERROR_IND	CHAR	1	504:504	
	STORE	INTEGER EXTERNAL	10	505:514	
	DAY	INTEGER EXTERNAL	3	515:517	

Table 6-59 File Name: Satpymt.ctl

Table Name	Column Name	Field Type	Field Width	Position	Description
SA_TRAN_PAYMENT	TRAN_SEQ_NO	INTEGER EXTERNAL	20	1:20	
	PAYMENT_SEQ_NO	INTEGER EXTERNAL	20	21:24	
	PAYMENT_AMT	DECIMAL EXTERNAL	5	25:46	
	ERROR_IND	CHAR	10	47:47	
	STORE	INTEGER EXTERNAL	6	48:57	
	DAY	INTEGER EXTERNAL	3	58:60	

Table 6-60 File Name: Samisstr.ctl

Table Name	Column Name	Field Type	Field Width	Position	Description
SA_MISSING_TRAN	MISS_TRAN_SEQ_NO	INTEGER EXTERNAL	20	1:20	
	STORE_DAY_SEQ_NO	INTEGER EXTERNAL	20	21:40	
	REGISTER	CHAR	5	41:45	
	TRAN_NO	INTEGER EXTERNAL	10	46:55	
	STATUS	CHAR	6	56:61	
	RTLOG_ORIG_SYS	CHAR	3	62:64	

Table 6-61 File Name: Sattax.ctl

Table Name	Column Name	Field Type	Field Width	Position	Description
SA_TRAN_TAX	TRAN_SEQ_NO	INTEGER EXTERNAL	20	1:20	
	TAX_CODE	CHAR	6	21:26	
	TAX_SEQ_NO	INTEGER EXTERNAL	4	27:30	
	TAX_AMT	DECIMAL EXTERNAL	22	31:52	INCLUDES AN OPTIONAL NEGATIVE SIGN AND A DECIMAL POINT
	ERROR_IND	CHAR	1	53:53	
	REF_NO17	CHAR	30	54:83	

Table 6-61 (Cont.) File Name: Sattax.ctl

Table Name	Column Name	Field Type	Field Width	Position	Description
	REF_NO18	CHAR	30	84:113	
	REF_NO19	CHAR	30	114:143	
	REF_NO20	CHAR	30	144:173	
	STORE	INTEGER EXTERNAL	10	174:183	
	DAY	INTEGER EXTERNAL	3	184:186	

Table 6-62 File Name: Sacustatt.ctl

Table Name	Column Name	Field Type	Field Width	Position	Description
SA_CUST_AT TRIB	TRAN_SEQ_NO	INTEGER EXTERNAL	20	1:20	
	ATTRIB_SEQSO	CHAR	4	21:24	
	ATTRIB_TYPE	CHAR	6	25:30	
	ATTRIB_VALUE	CHAR	6	31:36	
	STORE	INTEGER EXTERNAL	10	37:46	
	DAY	INTEGER EXTERNAL	3	47:49	

Table 6-63 File Name: Saerror.ctl

Table Name	Column Name	Field Type	Field Width	Position	Description
SA_ERROR	ERROR_SEQ_NO	INTEGER EXTERNAL	20	1:20	
	STORE_DAY_SEQ_ NO	INTEGER EXTERNAL	20	21:40	
	BAL_GROUP_SEQ_ _NO	INTEGER EXTERNAL	20	41:60	
	TOTAL_SEQ_NO	INTEGER EXTERNAL	20	61:80	
	TRAN_SEQ_NO	INTEGER EXTERNAL	20	81:100	
	ERROR_CODE	CHAR	25	101:125	
	KEY_VALUE_1	INTEGER EXTERNAL	4	126:129	
	KEY_VALUE_2	INTEGER EXTERNAL	4	130:133	
	REC_TYPE	CHAR	6	134:139	

Table 6-63 (Cont.) File Name: Saerror.ctl

Table Name	Column Name	Field Type	Field Width	Position	Description
	STORE_OVERRIDE_IND	CHAR	1	140:140	
	HQ_OVERRIDE_IND	CHAR	1	141:141	
	UPDATE_ID	CHAR	30	142:171	
	UPDATE_DATE_TIME	DATE	14	172:185	FORMAT IS "YYYYMMDD HH24MISS"
	ORIG_VALUE	CHAR	70	186:255	
	STORE	INTEGER EXTERNAL	10	256:265	
	DAY	INTEGER EXTERNAL	3	266:268	
	KEY_VALUE_3	INTEGER EXTERNAL	4	269:272	

Table 6-64 File Name: Sathatt.ctl

Table Name	Column Name	Field Type	Field Width	Position	Description
SA_TRAN_HEAD_ATTRIB	TRAN_SEQ_NO	INTEGER EXTERNAL	20	1:20	
	ATTRIB_SEQ_NO	INTEGER EXTERNAL	4	21:24	
	ATTRIB_TYPE	CHAR	6	25:30	
	ATTRIB_VALUE	CHAR	120	31:150	
	STORE	INTEGER EXTERNAL	10	151:160	
	DAY	INTEGER EXTERNAL	3	161:163	
	ERROR_IND	CHAR	1	164:164	

Table 6-65 File Name: Saitatt.ctl

Table Name	Column Name	Field Type	Field Width	Position	Description
SA_TRAN_ITEM_ATTRIB	TRAN_SEQ_NO	INTEGER EXTERNAL	20	1:20	
	ITEM_SEQ_NO	INTEGER EXTERNAL	4	21:24	
	ATTRIB_SEQ_NO	INTEGER EXTERNAL	4	25:28	
	ATTRIB_TYPE	CHAR	6	29:34	

Table 6-65 (Cont.) File Name: Saitatt.ctl

Table Name	Column Name	Field Type	Field Width	Position	Description
	ATTRIB_VALUE	CHAR	120	35:154	
	STORE	INTEGER EXTERNAL	10	155:164	
	DAY	INTEGER EXTERNAL	3	165:167	
	ERROR_IND	CHAR	1	168:168	

Table 6-66 File Name: Saidatt.ctl

Table Name	Column Name	Field Type	Field Width	Position	Description
SA_TRAN_DI SC_ATTRIB	TRAN_SEQ_NO	INTEGER EXTERNAL	20	1:20	
	ITEM_SEQ_NO	INTEGER EXTERNAL	4	21:24	
	DISCOUNT_SEQ_NO	INTEGER EXTERNAL	4	25:28	
	ATTRIB_SEQ_NO	INTEGER EXTERNAL	4	29:32	
	ATTRIB_TYPE	CHAR	6	33:38	
	ATTRIB_VALUE	CHAR	120	39:158	
	STORE	INTEGER EXTERNAL	10	159:168	
	DAY	INTEGER EXTERNAL	3	169:171	
	ERROR_IND	CHAR	1	172:172	

Table 6-67 File Name: Saixatt.ctl

Table Name	Column Name	Field Type	Field Width	Position	Description
SA_TRAN_IG TAX_ATTRIB	TRAN_SEQ_NO	INTEGER EXTERNAL	20	1:20	
	ITEM_SEQ_NO	INTEGER EXTERNAL	4	21:24	
	IGTAX_SEQ_NO	INTEGER EXTERNAL	4	25:28	
	ATTRIB_SEQ_NO	INTEGER EXTERNAL	4	29:32	
	ATTRIB_TYPE	CHAR	6	33:38	
	ATTRIB_VALUE	CHAR	120	39:158	

Table 6-67 (Cont.) File Name: Saixatt.ctf

Table Name	Column Name	Field Type	Field Width	Position	Description
	STORE	INTEGER EXTERNAL	10	159:168	
	DAY	INTEGER EXTERNAL	3	169:171	
	ERROR_IND	CHAR	1	172:172	

Table 6-68 File Name: Satxatt.ctf

Table Name	Column Name	Field Type	Field Width	Position	Description
SA_TRAN_TAX _ATTRIB	TRAN_SEQ_NO	INTEGER EXTERNAL	20	1:20	
	TAX_SEQ_NO	INTEGER EXTERNAL	4	21:24	
	ATTRIB_SEQ_NO	INTEGER EXTERNAL	4	25:28	
	ATTRIB_TYPE	CHAR	6	29:34	
	ATTRIB_VALUE	CHAR	120	35:154	
	STORE	INTEGER EXTERNAL	10	155:164	
	DAY	INTEGER EXTERNAL	3	165:167	
	ERROR_IND	CHAR	1	168: 168	

Table 6-69 File Name: Sattatt.ctf

Table Name	Column Name	Field Type	Field Width	Position	Description
SA_TRAN_TEN DER_ATTRIB	TRAN_SEQ_NO	INTEGER EXTERNAL	20	1:20	
	TENDER_SEQ_NO	INTEGER EXTERNAL	4	21:24	
	ATTRIB_SEQ_NO	INTEGER EXTERNAL	4	25:28	
	ATTRIB_TYPE	CHAR	6	29:34	
	ATTRIB_VALUE	CHAR	120	35:154	
	STORE	INTEGER EXTERNAL	10	155:164	
	DAY	INTEGER EXTERNAL	3	165:167	
	ERROR_IND	CHAR	1	168:168	

Table 6-70 File Name: Satwritelock.ctl

Table Name	Column Name	Field Type	Field Width	Position	Description
SA_TRAN_W	STORE_DAY_SEQ_	INTEGER	20	1 :20	
RITE_LOCK	NO	EXTERNAL DATE			
	TRAN_SEQ_NO	INTEGER EXTERNAL DATE	20	21 :40	

Sales Audit Interface File Layout [rtlog]

The following illustrates the file layout format of the Oracle Retail TLOG. The content of each Oracle Retail TLOG file is per store per day. The filename convention is RTLOG_STORE_DATETIME.DAT (for example, RTLOG_1234_01221989010000.DAT).

Involves round off fields, credit promotion id, tax (vat) at item level and payment amount of customer orders.

Document has been modified with new tender types.

Added logic of handling both VAT-TAX in the system.

```
FHEAD      (Only 1 per file, required)
THEAD      (Multiple expected, one per transaction, required for each
transaction)
THATT      (Attribute record specific to the THEAD record - Multiple allowed,
optional)
TCUST      (Only 1 per THEAD record allowed, optional for some transaction
types, see table below)
CATT      (Attribute record specific to the TCUST record - Multiple allowed,
only valid if TCUST exists)
TITEM      (Multiple allowed per transaction, optional for some transaction
types, see table below)
ITATT      (Attribute record specific to the TITEM record - Multiple allowed,
optional and only valid if TITEM exists)
IDISC      (Discount record specific to the TITEM record - Multiple allowed per
item, optional see table below)
IDATT      (Attribute record specific to the IDISC record - Multiple allowed,
optional and only valid if IDISC exists)
IGTAX      (VAT/Tax record specific to the TITEM record - Multiple allowed per
item, optional. Either TTAX or IGTAX should appear in a given RTLOG, if
originating system is POS, but not both, see table below). If originating system
is OMS, both IGTAX and TTAX can appear but only the one matching the store's tax
type will be processed, the other record will be ignored.
IXATT      (Attribute record specific to the IGTAX record - Multiple allowed,
optional and only valid if IGTAX exists)
TTAX      (Vat/Tax record specific to the THEAD record - Multiple allowed per
transaction, optional. Either TTAX or IGTAX should appear in a given RTLOG, if
originating system is POS, but not both, see table below). If originating system
is OMS, both IGTAX and TTAX can appear but only the one matching the store's tax
type will be processed, the other record will be ignored.
TXATT      (Attribute record specific to the TTAX record - Multiple allowed,
optional and only valid if TTAX exists)
TPYMT      (Multiple allowed per transaction, will have the deposit amount for
pickup/delivery/layaway orders, optional see table below)
```

TTEND (Multiple allowed per transaction, optional for some transaction types, see table below)
 TTATT (Attribute record specific to the TTEND record - Multiple allowed, optional and only valid if TTEND exists)
 TTAIL (1 per THEAD, required)
 FTAIL (1 per file, required)

The order of the records within the transaction layout above is important. It aids processing by ensuring that the information is present when it is needed.

