Oracle® Retail Enterprise Inventory Cloud Service

Administration Guide

Release 24.0.101.0
F89585-11
2024

ORACLE

Oracle Retail Enterprise Inventory Cloud Service Administration Guide, Release 24.0.101.0

F89585-11

Primary Author: Barbara Clemmer

Contributing Authors: Bipin Pradhan

Contents

Send Us Your Comments

Preface

Audience xxi
Documentation Accessibility xxi
Related Documents xxi
Improved Process for Oracle Retail Documentation Corrections xxi
Oracle Retail Documentation at the Oracle Help Center xxii
Conventions xxii

1 Technical Architecture

Multiple Products 1-1
Logical Model 1-2
Cloud Deployment 1-2
- EICS Client 1-2
- SOCS Mobile Client 1-2
- Web Services 1-2
- WTSS / IDCS or OCI IAM 1-3
- EICS Application Server(s) 1-3
- Oracle DB Server (DBaaS) 1-3
- Client-Server Communication 1-3

Integration 1-3
- Oracle Retail Integration Bus (RIB) 1-3

Deployment 1-3
Deployment - Performance: Bandwidth, Scaling 1-4
- Bandwidth Requirements for Browser Clients 1-4
- Network Latency Constraints 1-4
- Application Server Clustering 1-4
- Database Clustering with RAC 1-4

2 Data Seeding

Data Seeding from Merchandising Foundation Cloud Service 2-1
- Overview 2-1
- When to Run Data Seeding 2-3
- Data Seeding Modules 2-3
- Data Seeding Steps 2-3
  - Pre-requisites for Seeding from Co-Deployed MFCS 2-3
  - Assign Application Roles for Initial Data Loading 2-4
  - Assign Security Permissions for Initial Data Loading 2-4
  - Initial Data Loading System Configuration 2-4
  - Submit Seed 2-4
  - View Selected Module Executions 2-5
  - View Selected Module Execution Details 2-5
  - Re-run Initial Data Loading 2-6
  - Initial Data Loading Post Steps 2-6
- Data Seeding MFCS-SIOCS View Mappings 2-6

Standalone Data Seeding 2-8
- Overview 2-8
- System Admin Parameters 2-9
- Initial Data Loading Process 2-9
- Data Seeding Modules 2-11
- File Layouts 2-13
  - Differentiator File 2-13
  - Differentiator Type File 2-13
  - Item CFA File 2-14
  - Item Component File 2-14
  - Item Description File 2-15
  - Item File 2-15
  - Item Hierarchy File 2-17
  - Item Image File 2-18
  - Item UDA File 2-18
  - Partner Address File 2-19
  - Partner File 2-19
  - Partner Item File 2-20
  - Related Item File 2-21
  - Related Item Type File 2-22
  - Store Address File 2-22
  - Store File 2-23
  - Store Item CFA File 2-24
  - Store Item File 2-24
  - Store Item Price File 2-25

Store Item Price History File 2-27
Store Item Stock File 2-29
Store UIN Admin Item File 2-29
Supplier Address File 2-30
Supplier CFA File 2-31
Supplier File 2-31
Supplier Item Country File 2-32
Supplier Item Country Dimension File 2-33
Supplier Item File 2-34
Supplier Item Manufacture File 2-34
Supplier Item UOM File 2-35
Supplier Organization File 2-35
Transfer Zone File 2-35
UDA File 2-36
UDA LOV File 2-36
UOM Class File 2-37
UOM Conversion File 2-37
Warehouse Address File 2-37
Warehouse Class File 2-38
Warehouse Item File 2-39
Transactional Data Seeding 2-40
Process Flow 2-40
Process Ordering 2-40
Errors and Reprocessing 2-41
Volume Considerations 2-42
Integration Dashboard 2-42
Transaction Data Seeding Modules 2-42
File Layouts 2-42
File Date Requirements 2-43
Allocation File 2-43
DSD File 2-44
Purchase Order File 2-48
Transfer File 2-50
Transfer Shipment File 2-53
Transfer Delivery File 2-57
UIN File 2-61
Vendor Return 2-62
Vendor Shipment 2-64
Vendor Delivery UIN 2-68
Vendor Shipment UIN 2-69
Transfer Shipment UIN 2-70
Transfer Delivery UIN 2-71