Fields expected in RTLog format based on the changes adopted -

Version Number	1	2a	2b	3	4
Version Description	Initial Version (V16.0.x SaaS)	Initial Version Plus Fulfilment Type and Location	Initial Version Plus Reference Numbers 28-31	Initial Version Plus Fulfilment Type and Location and Reference Numbers 28-31	Version 3 Plus Posting Store
Reference No. 28	N	N	Y	Y	Y
Reference No. 29	N	N	Y	Y	Y
Reference No. 30	N	N	Y	Y	Y
Reference No. 31	N	N	Y	Y	Y
Fulfillment Location type	N	Y	N	Y	Y
Fulfillment Location	N	Y	N	Y	Y
Posting Store	N	N	N	N	Y

Record Name	Field Name	Field Type	Default Value	Description	Required?	Justification/ Padding
File Header	File Type Record Descriptor	Char(5)	FHEAD	Identifies file record type	Y	Left/Blank
	File Line Identifier	Number(10)	Specified by external system	ID of current line being processed by input file.	Y	Right/0
	File Type Definition	Char(4)	RTLGL	Identifies file as 'Oracle Retail TLOG'.	Y	Left/Blank
	File Create Date	Char(14)	Create date	Date and time file was written by external system (YYYYMMDDHHMMSS).	Y	Left/None
	Business Date	Char(8)	Business Date to process	Business date of transactions. (YYYYMMDD).	Y	Left/None
	Location Number	Char(10)	Specified by external system	Store or warehouse identifier.	Y	Left/None

Record Name	Field Name	Field Type	Default Value	Description	Required?	Justification/ Padding
	Reference Number	Char(30)	Specified by external system	This may contain the Polling ID associated with the consolidated TLOG file or used for other purpose.	N	Left/Blank
	RTLOG Originating System	Char(3)	POS	Identifies the system the RTLOG file originated from. Valid values are OMS and POS.	Y	Left/None
Transaction Header	File Type Record Descriptor	Char(5)	THEAD	Identifies file record type.	Y	Left/Blank
	File Line Identifier	Number(10)	Specified by external system	ID of current line being processed by input file.	Y	Right/0
	Register	Char(5)		Till used at store.	Y	Left/Blank
	Transaction Date	Char(14)	Transaction date	Date transactions were processed at the POS (YYYYMMDDHHMMSS).	Y	Left/None
	Transaction Number	Number(10)		Transaction identifier. If sa_system_options.wkstation_tran_append_ind is 'Y' then the first 3 digits indicate the workstation id and last seven digits indicate the transaction number.	Y	Right/0
	Cashier	Char(10)		Cashier identifier.	N	Left/Blank
	Salesperson	Char(10)		Salesperson identifier.	N	Left/Blank
	Transaction Type	Char(6)	Refer to 'TRAT' code_type for a list of valid types.	Transaction type.	Y	Left/Blank
	Sub-transaction type	Char(6)	Refer to 'TRAS' code_type for a list of valid types.	Sub-transaction type. For N sale, it can be employee, drive-off and so on	N	Left/Blank
	Orig_tran_no	Number(10)		Populated only for post-void even exchange and return transactions. Transaction number for the original tran that will be cancelled.	N	Right/0
	Orig_reg_no	Char(5)		Populated only for post-void transactions. Register number from the original tran.	N	Left/Blank

Record Name	Field Name	Field Type	Default Value	Description	Required?	Justification/ Padding
	Reason Code	Char(6)	Refer to 'REAC' code_type for a list of valid codes. If the transaction type is 'PAIDOU' and the sub transaction type is 'MV' or 'EV' than the valid codes come from the non_merch_code_head table.	Reason entered by cashier for some transaction types. Required for Paid In and Paid out transaction types, but can also be used for voids, returns, and so on	N	Left/Blank
	Vendor Number	Char(10)		Supplier id for a merchandise vendor paid out transaction, partner id for an expense vendor paid out transaction.	N	Left/Blank
	Vendor Invoice Number	Char(30)		Invoice number for a vendor paid out transaction.	N	Left/Blank
	Payment Reference Number	Char(16)		The reference number of the tender used for a vendor payout. This could be the money order number, check number, and so on	N	Left/Blank
	Proof of Delivery Number	Char(30)		Proof of receipt number given by the vendor at the time of delivery. This field is populated for a vendor paid out transaction.	N	Left/Blank
	Reference Number 1	Char(30)		Number associated with a particular transaction, for example weather for a Store Conditions transaction. The SA_REFERENCE table defines what this field can contain for each transaction type.	N	Left/Blank
	Reference Number 2	Char(30)		Second generic reference number.	N	Left/Blank
	Reference Number 3	Char(30)		Third generic reference number.	N	Left/Blank
	Reference Number 4	Char(30)		Fourth generic reference number.	N	Left/Blank

Record Name	Field Name	Field Type	Default Value	Description	Required?	Justification/ Padding
	Value Sign	Char(1)	Refer to 'SIGN' code_type for a list of valid codes.	Sign of the value.	Y if Value is present	Left/None
	Value	Number(20)		Value with 4 implied decimal places. Should only be populated by the retailer for TOTAL type transactions, populated by Oracle Retail sales audit for SALE, RETURN trans.	Y if tran is a TOTAL.	Right/0 when value is present. Blank when no value is sent.
	Banner id	Number(4)		Banner ID of the location	Y	Right/0 when value is present. Blank when no value is sent
	Rounded Amount Sign	Char(1)	Refer to 'SIGN' code_type for a list of valid codes.	Sign of Rounded Amount. Amount Sign is not used.	Y	Left/None
	Rounded Amount	Number(20)		Total Rounded Amount with 4 implied decimal places. Rounded Amount is not used.	Y	Right/0 when RoundedAmount is present otherwise blank
	Rounded Off Sign	Char(1)	Refer to 'SIGN' code_type for a list of valid codes.	Rounded Off Sign is not used.	Y	Left/None
	Rounded Off Amount	Number(20)		Rounded Off Amount with 4 implied decimal places. Rounded Off Amount is not used.	Y	Right/0 when RoundedAmount is present otherwise blank
	Credit Promotion Id	Char(10)		Credit Promotional Id	Y	Left/None
	Reference Number 25	Char(30)			N	Left/Blank
	Reference Number 26	Char(30)			N	Left/Blank
	Reference Number 27	Char(30)			N	Left/Blank
	Transaction Processing System	Char(3)	Valid values are OMS and POS.	Contains the ID of the system that processed the transaction.	N	Left/None

Record Name	Field Name	Field Type	Default Value	Description	Required?	Justification/ Padding
	Reference Number 28	Char(30)		Generic Reference Number. It can be ignored if not needed based on the type of RTLOG being used.	N	Left/Blank
	Reference Number 29	Char(30)		Generic Reference Number. It can be ignored if not needed based on the type of RTLOG being used.	N	Left/Blank
	Reference Number 30	Char(30)		Generic Reference Number. It can be ignored if not needed based on the type of RTLOG being used.	N	Left/Blank
	Reference Number 31	Char(30)		Generic Reference Number. It can be ignored if not needed based on the type of RTLOG being used.	N	Left/Blank
Transaction Header Attribute	File Type Record Descriptor	Char(5)	THATT	Identifies file record type	Y	Left/Blank
	File Line Identifier	Number(10)	Specified by external system	ID of current line being processed by input file.	Y	Right/0
	Attribute Type	Char(6)	Refer to 'SAHA' code_type for a list of valid types	Type of transaction header attribute	Y	Left/Blank
	Attribute Value	Char(120)		Value of transaction header attribute	Y	Left/Blank
Transaction Custom	File Type Record Descriptor	Char(5)	TCUST	Identifies file record type	Y	Left/Blank
	File Line Identifier	Number(10)	Specified by external system	ID of current line being processed by input file.	Y	Right/0
	Customer ID	Char(16)	Customer identifier	The ID number of a customer.	Y	Left/Blank
	Customer Type ID	Char(6)	Refer to 'CIDT' code_type for a list of valid types	Customer ID type.	Y	Left/Blank
	Customer Name	Char(120)		Customer name.	N	Left/Blank
	Address 1	Char(240)		Customer address.	N	Left/Blank

Record Name	Field Name	Field Type	Default Value	Description	Required?	Justification/ Padding
	Address 2	Char(240)		Additional field for customer address.	N	Left/Blank
	City	Char(120)		City.	N	Left/Blank
	State	Char(12)	State identifier	State.	N	Left/Blank
	Zip Code	Char(30)	Zip identifier	Zip code.	N	Left/Blank
	Country	Char(3)		Country.	N	Left/Blank
	Home Phone	Char(20)		Telephone number at home.	N	Left/Blank
	Work Phone	Char(20)		Telephone number at work.	N	Left/Blank
	E-mail	Char(100)		E-mail address.	N	Left/Blank
	Birthdate	Char(8)		Date of birth. (YYYYMMDD)	N	Left/Blank
Customer Attribute	File Type Record Descriptor	Char(5)	CATT	Identifies file record type	Y	Left/Blank
	File Line Identifier	Number(10)	Specified by external system	ID of current line being processed by input file.	Y	Right/0
	Attribute type	Char(6)	Refer to 'SACA' code_type for a list of valid types	Type of customer attribute	Y	Left/Blank
	Attribute value	Char(6)	Refer to members of 'SACA' code_type for a list of valid values	Value of customer attribute.	Y	Left/Blank
Transaction Item	File Type Record Descriptor	Char(5)	TITEM	Identifies file record type.	Y	Left/Blank
	File Line Identifier	Number(10)	Specified by external system	ID of current line being processed by input file.	Y	Right/0

Record Name	Field Name	Field Type	Default Value	Description	Required?	Justification/ Padding
	Item Status	Char(6)	Refer to 'SASI' code_type for a list of valid codes.	Status of the item within the transaction. Valid values are: V – Void item S - Sold item R - Returned item ORI – Order Initiate ORC – Order Cancel ORD – Order Complete LIN – Layaway Initiate LCA – Layaway Cancel LCO – Layaway Complete ADJ – Appeasement/ Adjustment	Y	Left/Blank
	Item Type	Char(6)	Refer to 'SAIT' code_type for a list of valid codes.	Identifies what type of item is transmitted.	Y	Left/Blank
	Item number type	Char(6)	Refer to 'UPCT' code_type for a list of valid codes.	Identifies type of item number if item type is ITEM or REF.	N	Left/Blank
	Format ID	Char(1)	VPLU format ID.	Used to interpret VPLU items.	N	Left/Blank
	Item	Char(25)	Item identifier	Identifies merchandise item.	N	Left/Blank
	Reference Item	Char(25)	Item identifier	Identifies sub-transaction level merchandise item.	N	Left/Blank
	Non-Merchandise Item	Char(25)	Item identifier	Identifies non-merchandise item.	N	Left/Blank
	Voucher	Char(25)		Gift certificate number	N	Right/0
	Department	Number(4)		Identifies the department this item belongs to. This is filled in by saimptlog.	N	Right/Blank
	Class	Number(4)	Item's class	Class of item sold or returned. Not required from a retailer, populated by Oracle Retail sales audit. This is filled in by saimptlog.	N	Right/Blank

Record Name	Field Name	Field Type	Default Value	Description	Required?	Justification/ Padding
	Subclass	Number(4)	Item's subclass	Subclass of item sold or returned. Not required from a retailer, populated by Oracle Retail sales audit. This is filled in by saimptlog.	N	Right/Blank
	Quantity Sign	Char(1)	Refer to 'SIGN' code_type for a list of valid codes.	Sign of the quantity	Y	Left/None
	Quantity	Number(12)		Number of items purchased with 4 decimal places.	Y	Right/0
	Selling Unit of Measure	Char(4)		Unit of measure of item's quantity.	Y	Left/None
	Unit Retail	Number(20)		Unit retail with 4 implied decimal places.	Y	Right/0
	Override Reason	Char(6)	Refer to 'ORRC' code_type for a list of valid codes.	This column will be populated when an item's price has been overridden at the POS to define why it was overridden.	Y if unit retail was manually entered	Left/Blank
	Original Unit Retail	Number(20)		Value with 4 implied decimal places. This column will be populated when the item's price was overridden at the POS and the item's original unit retail is known.	Y if unit retail was manually entered	Right/0
	Taxable Indicator	Char(1)	Refer to 'YSNO' code_type for a list of valid codes.	Indicates whether or not item is taxable.	Y	Left/None
	Pump	Char(8)		Fuel pump identifier.	N	Left/Blank
	Reference Number 5	Char(30)		Number associated with a particular item within a transaction, for example special order number. The sa_reference table defines what this field can contain for each transaction type.	N	Left/Blank
	Reference Number 6	Char(30)		Second generic reference number at the item level.	N	Left/Blank
	Reference Number 7	Char(30)		Third generic reference number at the item level.	N	Left/Blank

Record Name	Field Name	Field Type	Default Value	Description	Required?	Justification/ Padding
	Reference Number 8	Char(30)		Fourth generic reference number at the item level.	N	Left/Blank
	Item_swipe_d_ind	Char(1)	Refer to 'YSNO' code_type for a list of valid codes.	Indicates if the item was automatically entered into the POS system or if it had to be manually keyed.	Y	Left/None
	Return Reason Code	Char(6)	Refer to 'SARR' code_type for a list of valid codes.	The reason an item was returned.	N	Left/Blank
	Salesperson	Char(10)		The salesperson who sold the item.	N	Left/Blank
	Expiration_date	Char(8)		Gift certificate expiration date (YYYYMMDD).	N	
	Drop Ship Ind	Char(1)	Refer to 'YSNO' code type for a list of valid codes.	Indicates whether item is part of a drop shipment.	Y	Left/None
	Uom_qty	Number(12)		Qty of items purchased in the given UOM with 4 decimal places.	Y	Right/0
	Catchweight_ind	Char(1)	Valid values are 'Y','N'	Identifies if the item is a catchweight item		Left/None
	Selling item	Char(25)	Item identifier	Identifies selling item.	N	Left/Blank
	Customer order line no	Number(6)		Identifies the customer order number	N	Left/Blank
	Media id	Number(10)		Identifies the customer media id	N	Left/Blank
	Total Igtax Amount	Number(21)		Contains the Igtax amount.	N	Right/0
	Unique ID	Char(128)			N	Left/Blank
	Customer Order Number	Char(48)		Contains the customer order ID.	N	Left/None
	Customer Order Date	Char(14)		Contains the customer order date. Format is: YYYYMMDDHHMMSS. Customer orders and layaways require customer order date.	N	Left/Blank
	Fulfillment Order Number	Char(48)		Contains the order ID of the fulfillment order.	N	Left/None

Record Name	Field Name	Field Type	Default Value	Description	Required?	Justification/ Padding
	No Inventory Return	Char(1)		Indicates if there is associated inventory with the return transaction with an External Customer Order sales type.	N	Left/Blank
	Sales Type	Char(1)		Indicates if the transaction is an In Store Customer Order, External Customer Order or Regular Sale.	Y	Left/Blank
	Return Warehouse	Char(10)		Contains the ID of the physical warehouse the inventory is returned to.	N	Left/Blank
	Return Disposition	Char(10)		Contains the return disposition of the returned items.	N	Left/Blank
	Original Store	Char(10)		Contains the original store.	N	Left/Blank
	Original Transaction No	Char(10)		Contains the original transaction no.	N	Left/Blank
	Fulfillment Loc Type	Char(2)	Refer to 'FLTP' code type for a list of valid types	Contains the fulfillment order location type. It is needed only if the file is for an OMS transaction.	N	Left/Blank
	Fulfillment Loc	Number(10)		Fulfillment Location ID. It is needed only if the file is for an OMS transaction.	N	Left/Blank
	Posting Store	Number(10)		Contains the store at which the item sale/ return should be accounted for in case of cross-store sales happening at co-located stores. It is expected that this field will be populated only for items that are checked out at a different store from the one at which they are originally managed.	N	Left/Blank
Transaction Item Attribute	File Type Record Descriptor	Char(5)	ITATT	Identifies file record type	Y	Left/Blank
	File Line Identifier	Number(10)	Specified by external system	ID of current line being processed by input file.	Y	Right/0

Record Name	Field Name	Field Type	Default Value	Description	Required?	Justification/ Padding
Item Discount	Attribute Type	Char(6)	Refer to 'SAIA' code_type for a list of valid types	Type of item attribute	Y	Left/Blank
	Attribute Value	Char(120)		Value of item attribute	Y	Left/Blank
	File Type Record Descriptor	Char(5)	IDISC	Identifies file record type	Y	Left/Blank
	File Line Identifier	Number(10)	Specified by external system	ID of current line being processed by input file.	Y	Right/0
	Merchandising Promotion Number	Char(6)	Refer to 'PRMT' code_type for a list of valid types	The Merchandising promotion type.	Y	Left/Blank
	Discount Reference Number	Number(10)		Discount reference number is associated with the discount type (e.g. if discount type is a promotion, this contains the promotion number).	N	Left/Blank
	Discount Type	Char(6)	Refer to 'SADT' code_type for a list of valid types.	The type of discount within a promotion. This allows a retailer to further break down coupon discounts within the "In-store" promotion, for example.	N	Left/Blank
	Coupon Number	Char(40)		Number of a store coupon used as a discount.	Y if coupon	Left/Blank
	Coupon Reference Number	Char(16)		Additional information about the coupon, usually contained in a second bar code on the coupon.	Y if coupon	Left/Blank
	Quantity Sign	Char(1)	Refer to 'SIGN' code_type for a list of valid codes.	Sign of the quantity.	Y	Left/None
Quantity	Number(12)		The quantity purchased that discount is applied with 4 implied decimal places.	Y	Right/0	
Unit Discount Amount	Number(20)		Unit discount amount for this item with 4 implied decimal places.	Y	Right/0	

Record Name	Field Name	Field Type	Default Value	Description	Required?	Justification/ Padding
	Reference Number 13	Char(30)		Number associated with a particular transaction type at the discount level. The sa_reference table defines what this field can contain for each transaction type.	N	Left/Blank
	Reference Number 14	Char(30)		Second generic reference number at the discount level.	N	Left/Blank
	Reference Number 15	Char(30)		Third generic reference number at the discount level.	N	Left/Blank
	Reference Number 16	Char(30)		Fourth generic reference number at the discount level.	N	Left/Blank
	Uom_qty	Number(12)		Qty of items purchased in the given UOM with 4 decimal places.	Y	Right/0
	Catchweight_ind	Char(1)	Valid values are 'Y','N'	Identifies if the item is a catchweight item		Left/None
	Promo component	Number(10)		If the discount is a promotion, this field contains the Offer ID value associated with the Promotion (discount reference number)	N	Left/Blank
Transaction Item Discount Attribute	File Type Record Descriptor	Char(5)	IDATT	Identifies file record type	Y	Left/Blank
	File Line Identifier	Number(10)	Specified by external system	ID of current line being processed by input file.	Y	Right/0
	Attribute Type	Char(6)	Refer to 'SADA' code_type for a list of valid types	Type of transaction item discount attribute	Y	Left/Blank
	Attribute Value	Char(120)		Value of transaction item discount attribute	Y	Left/Blank
Item Tax	File Type Record Descriptor	Char(5)	IGTAX	Identifies file record type	Y	Left/Blank
	File Line Identifier	Number(10)	Specified by external system	ID of current line being processed by input file.	Y	Right/0

Record Name	Field Name	Field Type	Default Value	Description	Required?	Justification/ Padding
	Tax Authority	Char(10)			Y	Left/Blank
	Igtax Code	Char(6)	Refer to vat_code of vat_codes table.	Igtax code (tax code/vat code) to represent whether it is a State, City or some other tax code/vat code.	Y	Left/Blank
	Igtax Rate	Number(20)		Igtax rate with 4 implied decimal places.	Y	Right/0
	Igtax Amount Sign	Char(1)	Refer to 'SIGN' code_type for a list of valid codes.	Sign of the Igtax Amount	Y	Left/None
	Igtax Amount	Number(21)		Total igtax amount for this item with 5 implied decimal places.	Y	Right/0
	Reference Number 21	Char(30)			N	Left/None
	Reference Number 22	Char(30)			N	Left/None
	Reference Number 23	Char(30)			N	Left/None
	Reference Number 24	Char(30)			N	Left/None
	Tax Calculation Type	Char(6)	Refer to the 'GTTT' code type for the list of valid values.	Contains the tax calculation type.	N	Left/None
Transaction Item Tax Attribute	File Type Record Descriptor	Char(5)	IXATT	Identifies file record type	Y	Left/Blank
	File Line Identifier	Number(10)	Specified by external system	ID of current line being processed by input file.	Y	Right/0
	Attribute Type	Char(6)	Refer to 'SAXA' code_type for a list of valid types	Type of transaction item tax attribute	Y	Left/Blank
	Attribute Value	Char(120)		Value of transaction item tax attribute	Y	Left/Blank
Transaction Tax	File Type Record Descriptor	Char(5)	TTAX	Identifies file record type	Y	Left/Blank
	File Line Identifier	Number(10)	Specified by external system	ID of current line being processed by input file.	Y	Right/0

Record Name	Field Name	Field Type	Default Value	Description	Required?	Justification/ Padding
	Tax Code	Char(6)	Refer to 'TAXC' code_type for as list of valid types	Tax code (tax code/vat code) to represent whether it is a State, City or some other tax code/vat code.	Y	Left/Blank
	Tax Sign	Char(1)	Refer to 'SIGN' code_type for a list of valid codes.	Sign of the Tax Amount	Y	Left/None
	Tax Amount	Number(20)		Total Tax amount for this item with 4 implied decimal places.	Y	Right/0
	Reference Number 17	Char(30)			N	Left/None
	Reference Number 18	Char(30)			N	Left/None
	Reference Number 19	Char(30)			N	Left/None
	Reference Number 20	Char(30)			N	Left/None
Transaction Tax Attribute	File Type Record Descriptor	Char(5)	TXATT	Identifies file record type	Y	Left/Blank
	File Line Identifier	Number(10)	Specified by external system	ID of current line being processed by input file.	Y	Right/0
	Attribute Type	Char(6)	Refer to 'SAXA' code_type for a list of valid types	Type of transaction tax attribute	Y	Left/Blank
	Attribute Value	Char(120)		Value of transaction tax attribute	Y	Left/Blank
Transaction payment	File Type Record Descriptor	Char(5)	TPYMT	Identifies file record type	Y	Left/Blank
	File Line Identifier	Number(10)	Specified by external system	ID of current line being processed by input file.	Y	Right/0
	Payment Sign	Char(1)	Refer to 'SIGN' code_type for a list of valid codes.	Sign of Deposit Amount.	Y	Left/None
	Payment Amount	Number(20)		Deposit amount paid, with 4 implied decimal places.	Y	Right/0
Transaction Tender	File Type Record Descriptor	Char(5)	TTEND	Identifies file record type	Y	Left/Blank

Record Name	Field Name	Field Type	Default Value	Description	Required?	Justification/ Padding
	File Line Identifier	Number(10)	Specified by external system	ID of current line being processed by input file.	Y	Right/0
	Tender Type Group	Char(6)	Refer to 'TENT' code_type for as list of valid types	High-level grouping of tender types.	Y	Left/Blank
	Tender Type ID	Number(6)	Refer to the pos_tender_type_head table for as list of valid types	Low-level grouping of tender types.	Y	Left/Blank
	Tender Sign	Char(1)	Refer to 'SIGN' code_type for a list of valid codes.	Sign of the value.	Y	Left/None
	Tender Amount	Number(20)		Amount paid with this tender in the transaction with 4 implied decimal places.	Y	Right/0
	Cc_no	Char(40)		Credit card number. Merchandise is not a PCI compliant system. Full credit card numbers are not allowed in Merchandising. The value sent in the RTLOG should be masked so that it contains no more than the first 6 digits and last 4 digits in clear text. The remaining digits should be masked using the character defined in sa_system_options.cc_no_mask_char. If more than the first 6 and last 4 characters exist in any records, the transaction file will be rejected. Alternatively, the credit card number field can be left fully blank.	N	Left/Blank
	Cc_auth_no	Char(16)		Authorization number for a cc	N	Left/Blank
	cc authorization source	Char(6)	Refer to 'CCAS' code_type for as list of valid types		N	Left/Blank