Supported Locales 2-72

3 Reporting

Report URL Locations 3-1
Security Considerations 3-3
Previewing a Report 3-3
EICS Operational Reports 3-4
Using BI Publisher for Custom Reports 3-7
Accessing BI Publisher 3-7
Creating a BI Publisher Report 3-7
Displaying a BI Publisher Report 3-8
BI Publisher Reports Delivery Through Object Storage 3-8

4 Internationalization

Supported Locales 4-1
SOCS Client Translations 4-3
EICS Client Translations 4-3
EICS Server Translations 4-3
Translation Topics 4-3
Translation Keys 4-4
Translation Setup Screen 4-4
Bundle Selection Dialog 4-4
Translation Setup Screen 4-5
Translation File Upload 4-7
Report Translations 4-8

5 Batches

Overview 5-1
Batch Admin Users 5-1
Batch Configuration 5-1
Operational Batches 5-2
Auto Inventory Adjustment 5-2
Auto Replenish Capacity 5-4
Auto Ticket Generate 5-4
Auto Ticket Print 5-5
Clearance File Import 5-6
Generate Problem Line Stock Count 5-7
Generate Unit Amount Stock Count 5-8
Generate Unit Stock Count 5-9

Initial Foundation Data File Import 5-10
Initial Inventory Import 5-10
Initial Store Data File Import 5-11
Inventory Extract Export 5-11
Item Basket Maintenance 5-12
Item Price ICL Import Batch 5-13
POS Transaction Import 5-13
Price Change File Import 5-14
Retail Sale Audit Import Batch 5-15
Shelf Replenishment Closure 5-17
Stock Count Authorize Recovery 5-17
Stock Count Auto Cancel 5-18
Stock Count Export 5-19
Stock Count Unit and Amount Snapshot 5-20
Store Order Auto Approve 5-20
Store Order Auto Cancel 5-20
Store Order Auto Generate 5-21
Store Sequence Import 5-21
Third Party Price Import Batch 5-22
Third Party RFID File Import Batch 5-23
Third Party Stock Count Import 5-24
Warehouse Available Inventory Import 5-25
Cleanup Batches 5-26
Cleanup Closed Warehouse Containers 5-33
System Process Batches 5-35
Batch Job Administration 5-37
Job Administration 5-38
Job Admin Screen 5-38
Job Execution Log Screen 5-40
Job Launch Screen 5-41
Job Scheduler 5-43
Job Scheduler Screen 5-45

6 Technical Maintenance Screens

Credential Administration 6-1
Detail Panel 6-3
External Service Administration 6-4
Detail Panel 6-5
File Transfer Service 6-6
Detail Panel 6-7
Job Admin 6-7

Job Scheduler 6-7
MPS Staged Message 6-8
MPS Work Type 6-10
Detail Panel 6-12
MPS Work Type (DCS) 6-13
MPS Work Type (DPS) 6-14
MPS Work Type (RMS) 6-14
MPS Work Type (RIB) 6-15
DCS Work Type 6-16
Operational Issues Screens 6-19
Operational Issues List Screen 6-20
Operational Issues Review 6-21
Operational Review (Data Purge) 6-22
Operational Review (Message Processing) 6-23
Operational Review (Transactional Execution) 6-23
POS Transaction Resolution 6-24
Detail Screen 6-25
Sequence Administration 6-27
Detail Panel 6-28
Integration Dashboard 6-29

7 Integration

Retail Integration Cloud Service (RICS) - based Integration 7-1
Security Considerations 7-2
Customer Orders 7-2
Picking 7-2
Deliveries 7-2
Reverse Picking 7-3
Multi Leg 7-3
RIB Payloads 7-3
Purchase Orders and Vendor Deliveries 7-4
Inventory Adjustments 7-5
Items 7-5
Stock Counts 7-5
Transfers 7-6
Transfer Creation 7-6
Transfer Messages 7-6
Transfer Shipment Creation 7-6
Transfer Receiving 7-6
Transfer Doc 7-9
Transfer Shipment 7-10