Record Name	Field Name	Field Type	Default Value	Description	Required?	Justification/ Padding
	cc cardholder verification	Char(6)	Refer to 'CCVF' code_type for as list of valid types		N	Left/Blank
	cc expiration date	Char(8)		(YYYYMMDD)	N	Left/Blank
	cc entry mode	Char(6)	Refer to 'CCEM' code_type for as list of valid types	Indicates whether the credit card was swiped, thus automatically entered, or manually keyed.	N	Left/Blank
	cc terminal id	Char(5)		Terminal number transaction was sent from.	N	Left/Blank
	cc special condition	Char(6)	Refer to 'CCSC' code_type for as list of valid types		N	Left/Blank
	cc token	Char(40)		Holds unique token when the tender type used is credit, debit card, PayPal, Fonacot or Others.	N	Left/Blank
	Voucher_no	Char(25)		Gift certificate or credit voucher serial number. Voucher number needs to be included If a voucher is voided from a transaction.	Y if voucher	Right/0
	Coupon Number	Char(40)		Number of a manufacturer's coupon used as a tender.	Y if coupon	Left/Blank
	Coupon Reference Number	Char(16)		Additional information about the coupon, usually contained in a second bar code on the coupon.	Y if coupon	Left/Blank
	Cheque Account Number	Char(30)		Account number of the cheque. The value sent in the RTLOG should be masked.	N	Left/Blank
	Cheque Number	Number(10)		Check Number	Required for tender type 'CHECK'	Right/0

Record Name	Field Name	Field Type	Default Value	Description	Required?	Justification/ Padding
	Identification Method	Char(6)	Refer to 'IDMH' code_type for list of valid types	Identification Method (e.g. Driver's license #, Photo credit card)	N	Left/Blank
	Identification Id	Char(40)		Identification ID (License ID, Photo Card #)	N	Left/Blank
	Original Currency	Char(3)	Refer CURRENCIES table for valid currency codes.	The original currency at which the user made the payment	N	Left/Blank
	Original Currency Amount	Number(20)		Amount paid with this tender in the original currency with 4 implied decimal places.	N	Right/0
	Reference No 9	Char(30)		Number associated with a particular transaction type at the tender level. The sa_reference table defines what this field can contain for each transaction type.	N	Left/Blank
	Reference No 10	Char(30)		Second generic reference no at the tender level.	N	Left/Blank
	Reference No 11	Char(30)		Third generic reference no at the tender level.	N	Left/Blank
	Reference No 12	Char(30)		Fourth generic reference no at the tender level.	N	Left/Blank
Transaction Tender Attribute	File Type Record Descriptor	Char(5)	TTATT	Identifies file record type	Y	Left/Blank
	File Line Identifier	Number(10)	Specified by external system	ID of current line being processed by input file.	Y	Right/0
	Attribute Type	Char(6)	Refer to 'SATA' code_type for a list of valid types	Type of transaction tender attribute	Y	Left/Blank
	Attribute Value	Char(120)		Value of transaction tender attribute	Y	Left/Blank
Transaction Trailer	File Type Record Descriptor	Char(5)	TTAIL	Identifies file record type	Y	Left/Blank
	File Line Identifier	Number(10)	Specified by external system	ID of current line being processed by input file.	Y	Right/0

Record Name	Field Name	Field Type	Default Value	Description	Required?	Justification/ Padding
	Transaction Record Counter	Number(10)		No of records processed in current tran (only records between trans head & tail)		
File Trailer	File Type Record Descriptor	Char(5)	FTAIL	Identifies file record type	Y	Left/Blank
	File Line Identifier	Number(10)	Specified by external system	ID of current line being processed by input file.	Y	Right/0
	File Record Counter	Number(10)		No of transactions processed in current file (only records between file head & tail)	Y	Right/0

The RTLOG file is imported into the Sales Audit tables after validation by the batch program saimptlog. This section describes the requirements and validations performed on the records.

Common Requirements/Validations

This section details the common requirements and validations performed on all transactions. The following sections describe the specific requirements of each type of transaction. If a transaction is not mentioned, it does not have specific requirements.

Table 6-71 Record Type Requirements

Transaction Type	Includes item records?	Includes tender records?	Includes tax records? IG TAX?	Includes customer records?
OPEN	No	No	No	No
NOSALE	No	Optional	No	No
VOID	Optional	Optional	Optional	Optional
PVOID	No	No	No	No
SALE	Optional	Yes	Optional	Optional
RETURN	Yes	Yes	Optional	Optional
EEXCH	Yes	No	Optional	Optional
PAIDIN	No	Yes	No	No
PAIDOU	No	Yes	No	No
PULL	No	Yes	No	No
LOAN	No	Yes	No	No
COND	No	No	No	No
CLOSE	No	No	No	No
TOTAL	No	No	No	No

Table 6-71 (Cont.) Record Type Requirements

Transaction Type	Includes item records?	Includes tender records?	Includes tax records? IG TAX?	Includes customer records?
REFUND	This transaction is not sent through the RTLOG. It is entered at the HQ level. The TITEM and TCUST records are optional. The TTEND record is required. A TTAX record should not be included if IG TAX appears in a transaction if originating system is POS. IG TAX is an item-level tax and TTAX is a transaction-level tax. Either IG TAX or TTAX can be used if originating system is POS but not both. If originating system is OMS, both IG TAX and TTAX can appear but only the one matching the store's tax type will be processed, the other record will be ignored.			
METER	Yes	No	No	No
PUMPT	Yes	No	No	No
TANKDP	Yes	No	No	No
TERM	TERM records are created by saimptlog and then loaded into the database. They do not come from the RTLOG file. They require one TITEM, one TTEND, one TTAX, one TCUST record and one CATT record. IG TAX and one TPYMT which is newly coming up.			
DCLOSE	No	No	No	No
SPLORD	Optional	Yes	Optional	Optional
REOPEN	No	No	No	No

Requirements per Record Type

Table 6-72 Requirements per Record Type

Record Type	Requirements
IDISC	IDISC records must immediately follow their associated TITEM record. IDISC records should be after the ITATT record if it exists.
IGTAX	IGTAX will immediately follow TITEM or ITATT record (if it is present), if discount records are not present. Otherwise it should follow the IDISC or IDATT record (if it is present). Even if IGTAX is coming prior to IDISC, it will be processed, but for maintaining proper format, Sales Audit expects it to come after IDISC.
TTAX	Either this record or IGTAX should appear in the transaction. IGTAX and TTAX cannot be both used at the same time if originating system is POS. If originating system is OMS, both IGTAX and TTAX can appear but only the one matching the store's tax type will be processed, the other record will be ignored.
TPYMT	This record should be right before the TTEND record. It contains the deposit amount for pickup/delivery/layaway orders.
CATT	CATT records must immediately follow their associated TCUST record.
THATT	THATT records must immediately follow their associated THEAD record.
ITATT	ITATT records must immediately follow their associated TITEM record.
IDATT	IDATT records must immediately follow their associated IDISC record.
IXATT	IXATT records must immediately follow their associated IGTAX record.
TXATT	TXATT records must immediately follow their associated TTAX record.

Table 6-72 (Cont.) Requirements per Record Type

Record Type	Requirements
TTATT	TTATT records must immediately follow their associated TTEND record.

Code Type Validations

Table 6-73 Code Type Validations

Record Name	Field Name	Code Type
Transaction Header	Transaction Type	TRAT
	Sub-transaction Type	TRAS
	Reason Code	REAC or values from non_merch_code_head if the transaction type is 'PAIDOU' and the sub-transaction type is 'MV' or 'EV'.
	Value Sign	SIGN
	Vender No	If the transaction type is 'PAIDOU' and the sub-transaction type is 'MV', this field is validated against the supplier table. If the transaction type is 'PAIDOU' and the sub-transaction type is 'EV', this field is validated against the partner table.
	Transaction Processing System	TSYS
Transaction Header Attribute	Attribute Type	SAHA
Transaction Item	Item Type	SAIT
	Item Status	SASI
	Item Number Type	UPCT
	Quantity Sign	SIGN
	Taxable Indicator	YSNO
	Price Override	ORRC
	Reason Code	
	Item Swiped Indicator	YSNO
	Sales Type	SASY
	Return Disposition	INV_STATUS_CODES table
	No Inventory Return	YSNO
	Return Reason Code	SARR
	Fulfillment Loc Type	FLTP
Transaction Item Attribute	Attribute Type	SAIA
Item Discount	RMS Promotion Type	PRMT
	Discount Type	SADT
	Quantity Sign	SIGN
Item Discount Attribute	Attribute Type	SADA

Table 6-73 (Cont.) Code Type Validations

Record Name	Field Name	Code Type
Transaction Customer	Customer ID Type	CIDT
Customer Attribute	Attribute Type	SACA
	Attribute value	Code types from the codes in SACA
Item Tax	Tax Calculation Type	GTTT
	Tax Code	TAXC from the CODE_DETAIL table or VATC from the VAT_CODES table
Item Tax Attribute	Attribute Type	SAXA
Transaction Tax	Tax code	TAXC from the CODE_DETAIL table or VATC from the VAT_CODES table
	Tax sign	SIGN
Transaction Tax Attribute	Attribute Type	SAXA
Transaction Payment	Payment (Deposit Amount) Sign	SIGN
Transaction Tender	Tender Type Group	TENT
	Tender Sign	SIGN
	Tender Type ID	Pos_tender_type_head table
	CC Authorization Source	CCAS
	CC Cardholder Verification	CCVF
	CC Entry Mode	CCEM
Transaction Tender Attribute	CC Special Condition	CCSC
	Attribute Type	SATA

The following dates are validated: Business Date, Transaction Date, and Expiration Date. Also, saimptlog accepts only business dates that are within the PERIOD.VDATE minus the SA_SYSTEM_OPTIONS.DAYS_POST_SALE value.

The store number is validated against the STORE table. Numeric fields are checked for non-numeric characters.

For transactions of type SALE, RETURN and EEXCH, saimptlog checks whether a transaction is in balance. With the introduction of Item level tax and Payment amount lines, the balancing logic has been changed as below. Also with introduction of handling VAT/TAX, the logic of balancing has been modified as below.

- When TAX is on in the system (system_options.default_tax_type = 'SALES'):
 - Transaction Items (Unit Retail * Unit Retail Sign * Quantity) of items which are on Regular Sale, Return, or EEXCH
 - + Item Discounts (Unit Discount Amount * Unit Discount Sign * Quantity) of items which are on Regular Sale, Return, or EEXCH
 - + Item Level Tax (Total Igtax Amount) of items which are on Regular Sale, Return, or EEXCH

+ Transaction Tax (Tax Amount * Tax Sign)
 + Transaction payment (Payment Amount * Payment Sign)
 = Transaction Tenders (Tender Amount * Tender Sign)

saimptlog will populate the Value field (on THEAD) with the transaction's sales value (item value – discount value + tax value) from the preceding calculation if it was not provided in the RTLOG. The following change is made in the sale total balancing: Value field in THEAD will be (item value - discount value + tax value) for items which are on Regular Sale, Return, or EEXCH + payment value.

 **Note:**

If this 'Value' field is being used in creating some totals, then accordingly these totals needs to be modified to accommodate the extra amount coming in.

- When VAT is on in the system (system_options.default_tax_type in 'GTAX', 'SVAT', 'GTS'), look for the store level VAT indicator, which tells whether the unit retail is inclusive or exclusive of VAT. The logic of balancing will vary:

Transaction Items (Unit Retail * Unit Retail Sign * Quantity) of items which are on Regular Sale, Return, or EEXCH

+ Item Discounts (Unit Discount Amount * Unit Discount Sign * Quantity) of items which are on Regular Sale, Return, or EEXCH

+ Item Level Tax (Total Igtax Amount) of items which are on Regular Sale, Return, or EEXCH (when VAT is off at the item level).

+ Transaction Tax (Tax Amount * Tax Sign)

+ Transaction Payment (Payment Amount * Payment Sign)

= Transaction Tenders (Tender Amount * Tender Sign)

Treatment of vouchers:

- If an item sold is as a gift certificate (Transaction Item, Voucher field has a value), the issued information is written to the SA_VOUCHER table.
- If the Transaction Type is RETURN, and the Transaction Tender Type Group is voucher (VOUCH), the issued information is written to the SA_VOUCHER table.
- If the Transaction Type is SALE and the Transaction Tender Type Group is a voucher (VOUCH), the redeemed information is written to the SA_VOUCHER table.
- When a gift certificate is sold, customer information should always be included. A receiving customer name value should be populated in the ref_no5 field, a receiving customer state value should be populated in the ref_no6 field, and a receiving customer country should be populated in the ref_no7 field. These reference fields can be changed by updating the sa_reference table, but the code needs to be modified as well. The expiration date is put in the expiration_date field in the TITEM record.
- If an item sold is a voucher; populate REF_NO8 in TITEM with the voucher type ID. Otherwise the voucher type ID will be defaulted to a gift certificate (4030) in the voucher file.

Other validations and points to consider:

- The salesperson in the TITEM record takes precedence over the salesperson in the THEAD record.
- If an item sold is a sub-transaction (REF) item (Transaction Item, reference item field has a value and item does not), it will be converted to the corresponding transaction level item (ITEM).
- If an item sold is an ITEM (Transaction Item, item field has a value), it will be validated against the Merchandising item tables.
- The corresponding Department, Class, Subclass, and Taxable Indicator will be selected from the Merchandising tables and populated for an item.

The balancing level determines whether the register or the cashier fields are required:

- If the balancing level is 'R'egister, the register field on the THEAD must be populated.
- If the balancing level is 'C'ashier, the cashier field on the THEAD must be populated.
- If the balancing level is 'S'tore, neither field is required to be populated.
- The tax_ind and the item_swiped_ind fields can only accept 'Y' or 'N' values. If an invalid value is passed through the RTLOG, an error will be flagged and the value will be defaulted to 'Y'.

Transaction of Type 'SALE'

A transaction of type SALE is generated whenever an item is sold. If a sale is for an employee, the sub-transaction type is EMP. If it is a drive-off sale, when someone drives off with unpaid gas, the sub-transaction type is DRIVEO. A special type of sale is an odd exchange, sub-transaction type EXCH, where items are sold and returned in the same transaction. If the net value of the exchange is positive, then it is a sale. If the net value is negative, it is a return.

Requirements per record type (other than what is described in the Layout section above):

Table 6-74 Requirements per Record Type

Record Type	Requirements
THEAD	N/A

Table 6-74 (Cont.) Requirements per Record Type

Record Type	Requirements
TITEM	<ul style="list-style-type: none"> <li data-bbox="657 338 1461 478">• Item Status is a required field; it determines whether the item is 'S'old, 'R'eturned, or 'V'oided. If the item status is 'S', the quantity sign is expected to be P. If the item status is 'R', the quantity sign is expected to be N. Also, if the item status is ORI, LIN, ORD, or LCO, the quantity sign should be 'P', and in the case of ORC or LCA, it should be 'N'. Item status ADJ is to support appeasement transactions from OMS, where a customer is partially refunded for their original purchase based on an agreed-to price adjustment, or other reasons. <li data-bbox="657 579 1461 688">• If the item status is V, the quantity sign is the reverse of the quantity sign of the voided item. That is, if an item with status S is voided, the quantity sign would be N. Furthermore, the sum of the quantities being voided cannot exceed the sum of the quantities 'S'old or 'R'eturned. Note: Neither of the two validations are performed by saimptlog, but an audit rule could be created to check this. <li data-bbox="657 762 1461 1037">• The following item statuses are used for handling items on customer order layaway: ORI - Order Initiate ORD - Order Complete ORC - Order Cancel LIN - Layaway Initiate LCA - Layaway Cancel LCO - Layaway Complete <li data-bbox="657 1045 1461 1098">• In a typical sale, the items all have a status of 'S'. In the case of an odd exchange, some items will have a status of 'R'. <li data-bbox="657 1106 1461 1159">• In a typical return, the items all have a status of 'R'. In the case of an odd exchange, some items will have a status of 'S'. <li data-bbox="657 1167 1461 1308">• If an item has status R, then the Return Reason Code field may be populated. If it is, it will be validated against code type 'SARR'. Also, it's better to capture the Return Reason Code in the case of items on ORC or LCA, but it is not mandatory. No validation is kept for these new item statuses for checking of SARR. <li data-bbox="657 1316 1461 1367">• If the price of an item is overridden, the Override Reason and Original Unit Retail fields must be populated.
IDISC	<ul style="list-style-type: none"> <li data-bbox="657 1381 1461 1434">• The Merchandising Promotion Type field must always be populated with values of code type 'PRMT'. <li data-bbox="657 1442 1461 1495">• The Promotion field is validated, when a value is passed, against the promhead table. <li data-bbox="657 1503 1461 1556">• If the promotion is 'In Store' (code 1004), the Discount Type field must be populated with values of code type 'SADT'. <li data-bbox="657 1564 1461 1617">• The Discount Reference Number is a promotion number which is of status 'A', 'E' or 'M'. <li data-bbox="657 1625 1461 1707">• If the Discount Type is 'SCOUP' for Store Coupon, the Coupon Number field must be populated. The Coupon Reference Number field is optional.

Table 6-74 (Cont.) Requirements per Record Type

Record Type	Requirements
IGTAX	<ul style="list-style-type: none"> The IGTAX_CODE field must always be populated depending on the system's default tax type. For a default tax type of 'SALES', this field will be populated with values of code_type TAXC. For a default tax type of SVAT, GTAX, or GTS, this field will be populated with VATC (vat_code from vat_codes table). IGTAX is an Item-level tax. The TAX_AUTHORITY field must always be populated when the default tax type is GTAX. If Item Status is ADJ (appeasement transaction), the IGTAX Tax Code from OMS is TOTAX.
TTAX	<ul style="list-style-type: none"> The TAX_CODE field must always be populated depending on the system's default tax type. For a default tax type of 'SALES', this field will be populated with values of code_type TAXC. For a default tax type of 'GTAX', 'SVAT' or 'GTS', this field will be populated with VATC (vat_code from vat_codes table). TTAX is a Transaction-level tax.
TPYMT	Payment (Deposit amount) sign and Payment (Deposit) amount fields are necessary if this line is appearing. Basically, this is the accumulation of various items being considered in one transaction, which are on pick up/delivery/lay away.
TTEND	If the tender type group is 'COUPON', the Coupon Number field must be populated. The Coupon Reference Number field is optional.

Meaning of reference number fields:



Note:

The meaning of these reference number fields may be changed through the sa_reference table. The transaction type 'SPLORD' is the same as SALE, but the inventory will not be reserved for the orders at its line level.

Table 6-75 Meaning of Reference Number Fields

Transaction Type	Sub-transaction Type	Item Type	Tender Type Group	Reference Number Field	Meaning of Reference Field	Req?
SALE	N/A	N/A	N/A	1	Speed Sale Number	Y
SALE	N/A	GCN	N/A	5	Recipient Name	N
SALE	N/A	GCN	N/A	6	Recipient State	N
SALE	N/A	GCN	N/A	7	Recipient Country	N
SALE	N/A	N/A	CHECK	9	Check Number	N
SALE	N/A	N/A	CHECK	10	Driver's License Number	N

Table 6-75 (Cont.) Meaning of Reference Number Fields

Transaction Type	Sub-transaction Type	Item Type	Tender Type Group	Reference Number Field	Meaning of Reference Field	Req?
SALE	N/A	N/A	CHECK	11	Credit Card Number	N
SALE	DRIVEO	N/A	N/A	1	Incident Number	Y
SALE	EMP	N/A	N/A	3	Employee Number of the employee receiving the goods.	N

Table 6-76 Expected Values for Sign Fields

TRANSACTION TYPE	TITEM.Quantity Sign	TTEND.Tender Sign	TTAX.Tax Sign	IDISC.Quantity Sign
SALE	P if item is sold; N if item is returned; reverse of original item if item is voided.	P	P	P if item is sold; N if item is returned; reverse of original item if item is voided.
SALE	P if item is on ORI, LIN, ORD, or LCO; N if item is on ORC or LCA.	P	P	P if item is on ORI, LIN, ORD, or LCO; N if item is on ORC or LCA.

Transaction of Type 'PVOID'

This transaction is generated at the register when another transaction is being post voided. The orig_tran_no and orig_reg_no fields must be populated with the appropriate information for the transaction being post voided. The PVOID transaction must be associated with the same store day as the original transaction. If the PVOID needs to be generated after the store day is closed, the transaction needs to be created using the forms.

Transaction of Type 'RETURN'

This transaction is generated when a customer returns an item.

This type of transaction has similar record type requirements as a 'SALE' transaction.

Meaning of reference number fields:



Note:

The assumption is that new item statuses will not come under transaction type 'RETURN'.

If a customer wants to return the items (ORI, LIN), these will come under 'SALE' but with item statuses as ORC or LCA.

**Note:**

The meaning of these reference number fields may be changed through the sa_reference table.

Table 6-77 Meaning of Reference Number Fields

Transaction Type	Sub-transaction Type	Reference Number Field	Meaning of Reference Field	Req?
RETURN	N/A	1	Receipt Indicator (Y/N)	Y
RETURN	N/A	2	Refund Reference Number	N
RETURN	EMP	3	Employee Number of the employee returning the goods.	N

Expected values for sign fields:

Table 6-78 Expected Values for Sign Fields

TRANSACTION TYPE	TITEM.Quantity Sign	TTEND.Tender Sign	TTAX.Tax Sign	IDISC.Quantity Sign
RETURN	P if item is sold; N if item is returned; reverse of original item if item is voided.	N	N	P if item is sold; N if item is returned; reverse of original item if item is voided.

Transaction of Type 'SPLORD'

This transaction is generated when a customer picks up an item, which is not in stock. The item status can be 'ORI', 'ORC' or 'ORD'. ('Order Initiate', 'Order Cancel' or 'Order Complete').

Transaction of Type 'EEXCH'

This transaction is generated when there is an even exchange.

This type of transaction has similar record type requirements as a 'SALE' transaction.

It is expected that the number of items returned equals the number of items sold. However, this validation is not performed by saimptlog. An audit rule could be created for this. Saimptlog only expects that there would be at least two item records.

No tender changes hands in this transaction.

Meaning of reference number fields:

 **Note:**

The items, which are on customer order or layaway, should not be come under this transaction type.

The meaning of these reference number fields may be changed through the sa_reference table.

Table 6-79 Meaning of Reference Number Fields

Transaction Type	Sub-transaction Type	Reference Number Field	Meaning of Reference Field	Req?
EEXCH	N/A	1	Receipt Indicator (Y/N)	Y
EEXCH	EMP	3	Employee Number of the employee exchanging the goods.	N

Transaction of Type 'PAIDIN'

This type of transaction has only one TTEND record.

A reason code is required.

Meaning of reference number fields:

 **Note:**

The meaning of these reference number fields may be changed through the sa_reference table.