Transfer Receiving 7-10
Vendor Return 7-10
RTV Creation 7-10
RTV Shipment 7-10
SOAP Web Services 7-14
Security Considerations 7-14
Functionality 7-15
Available Web Services 7-15
Web Services Basic Design Principles 7-16
Internally Managed vs Externally Managed 7-17
Web Service Operation Basic Design Standards 7-18
Interpreting Validation Errors 7-19
Common Error Codes 7-19
Validation Error (Fault Example) 7-20
Business Error (Fault Example) 7-21
Web Services 7-21
ActivityLock 7-22
FulfillmentOrderDelivery 7-22
FulfillmentOrderPick 7-23
FulfillmentOrderReversePick 7-23
InventoryAdjustment 7-24
ItemBasket 7-25
OrderRequest 7-25
POSTransaction 7-26
ProductGroup 7-26
ProductGroupSchedule 7-27
ReplenishmentGap 7-27
RfidInventory 7-28
ShelfAdjustment 7-28
ShelfReplenishment 7-29
StockCount 7-29
Store 7-30
StoreFulfillmentOrder 7-30
StoreInventory 7-31
StoreInventoryISN 7-31
StoreInventoryUIN 7-32
StoreItem 7-32
StoreItemPrice 7-33
StoreNotification 7-33
StoreShipmentManifest 7-34
StoreShipmentReason 7-34
StoreTicket 7-34

StoreTransfer 7-34
TransferDelivery 7-35
TransferShipment 7-37
VendorDelivery 7-38
VendorReturn 7-39
VendorShipment 7-39
Enterprise Documentation 7-41
REST Web Services 7-41
REST WEB Services Security Considerations 7-41
REST WEB Services Basic Design Principles 7-41
Requests and Responses 7-42
API Versioning 7-42
Content-Type 7-42
JSON Validation 7-42
Synchronous vs Asynchronous 7-42
Configured System Options In EICS 7-43
External vs Internal Attributes 7-43
Dates In Content 7-44
Links In Content 7-44
Hypertext Transfer Protocol Status Codes 7-44
Success Codes 7-44
Client Failure Codes 7-45
System Failure Codes 7-45
JSON Error Element and Error Codes 7-45
Error Attribute Definitions 7-46
Integration Error Codes 7-46
Error Code Data Elements 7-47
REST Service: Activity Lock 7-47
Service Base URL 7-48
API Definitions 7-48
API: Find Lock 7-48
API: Create Lock 7-51
API: Delete Lock 7-52
API: Read Lock 7-52
REST Service: Address 7-53
Service Base URL 7-53
API Definitions 7-53
API: Import Address 7-54
API: Delete Address 7-55
API: Read Address 7-55
REST Service Batch 7-57
Service Base URL 7-57

API Definitions 7-57
API: Execute Batch 7-58
API: Find Batch Jobs 7-58
REST Service: Differentiator 7-60
Service Base URL 7-60
API Definitions 7-60
API: Import Differentiator Types 7-60
API: Import Differentiators 7-61
API: Delete Differentiator Type 7-62
REST Service: Finisher 7-62
Service Base URL 7-62
API Definitions 7-62
API: Import Finishers 7-62
API: Delete Finisher 7-64
API: Import Items 7-64
API: Remove Items 7-65
Rest Service Inventory Adjustment 7-66
Service Base URL 7-66
APIs 7-66
API: Find Adjustments 7-66
API: Read Adjustment 7-67
API: Confirm Adjustment 7-69
API: Cancel Adjustment 7-69
API: Create Adjustment 7-70
REST Service Item 7-71
Service Base URL 7-72
API Definitions 7-72
API: Import Items 7-72
API: Remove Items 7-75
API: Import Hierarchies 7-75
API: Remove Hierarchies 7-76
API: Import Related Items 7-77
API: Delete Related Items 7-78
API: Import Image Urls 7-79
API: Delete Image Urls 7-80
REST Service: Item Inquiry 7-81
Service Base URL 7-81
API Definitions 7-82
API: Find Item by Search Scan 7-82
API: Find Item by Source 7-83
API: Find Items 7-84
REST Service: Item Inventory 7-88