Table 6-80 Meaning of Reference Number Fields

Reason Code	Reference Number Column	Meaning	Req?
NSF	1	NFS Check Credit Number	N
ACCT	1	Account Number	N

Transaction Type 'PAIDOU'

This type of transaction has only one TTEND record.

A reason code is required (code type REAC). If the sub-transaction type is 'EV' or 'MV', the reason code comes from the non_merch_codes_head table.

If the sub-transaction type is 'EV' or 'MV', then at least one field among the vendor number, vendor invoice number, payment reference number, and proof of delivery number fields should be populated.

If the sub-transaction type is 'EV', then the vendor number comes from the partner table. If the sub-transaction type is 'MV', then the vendor number comes from the supplier table.

Meaning of reference number fields:



Note:

The meaning of these reference number fields may be changed through the sa_reference table.

Table 6-81 Meaning of Reference Number Fields

Sub Transaction Type	Reason Code	Reference Number Column	Meaning	Req?
EV	N/A	2	Personal ID Number	N
EV	N/A	3	Routing Number	N
EV	N/A	4	Account Number	N
N/A	PAYRL	1	Money Order Number	N
N/A	PAYRL	2	Employee Number	N
N/A	INC	1	Incident Number	N

Transaction of Type 'PULL'

This transaction is generated when cash is withdrawn from the register.

This type of transaction has only one TTEND record.

Expected values for sign fields:

Table 6-82 Expected Values for Sign Fields

TRANSACTION TYPE	TITEM.Quantity Sign	TTEND.Tender Sign	TTAX.Tax Sign	IDISC.Quantity Sign
PULL	N/A	N	N/A	N/A

Transaction of Type 'LOAN'

This transaction is generated when cash is added to the register.

This type of transaction has only one TTEND record.

Expected values for sign fields:

Table 6-83 Expected Values for Sign Fields

TRANSACTION TYPE	TITEM.Quantity Sign	TTEND.Tender Sign	TTAX.Tax Sign	IDISC.Quantity Sign
LOAN	N/A	P	N/A	N/A

Transaction of Type 'COND'

This transaction records the condition at the store when it opens. There can be at most one COND record containing weather information and at most one COND record containing temperature information. Both of these pieces of information may be in the same COND record. There may be any number of COND records containing traffic and construction information.

This type of transaction does not have TITEM, IDISC, IGTAX, TTAX, TPYMT, or TTEND records



Note:

The meaning of these reference number fields may be changed through the sa_reference table.

Table 6-84 Meaning of Reference Number Fields

Reference Number Column	Meaning	Req?
1	Weather - code type 'WEAT'	N
2	Temperature - a signed 3 digit number.	N
3	Traffic - code_type 'TRAF'	N
4	Construction - code_type 'CONS'	N

Transaction of Type 'TOTAL'

This transaction records the totals that are reported by the POS and OMS. The value field must be populated. Some systems generate only one transaction number for all totals. In order to avoid duplicate errors being reported, only one total transaction can have a transaction number and the subsequent ones can have blank transaction numbers. In other words, a TOTAL transaction is not required to have a transaction number.

This type of transaction does not have TITEM, IDISC, IGTAX, TTAX, TPYMT, TTEND records.

Transaction of Type 'METER'

This transaction is generated when a meter reading of a fuel pump is taken.

This type of transaction has only TITEM records.

Meaning of reference number fields:



Note:

The meaning of these reference number fields may be changed through the sa_reference table.

Table 6-85 Meaning of Reference Number Fields

Reference Number Column	Meaning	Req?
1	Reading Type: ('A' Adjustment, 'S' shift change, 'P' price change, 'C' store close)	Y
5	Opening Meter Readings	Y
6	Closing Meter Reading	Y
7	If the reading type is 'P' for price change, the old unit retail should be placed here. Decimal places are required.	Y
8	Closing Meter Value	Y

Transaction of Type 'PUMPT'

This transaction is generated when a pump test is performed. This type of transaction has only TITEM records.

Transactions of Type 'TANKDP'

This transaction is generated when a tank dip measurement is taken.

This type of transaction has only TITEM records.

Meaning of reference number fields:

**Note:**

The meaning of these reference number fields may be changed through the sa_reference table.

Table 6-86 Meaning of Reference Number Fields

Reference Number Column	Column Meaning	Req?
1	Tank identifier	Y
5	Dip Type (FUEL, WATER, and so on)	Y
6	Dip Height Major (decimal places required)	Y
7	Dip Height Minor (decimal places required)	Y

Transaction of Type 'DCLOSE'

This transaction is generated when the day closed. The transaction number for this type of transaction has to be blank.

 **Note:**

Vouchers are minimally handled by saimptlog. Voucher information is written to the savouch file which is passed to the program savouch.pc.

- A voucher will appear on the TITEM record only if it was sold. When saimptlog encounters a 'SALE' transaction with a voucher, it writes the voucher to the savouch file as an 'Issued voucher.
- A voucher will be issued when it appears on the TTEND record of transactions of type 'RETURN' and 'PAIDOU'. In other words, saimptlog will write it to the savouch file with status 'I'.
- A voucher will be redeemed when it appears on the TTEND record of transactions of type 'SALE' and 'PAIDIN'. In other words, saimptlog will write it to the savouch file with status 'R'.

Vouchers may not be returned. However, a transaction of type 'PAIDOU' may be generated when the customer exchanges a voucher for another form of tender.

Transaction of Type 'REOPEN'

This transaction is generated when a store day which was closed needs to be reopened to process additional transactions. Transaction number for this type of transaction has to be blank.

Transaction of Type 'OTHER'

This transaction is a generic transaction type to support Micros Xstore integration. This will identify all the other transaction types that are not currently supported. This type of transaction has only THEAD and TTAIL records.

Design Assumptions**Table 6-87 Sales Audit Valid Transaction Types**

Transaction Code	Transaction Type
OPEN	Open
CLOSE	Close
COND	Daily Store Conditions
DCLOSE	Day close indicator
LOAN	Loan
METER	Meter Reading for Fuel
NOSALE	No Sale
PAIDIN	Paid In
PAIDOU	Paid Out
PULL	Pull
PUMPT	Pump Test for Fuel
PVOID	Post Void (A transaction that was rung later into the register to void something that occurred earlier at the same store/day. A post void updates the original transaction's sub-transaction type.)

Table 6-87 (Cont.) Sales Audit Valid Transaction Types

Transaction Code	Transaction Type
REFUND	Return of customer's original check.
RETURN	Return
SALE	Sale
TANKDP	Tank Dip
TOTAL	POS generated totals
EEXCH	Even exchange
VOID	Void (aborted transaction)
OTHER	Others
REOPEN	Reopen Store Day from POS

DCLOSE Transaction Type

When the retailer is sending only one file to the system, SAIMPTLOG.PC marks the store day record in the Sales Audit import log as partially or fully loaded in the database by looking for a transaction type of DCLOSE. However, if the retailer is sending more than one file (as in, for example, a trickle polling situation), the retailer can specify the number of files that the system should expect in combination with the DCLOSE transaction type. This ensures that the system receives all of the files, even if the DCLOSE transaction type is, for some reason, received before the final file.

For example, if 24 files are expected over a given amount of time, and the file with the DCLOSE transaction type is, for some reason, sent before the 24th file, the Merchandising system waits until the last file arrives before marking the store day record as partially or fully loaded in the database.

The import process is completed after SAIMPTLOGFIN.PC has updated the store, data, and audit status of each store day record.

The Reopen Transaction Type

When the retailer is sending transaction of type of REOPEN for store and business day system should expect REOPEN as first transaction in the file before any additional transactions.

When secondary DCLOSE transaction is sent after REOPEN transaction type system should expect count of files since the prior DLCOSE transaction (not the full count for store/day).

SAIMPTLOGFIN.PC batch program would sum up the file counts in case of multiple DCLOSE transactions for the store day and compare against the files loaded in Sales Audit and update the store, data and audit status.

Import Total Value Adjustments From External Systems (saimpadj)

Module Name	saimpadj.pc
Description	Import Total Value Adjustments From External Systems to Sales Audit
Functional Area	Oracle Retail Sales Audit

Module Type Integration
Module Technology ProC
Catalog ID RSA07
Wrapper Script rmswrap_in_rej.ksh

Schedule

Oracle Retail Merchandising Batch Schedule

Design Overview

This module posts external system adjustments to the Sales Audit total value table.

The sales audit adjustments are passed to the module in an external file.

Records that fail necessary validations would be written to the reject file. The input and reject file names are passed as arguments.

Restart/Recovery

Restart/recovery logic for file-based processing is used. The logical unit of work for this module is a parameterized number defined in the restart tables.

Record level locking is done on sa_store_day before updating.

I/O Specification

Integration Type Upload to Sales Audit
File Name Determined by runtime parameters.
Integration Contract IntCon000047

Input File Layout

Table 6-88 Input File Layout

Record Name	Field Name	Field Type	Default Value	Description
FHEAD	File Type Record Descriptor	Char(5)	FHEAD	Identifies file record type (the beginning of the input file).
	File Line Identifier	Number(10)	Sequential number	ID of the current line being read from input file.
	File head descriptor	Char(4)	IMPA	Describes file line type.
	Current date	Char(14)	N/A	File date in YYYYMMDDHH24MISS format.
FDETL	File Type Record Descriptor	Char(5)	FDETL	Identifies the file record type to upload a new deal header.

Table 6-88 (Cont.) Input File Layout

Record Name	Field Name	Field Type	Default Value	Description
	File Line Identifier	Number(10)	Sequential number	ID of the current line being read from input file.
	Data source	Char(6)	N/A	Name of the external system that produced the file.
	New value sign	Char(1)	N/A	Sign(+/-) for the new value.
	New Value	Number(20)	N/A	Value for the total entered by Headquarters user*10000 (4 implied decimal places).
	Total seq no	Number(20)	N/A	Identifies the unique result set for this total ID, total revision, or store/day. Balancing group and index values.
	Store	Number(10)	N/A	Store number for a store/day combination.
	Business Date	Char(8)	N/A	Date for store/day combination.
	Total id	Char(10)	N/A	ID to uniquely identify the total.
	Ref no 1	Char(30)	N/A	The first reference value based by which the total is grouped.
	Ref no 2	Char(30)	N/A	The second reference value based by which the total is grouped.
	Ref no 3	Char(30)	N/A	The third reference value based by which the total is grouped.
FTAIL	File Type record descriptor	Char(5)	FTAIL	Identifies the file record type (the end of the input file).
	File Line Identifier	Number(10)	Sequential number	ID of the current line being read from input file.
	File Record Counter	Number(10)	Sequential number	Number of records/ transactions in the current file (only records between head and tail).

Design Assumptions

N/A

Sales Audit Voucher Upload (savouch)

Module Name	savouch.pc
Description	Sales Audit Voucher Upload
Functional Area	Oracle Retail Sales Audit
Module Type	Integration
Module Technology	ProC
Catalog ID	RSA08
Wrapper Script	batch_savouch.ksh

Schedule

Oracle Retail Merchandising Batch Schedule

Design Overview

Because gift certificates can enter the Sales Audit system as either items or tender, processing must be done to match up the sales and redemptions. This module is used to aggregate gift certificate and voucher records. It compares records in the input files to the database. If a record for the voucher does not exist on the database, the record is inserted. If the voucher already exists on the database, the record should be updated with the appropriate information. The voucher details are updated to SA_VOUCHER table.

Some retailers assign gift certificates to a given store, which means that before a gift certificate is sold at a store, it is assigned to a given store. When a retailer assigns a gift certificate to a given store, a record is written to the database. When the gift certificate is then sold by the store and redeemed by the consumer, this existing record must be updated to include the sale and redemption information. Some retailers choose not to assign gift certificates and instead simply sell gift certificates. In that case, the record will be inserted into the database when the gift certificate is sold and then updated when the gift certificate is redeemed.

Restart/Recovery

Restart/recovery logic for file-based processing is used. Records will be committed to the database when the commit_max_ctr defined in the RESTART_CONTROL table is reached.

I/O Specification

Integration Type	Upload to Sales Audit
File Name	The input file name is not fixed; the input file name is determined by a runtime parameter. Records rejected by the import process are written to a reject file. The reject file name is not fixed; the reject file name is determined by a runtime parameter.
Integration Contract	IntCon000160 (SAVO)

Input File Layout

Table 6-89 Input File Layout

Record Name	Field Name	Field Type	Default Value	Description
FHEAD	Record descriptor/	Char(5)	FHEAD	File head marker.
	Line id	Number(10)	0000000001	Unique line ID.
	Translator id	Char(5)	SAVO	Identifies transaction type.
	File create date	Char(14)	N/A	Vdate in YYYYMMDDHH24MISS format.
FDETL	Business Date	Char(8)	Business Date	Vdate in YYYYMMDD format.
	Record descriptor/	Char(5)	FDETL	File head marker.
	Line id	Number(10)	N/A	Unique line ID.
	Voucher seq Number	Number(20)	N/A	Unique identifier for an entry to the SA_VOUCHER table.
	Voucher No	Char(16)	N/A	Serial Number of the voucher.
	Tender Type Id	Number(6)	N/A	Type of Voucher (Valid values for tender type are maintained in the pos_tender_type_head table with tender_type_group as VOUCH).
	Assigned Date	Char(8)	N/A	Date the voucher was assigned.
	Assigned store	Number(10)	N/A	Store to which the voucher is assigned.
	Issuing Date	Char(8)	N/A	Date this document was issued.
	Issuing store	Number(10)	N/A	Store this document was issued from.
	Issuing Register	Char(5)	N/A	Register this document was issued from.
	Issuing Cashier	Char(10)	N/A	Cashier issuing the document.
	Issued transaction number	Number(20)	N/A	Transaction number at the time of issuance.
	Issued item seq number	Number(4)	N/A	Will hold the item sequence of the item when the voucher is sold as an item (gift voucher).
	Issued tender seq number	Number(4)	N/A	Will hold the tender sequence of the tender when the voucher is sold as a tender (Merchandise Credit).
Issued Amount	Number(20)	N/A	Amount the voucher was issued for*10000 (4 implied decimal places).	
Issued Customer Name	Char(120)	N/A	Name of the customer, who was issued the voucher.	
Issued Customer Addr1	Char(240)	N/A	The address of the customer who was issued the voucher.	

Table 6-89 (Cont.) Input File Layout

Record Name	Field Name	Field Type	Default Value	Description
	Issued Customer Addr2	Char(240)	N/A	The second line address of the customer who was issued the voucher.
	Issued Customer City	Char(120)	N/A	City of the customer, the voucher is issued.
	Issued Customer State	Char(3)	N/A	State of the customer.
	Issued Customer Postal Code	Char(30)	N/A	Postal address of the customer.
	Issued Customer Country	Char(3)	N/A	Country of the customer where the voucher was issued.
	Recipient Name	Char(120)	N/A	Name of the intended recipient.
	Recipient State	Char(3)	N/A	The state of the intended recipient.
	Recipient Country	Char(3)	N/A	The country of the intended recipient.
	Redemption Date	Char(8)	N/A	Date the voucher was redeemed.
	Redemption Store	Number(10)	N/A	Store at which the voucher was redeemed.
	Redemption Register	Char(5)	N/A	Register at which the document was redeemed.
	Redemption cashier	Char(10)	N/A	Cashier redeeming the voucher.
	Redemption tran seq number	Number(20)	N/A	Transaction Number when the document was redeemed.
	Redemption Tender seq number	Number(4)	N/A	This column will hold the tender sequence of the tender within the transaction when a voucher is redeemed as tender.
	Redemption Amount	Number(20)	N/A	Amount the voucher was redeemed for*10000 (4 implied decimal places).
	Expiry Date	Char(8)	N/A	Expiry Date.
	Status	Char(1)	N/A	Indicator showing the document's status - issued or redeemed. Valid values = I - Issued, R - Redeemed.
	Comments	Char(2000)	N/A	Comments.
FTAIL	Record type	Char(5)	FTAIL	Describes file record and marks the end of file.
	Line id	Number(10)	N/A	Unique file line ID.

Table 6-89 (Cont.) Input File Layout

Record Name	Field Name	Field Type	Default Value	Description
	#lines	Number(10)	N/A	Total number of transaction lines in file (not including FHEAD and FTAIL).

Design Assumptions

N/A

Sales Posting

All sales data, whether imported from Sales Audit or directly from POS and OMS solutions, are uploaded into Merchandising using one of the following scheduled inbound integrations are included in this functional area:

- [Process Multiple POSU Files \(uploadsales_all.ksh\)](#)
- [Upload POSU File for Processing \(uploadsales.ksh\)](#)

For more on sales processing, see *Merchandising Operations Guide – Volume 1*.

Process Multiple POSU Files (uploadsales_all.ksh)

Module Name	uploadsales_all.ksh
Description	Process Multiple POSU Files
Functional Area	Sales Posting
Module Type	Integration
Module Technology	Ksh
Catalog ID	RMS157
Wrapper Script	batch_uploadsales.ksh

Schedule

Oracle Retail Merchandising Batch Schedule

Design Overview

The purpose of this script is to execute the uploadsales.ksh module for all POSU files that are for upload. This wrapper will simplify the sales upload process for multiple POSU files, removing the need to call the uploadsales.ksh individually for each file.

Restart/Recovery

N/A

Locking Strategy

N/A

Security Considerations

N/A

Performance Considerations

The number of threads, the amount of waiting time, number for retries, and average volume of data should be considered. `RETRY_WAIT_TIME` shouldn't be increased significantly.

The rows, bindsize and readsize parameter of the `sqlldr` command can be configured for better performance. This gives more control over how many times the inserts are committed/executed.

Security Considerations

N/A

I/O Specification

Integration Type	Upload to Merchandising
File Name	POSU_<store>_<tran_date>_<sysdate>.<thread_val>
Integration Contract	IntCon000044

Input File Layout

Refer to the Input File Layout section in `uploadsales.doc`.

Upload POSU File for Processing (`uploadsales.ksh`)

Module Name	<code>uploadsales.ksh</code>
Description	Upload POSU File for Processing
Functional Area	Sales Posting
Module Type	Integration
Module Technology	Ksh
Catalog ID	RMS112
Wrapper Script	<code>batch_uploadsales.ksh</code>

Schedule

Oracle Retail Merchandising Batch Schedule

Design Overview

The purpose of this module is to upload the contents of the POSU file from Sales Audit or 3rd Party POS to the staging table for further processing.

Restart/Recovery

N/A

Locking Strategy

N/A

Security Considerations

N/A

Performance Considerations

The number of threads, the amount of waiting time, number for retries, and average volume of data should be considered. `RETRY_WAIT_TIME` shouldn't be increased significantly.

The rows, bindsize and readsize parameter of the `sqlldr` command can be configured for better performance. This gives more control over how many times the inserts are committed/ executed.

Security Considerations

N/A

I/O Specification

Integration Type	Upload to Merchandising
File Name	POSU_<store>_<tran_date>_<sysdate>.<thread_val>
Integration Contract	IntCon000044

Input File Layout

Table 6-90 Input File

Record Name	Field Name	Field Type	Default Value	Description
File Header	File Type Record Descriptor	Char(5)	FHEAD	Identifies file record type
	File Line Identifier	Number(10)	Specified by external system	ID of current line being processed by input file
	File Type Definition	Char(4)	POSU	Identifies file as 'POS Upload'
	File Create Date	Char(14)	N/A	Date file was written by external system
	Location Number	Number(10)	N/A	Store identifier
	Vat include indicator	Char(1)	N/A	Determines whether or not the store stores values including vat. Not required but populated by Sales Audit
	Vat region	Number(4)	N/A	Vat region the given location is in. Not required but populated by Sales Audit

Table 6-90 (Cont.) Input File

Record Name	Field Name	Field Type	Default Value	Description
	Currency code	Char(3)	N/A	Currency of the given location. Not required but populated by sales audit
	Currency retail decimals	Number(1)	N/A	Number of decimals supported by given currency for retails. Not required but populated by Sales Audit
Transaction Header	File Type Record Descriptor	Char(5)	THEAD	Identifies transaction record type
	File Line Identifier	Number(10)	Specified by external system	ID of current line being processed by input file
	Transaction Date	Char(14)	Transaction date	Date sale/return transaction was processed at the POS
	Item Type	Char(3)	REF or ITM	Item type will be represented as a REF or ITM
	Item Value	Char(25)	N/A	The ID number of an ITM or REF
	Dept	Number(4)	N/A	Dept of item sold or returned. Not required but populated by Oracle Retail Sales Audit
	Class	Number(4)	N/A	Class of item sold or returned. Not required but populated by Oracle Retail Sales Audit
	Subclass	Number(4)	N/A	Subclass of item sold or returned. Not required but populated by Oracle Retail Sales Audit
	Pack Indicator	Char(1)	N/A	Pack indicator of item sold or returned. Not required but populated by Oracle Retail Sales Audit
	Item level	Number(1)	N/A	Item level of item sold or returned. Not required but populated by Oracle Retail Sales Audit
	Tran level	Number(1)	N/A	Tran level of item sold or returned. Not required but populated by Oracle Retail Sales Audit
	Wastage Type	Char(6)	N/A	Wastage type of item sold or returned. Not required but populated by Oracle Retail Sales Audit

Table 6-90 (Cont.) Input File

Record Name	Field Name	Field Type	Default Value	Description
	Wastage Percent	Number(12)	N/A	Wastage Percent*10000 (4 implied decimal places.), wastage percent of item sold or returned. Not required but populated by Oracle Retail Sales Audit
	Transaction Type	Char(1)	S - sales R - return	Transaction type code to specify whether transaction is a sale or a return
	Drop Shipment Indicator	Char(1)	Y N	Indicates whether the transaction is a drop shipment or not. If it is a drop shipment, indicator will be 'Y'. This field is not required, but will be defaulted to 'N' if blank
	Total Sales Quantity	Number(12)	N/A	Total sales quantity * 10000 (4 implied decimal places), number of units sold at a particular location
	Selling UOM	Char(4)	N/A	UOM at which this item was sold
	Sales Sign	Char(1)	P - positive N - negative	Determines if the Total Sales Quantity and Total Sales Value are positive or negative
	Total Sales Value	Number(20)	N/A	Total Sales Value * 10000 (4 implied decimal places), sales value, net sales value of goods sold
	Last Modified Date	Char(14)	N/A	For VBO future use
	Catchweight Indicator	Char(1)	NULL	Indicates if the item is a catch weight item. Valid values are 'Y' or NULL
	Actual Weight Quantity	Number(12)	NULL	Actual Weight Quantity*10000 (4 implied decimal places), the actual weight of the item, only populated if catchweight_ind = 'Y'
	Sub Trantype Indicator	Char(1)	NULL	Tran type for Sales Audit Valid values are 'A', 'D', NULL
	Total Igtax Value	Number(20)	N/A	Total Igtax Value * 10000 (4 implied decimal places), goods sold or returned

Table 6-90 (Cont.) Input File

Record Name	Field Name	Field Type	Default Value	Description
	Sales Type	Char(1)	N/A	Indicates whether the line item is a Regular Sale, a customer order serviced by OMS (External CO) or a customer order serviced by a store (In Store CO). Valid values are 'R','E', or 'I'
	No Inventory Return Indicator	Char(1)	N/A	Contains an indicator that identifies a return without inventory. This is generally a non-required column, but in case of Returns, this is required. Valid values are 'Y' or 'N'
	Return Disposition	Char(10)	N/A	Contains the disposition code published by RWMS as part of the returns upload to OMS
	Return Warehouse	Number(10)	N/A	Contains the physical warehouse ID for the warehouse identifier where the item was returned
	Customer Order No	Char(48)	N/A	This column contains the customer order number ID.
	Fulfillment Order No	Char(48)	N/A	This column contains the fulfillment order number ID.
	Fulfillment Loc Type	Char(2)	N/A	This column contains the fulfillment location type. Code for the fulfillment loc type from code_detail where code_type = 'FLTP'
	Fulfillment Loc	Number(10)	N/A	This column contains the fulfillment loc ID.
	Orig Store	Number(10)	N/A	This column contains the original store value for a Return transaction.
	POS Tran Id	Number(20)	N/A	This column contains the unique identifier for a sale transaction. This is an Optional field. Customer needs to provide a unique identifier for a sale transaction. If not provided, Merchandising assigns a unique value from a DB sequence.