Service Base URL 7-88
API Definitions 7-88
API: Find Available Inventory 7-89
API: Find Inventory 7-91
API: Find Expanded Inventory 7-94
API: Find Future Inventory 7-98
API: Find Inventory in Buddy Stores 7-100
API: Find Inventory in Transfer Zone Stores 7-103
Rest Service: Item ISN 7-106
APIs 7-107
API: Create ISN 7-107
API: Update ISN 7-108
API: Delete ISN 7-109
API: Find ISNs 7-109
API: Read ISN Types 7-110
REST Service: Item Price 7-110
Service Base URL 7-110
API Definitions 7-111
API: findPrices 7-111
API: findPriceByIds 7-112
REST Service Item Uda 7-114
REST Service: Item Price 7-114
Service Base URL 7-114
API Definitions 7-114
API: Import Udas 7-115
API: Delete Udas 7-115
API: Import Item Udas 7-116
API: Delete Item Udas 7-117
API: Find Item Udas 7-118
Rest Service: Item UIN 7-119
Service Base URL 7-119
API Definitions 7-119
REST Service: Activity Lock 7-119
API: createUin 7-120
API: readUin 7-121
API: findUins 7-122
API: findUinLabels 7-122
API: findUinHistory 7-123
API: generateUins 7-124
REST Service: Manifest 7-126
Service Base URL 7-126
API Definitions 7-126

API: Close Manifest 7-126
REST Service: POS Transaction 7-127
Service Base URL 7-127
API Definitions 7-127
API: Import POS Transactions 7-127
REST Service: Reason Code 7-130
Service Base URL 7-131
API Definitions 7-131
API: Find Adjustment Reason Codes 7-131
API: Find Shipment Reason Codes 7-133
REST Service: Stock Count 7-134
Service Base URL 7-134
API Definitions 7-134
API: Snapshot Stock Count 7-135
REST Service: Store 7-135
Service Base URL 7-135
API Definitions 7-135
API: Import Stores 7-136
API: Delete Store 7-137
API: Read Store 7-138
API: Find Stores 7-139
API: Find Associated Stores 7-142
API: Find Auto Receive Stores 7-143
API: Find Transfer Zone Stores 7-145
REST Service: Store Item 7-146
Service Base URL 7-146
API Definitions 7-146
API: Import Store Items 7-147
API: Remove Store Items 7-149
API: Import Replenishment Items 7-150
API: Remove Replenishment Items 7-151
REST Service: Store Order 7-152
APIs 7-152
Rest Service: Store Sequencing 7-156
APIs 7-156
API: Find Sequence Areas 7-157
API: Read Sequence Area 7-158
API: Create Sequence Area 7-159
API: Update Sequence Area 7-161
API: Delete Sequence Area 7-163
API: Create Sequence Item 7-163
API: Update Sequence Item 7-164

API: Delete Sequence Item 7-165
Additional Data Definitions 7-166
REST Service Supplier 7-166
Service Base URL 7-167
API Definitions 7-167
API: Import Suppliers 7-167
API: Delete Supplier 7-169
API: Import Items 7-170
API: Delete Items 7-171
API: Import Item UOMs 7-172
API: Delete Item UOMs 7-173
API: Import Item Countries 7-175
API: Delete Item Countries 7-176
API: Import Item Dimensions 7-177
API: Delete Item Dimensions 7-178
API: Import Item Manufacturers 7-180
API: Delete Item Manufacturers 7-181
REST Service: Ticket 7-182
Service Base URL 7-182
API Definitions 7-182
API: readTicket 7-183
API: findTickets 7-185
API: readArchivedTicket 7-186
API: findArchivedTickets 7-186
API: createTickets 7-188
API: Update Tickets 7-189
API: printTickets 7-191
API: findTicketFormats 7-192
API: findTicketPrinters 7-192
Rest Service: Transfer 7-194
Service Base URL 7-194
API Definitions 7-194
API: Read Transfer 7-194
API: Find Transfer 7-196
API: Request Transfer 7-197
API: Approve Request 7-198
API: Reject Request 7-198
API: Cancel Request 7-199
API: Cancel Transfer 7-199
API: Close Transfer 7-200
API: Find Transfer Contexts 7-200
Additional Data Definitions 7-200