Table 6-90 (Cont.) Input File

Record Name	Field Name	Field Type	Default Value	Description
	Posting Store	Number(10)		This column contains the store at which the item sale/return should be accounted for in case of cross-store sales happening at co-located stores. It is expected that this field will be populated only for items that are checked out at a different store from the one at which they are originally managed.
Transaction Tax	File Type Record Descriptor	Char(5)	TTAX	Identifies the file record type
	File Line Identifier	Number(10)	Specified by external system	Sequential file line number
	Tax Code	Char(6)	N/A	Holds the tax code associated to the item
	Tax Rate	Number(20)	N/A	Tax rate*10000000000(10 implied decimal places), holds the tax rate for the tax code associated to the item
	Total Tax Value	Number(20)	N/A	Total Tax value*10000(4 implied decimal places), total tax amount for the line item
Transaction Detail	File Type Record Descriptor	Char(5)	TDETL	Identifies transaction record type
	File Line Identifier	Number(10)	Specified by external system	ID of current line being processed by input file
	Promotional Tran Type	Char(6)	N/A	Code for promotional type from code_detail, code_type = 'PRMT'
	Promotion Number	Number(10)	N/A	Promotion number from the Merchandising
	Sales Quantity	Number(12)	N/A	Sales quantity*10000 (4 implied decimal places.), number of units sold in this prom type
	Sales Value	Number(20)	N/A	Sales value*10000 (4 implied decimal places.), value of units sold in this prom type
	Discount Value	Number(20)	NA	Discount quantity*10000 (4 implied decimal places.), value of discount given in this prom type

Table 6-90 (Cont.) Input File

Record Name	Field Name	Field Type	Default Value	Description
	Promotion Component	Number(10)	N/A	Links the promotion to additional pricing attributes
Transaction Trailer	File Type Record Descriptor	Char(5)	TTAIL	Identifies file record type
	File Line Identifier	Number(10)	Specified by external system	ID of current line being processed by input file
	Transaction Count	Number(6)	Specified by external system	Number of TDETL records in this transaction set
File Trailer	File Type Record Descriptor	Char(5)	FTAIL	Identifies file record type
	File Line Identifier	Number(10)	Specified by external system	ID of current line being processed by input file
	File Record Counter	Number(10)	N/A	Number of records/ transactions processed in current file (only records between fhead & ftail)

Fields expected in POSU format based on changes adopted:

	V16	V16+ Customer Order Changes	V19
Fulfillment Order No	No	Yes	Yes
Fulfillment Loc Type	No	Yes	Yes
Fulfillment Loc	No	Yes	Yes
Orig Store	No	Yes	Yes
POS Tran Id	No	No	Yes

Design Assumptions

Multiple taxes for an item if sent from POS to Sales Audit, will be summed to a single tax in Merchandising and assigned one of the applicable tax codes.

Rolling up transactions to the item/store/price point

The program uploadsales.ksh requires that transactions be rolled up the item/store/price point level. The tables below give a hypothetical (though not particularly realistic) example of the type of rollup required by upload_sales.ksh.

Table 6-91 Sales for Item Number 1234 (at one store during one period of the day)

Transaction Number	Number of Items Sold	Amount (in specified currency unit)	Price point (price reason)
167	1	9.99	Regular
395	2	18.00	Promotional

Table 6-91 (Cont.) Sales for Item Number 1234 (at one store during one period of the day)

Transaction Number	Number of Items Sold	Amount (in specified currency unit)	Price point (price reason)
843	1	7.99	Clearance
987	3	27.00	Promotional
1041	1	9.99	Regular
1265	4	31.96	Clearance

 **Note:**

The variation of the price per item in different transactions. This is the result of the price applied at the time of sale—the price point. Now look at the next table that shows the same transactions rolled up by item and price point.

Table 6-92 Sales for Item Number 1234

Number of Items Sold	Price Reason (price point)	Total Amount for Item-Price point (in currency)
2	Regular price	19.98
5	Promotional price	45.00
5	Clearance price	39.95

uploadsales.ksh takes the totals and looks for any discounts for transactions in the POSU file. It applies the discounts to an expected total dollar amount using the price listed for that item from the pricing table (PRICE_HIST). It next compares this expected total against the reported total. If the program finds a discrepancy between the two amounts, it is reported. If the two totals match, the rollup is considered valid. If value-added tax (VAT) is included in any sales transaction amounts, it is removed from those transactions prior to the validation process.

Reject File

The module produces a reject file similar to the input file if it is found to have missing or duplicate FHEAD or FTAIL records. Records in these types of files are loaded to the svc_posupld_load table, but not in the svc_posupld_staging table.

Forecasting

Merchandising has the ability to upload forecast data from an external source. Forecasts can be uploaded by week or by day.

Scheduled forecast integration processes include:

- [Daily Demand Item Forecast Subscription API](#)
- [Weekly Demand Item Forecast Subscription API](#)

- [Weekly/Daily Item Forecast Upload \(load_item_forecast\)](#)

Daily Demand Item Forecast Subscription API

This section describes the Daily Demand Item Forecast Subscription API.

Functional Area

Foundation

Design Overview

This API is used to import daily forecast data from Oracle Retail Demand Forecast Cloud Service (RDFCS) to Merchandising. It uses BDI (Bulk Data Integration), which is an integration layer that facilitates the bulk transfer of information between solutions. On this particular integration, the data flow is from RDFCS to BDI, and then BDI to Merchandising. To accomplish this data transfer, BDI will invoke a Merchandising owned API that will pull data from the BDI integration layer and load into the Merchandising daily forecast table (DAILY_ITEM_FORECAST).



Note:

The job that manages this import is scheduled in RDFCS, rather than as part of the Merchandising batch schedule.

Data Definition XML

The BDI interface staging tables are generated based on the XML schema definition.

Data Flow	Description	XML Schema Definition (XSD)
Daily Demand Item Forecast	Import daily demand item forecast from BDI	DlyDmdFst_Tx_BdiInterfaceModule.xml

Tables

TABLE	SELECT	INSERT	UPDATE	DELETE
DLY_DMND_FRCST_IN	Yes	No	No	No
DAILY_ITEM_FORECAST	No	Yes	Yes	Yes

Weekly Demand Item Forecast Subscription API

This section describes the Weekly Demand Item Forecast Subscription BDI.

Functional Area

Foundation

Design Overview

This API is used to import weekly forecast data from Oracle Retail Demand Forecast Cloud Service (RDFCS) to Merchandising. It uses BDI (Bulk Data Integration), which is an integration layer that facilitates the bulk transfer of information between solutions. On this particular integration stream, the data flow is from RDFCS to BDI, and then BDI to Merchandising. To accomplish this data transfer, BDI will invoke a Merchandising-owned API that will pull data from BDI integration layer BDI table and load into the Merchandising weekly forecast table (ITEM_FORECAST). This process begins by preserving the previous 4 weeks of forecasted sales data in ITEM_FORECAST_HIST and then the ITEM_FORECAST table is truncated. Then the new forecast data is imported.



Note:

The job that manages this import is scheduled in RDFCS, rather than as part of the Merchandising batch schedule.

Data Definition XML

The BDI interface staging tables are generated based on the XML schema definition.

Data Flow	Description	XML Schema Definition (XSD)
Weekly Demand Item Forecast	Import weekly demand item forecast from BDI	WklyDmdFst_Tx_BdiInterfaceModule.xml

Tables

TABLE	SELECT	INSERT	UPDATE	DELETE
WKLY_DMND_FRCST_IN	Yes	No	No	No
ITEM_FORECAST	No	Yes	Yes	Yes
ITEM_FORECAST_HIST	No	Yes	No	Yes

Weekly/Daily Item Forecast Upload (load_item_forecast)

Module Name	load_item_forecast.ksh
Description	Load daily/weekly item forecast from Oracle Retail Demand Forecasting (RDF) Cloud Service
Functional Area	Integration - Forecast
Module Type	Integration
Module Technology	Ksh
Catalog ID	N/A
Wrapper Script	rmswrap_shell_in.ksh

Schedule

Oracle Retail Merchandising Batch Schedule

Design Overview

This script loads item forecast data into the Merchandising forecast tables.

The forecast data comes from Demand Forecasting in a CSV (comma separated) format file. Merchandising expects a single comma-delimited input file (that is, a csv file) in the format specified in the sqlldr control scripts `load_item_forecast.ctl` (for Weekly) and `load_daily_item_forecast.ctl` (for Daily). Please refer to the "[Integration Contract](#)" for more details. A run-time parameter (that is, run type) of 'D' or 'W' indicates whether the Daily or Weekly forecast data is being loaded into Merchandising. If the forecast is a daily forecast, information is written to the `DAILY_ITEM_FORECAST` table. If the forecast is a weekly forecast, information is written to the `ITEM_FORECAST` table. Depending on the run type parameter, the batch truncates the respective forecast table prior to loading.

Restart/Recovery

Evaluate the successful load of the data.

In case of any failures:

SQL load – SQL load dumps invalid records that do not meet certain technical requirements (that is, data type inconsistencies, and so on). The rejected record is written either to a bad file or to a discard file. The discard file contains records that do not satisfy conditions such as missing or invalid record types. Records with other technical issues are written to the bad file.



Note:

A non-fatal code is returned by the program and a message will be written to the log file if reject files are created.

User Action: When such conditions exist, you may update either the bad or the discard file and attempt to reload using the same files. You may also fix the data input file and reload, so that the item forecast tables will be truncated and upload item forecast tables with the corrected the data.

Integration Contract

If a run-time parameter of 'weekly' is used, the input file is a single comma-delimited file (that is, a CSV file):

Field Name	Field Type	Required	Description
EOW_DATE	Date(8)	Yes	Item_forecast.eow_date (YYYYMMDD)
ITEM	Char(25)	Yes	Item_forecast.item

Field Name	Field Type	Required	Description
LOC	Char(10)	Yes	Item_forecast.loc
FORECAST_SALES	Double(14)	Yes	Item_forecast.forecast_sales Note - this field can contain decimal quantities. Unlike quantity fields in Merchandising ProC Batch files, this qty field is not assumed to be extended to significant digits.
FORECAST_STD_DEV	Double(14)	Yes	Item_forecast.forecast_std_dev Note - this field can contain decimal quantities. Unlike quantity fields in Merchandising ProC Batch files, this qty field is not assumed to be extended to significant digits.

If a run-time parameter of 'daily' is used, the input file is a single comma-delimited file (that is, a CSV file):

Field Name	Field Type	Required	Description
DATA_DATE	Date(8)	Yes	Daily_item_forecast.data_date (YYYYMMDD)
ITEM	Char(25)	Yes	Daily_item_forecast.item
LOC	Char(10)	Yes	Daily_item_forecast.loc
FORECAST_SALES	Double(14)	Yes	Daily_item_forecast.forecast_sales Note - this field can contain decimal quantities. Unlike quantity fields in Merchandising ProC Batch files, this qty field is not assumed to be extended to significant digits.
FORECAST_STD_DEV	Double(14)	Yes	Daily_item_forecast.forecast_std_dev Note - this field can contain decimal quantities. Unlike quantity fields in Merchandising ProC Batch files, this qty field is not assumed to be extended to significant digits.

I/O Specification

N/A

Design Assumption

Domain is not a relevant concept any more. Domain_id on ITEM_FORECAST and DAILY_ITEM_FORECAST will always be 1.