Rest Service: Transfer Shipment 7-202
Service Base URL 7-202
API Definitions 7-202
API: Read Shipment 7-202
API Find Transfer Shipment 7-207
API: Submit Shipment 7-208
API: Cancel Submit Shipment 7-209
API: Dispatch Shipment 7-209
API: createShipment 7-209
API: Cancel Shipment 7-212
API: updateShipment 7-213
API: createCarton 7-214
API: Confirm Carton 7-215
API: Cancel Carton 7-216
API: Open Carton 7-216
API: updateCarton 7-217
Additional Data Definitions 7-218
REST Service Translations 7-219
Service Base URL 7-219
API Definitions 7-219
API: Find Locales 7-219
API: Find Translations 7-220
API: Find Items Descriptions 7-221
Rest Service: Vendor Delivery 7-223
Service Base URL 7-223
API Definitions 7-223
API: Read Delivery 7-223
API: Find Vendor Delivery 7-227
API: Receive Delivery 7-228
API: Confirm Delivery 7-229
API: Reject Delivery 7-229
API: Cancel Delivery 7-229
API: Submit Carton 7-230
API: Cancel Submit Carton 7-230
API: Confirm Carton 7-231
API: Cancel Carton 7-231
API: Open Carton 7-232
Additional Data Definitions 7-232
REST Service Warehouse 7-233
Service Base URL 7-234
API Definitions 7-234
API: Import Warehouses 7-234

API: Delete Warehouse 7-235
API: Import Items 7-236
API: Delete Items 7-237
API: Import Adjustments 7-238
API: Import Inventory 7-239
Sales Integration 7-241
POS and Sales Audit Process Flow 7-241
Sales and Return Processing 7-242
Customer Order Processing 7-242
Item Disposition 7-243
Drop Ship 7-243
Item Types 7-243
RFID 7-243
Integration with Customer Order System 7-244
Integration with Manifesting Systems 7-244
Integration for Notifications 7-245
Integration for Sales Forecast 7-245
Integration for Store Order 7-246
Integration for Ticket Printing 7-246
Rest Ticket Printing 7-246
API Publish Tickets 7-247
Retail Home Integration 7-249
Launch SIOCS from Retail Home 7-249
Tile Reports 7-251
EICS Endpoints 7-251
Shop Floor Out of Stock Items 7-251
Stock Counts - Ready to Authorize 7-252
SIOCS Operational Views 7-252
Launch SIOCS Operational Views from Tile Report 7-253
Subscription Usage Batch 7-253
REST Web Service OAuth2 Requests 7-253
Using the OAuth Protocol 7-253
Obtaining a Token 7-253
Calling the EICS Web Service 7-255

8 Configuration

System Admin Parameters 8-1
Store Admin Parameters 8-42
Permissions 8-64
Data Permissions 8-91

9 File Transfer Services

Overview 9-1
How to Call FTS APIs 9-2
- Service Base URL 9-3
- FTS Bucket Name 9-3
- FTS Endpoints 9-3
- Preparing for Authorization 9-4
- Retrieving Access Client Token 9-6
- FTS API Call Common Headers 9-6
- How to Use FTS API to find Object Storage Prefixes 9-7
- How to Use FTS APIs to Upload Files to Object Storage 9-7
  - Step1: Request upload PAR 9-7
  - Step2: Use PAR to upload data files to Object Storage 9-7
- How to Use FTS API to List Files in Object Storage 9-8
- How to Use FTS APIs to Download Files from Object Storage 9-8
  - Step1: Find what files are available for downloads 9-8
  - Step2: Request Download PAR for downloading data files from Object Storage 9-8
  - Step3: Download the file using the par returned from step2 9-9

Handling Import Data Files 9-9
- Supported Import Data Files 9-9
- Upload Import Data Files to Object Storage 9-10

Handling Export Data Files 9-10
- Supported Export Data Files 9-11
- Steps to Download Export Data Files from Object Storage 9-11

File Transfer Service UI 9-11
FTS API Specifications 9-14
- Ping 9-14
- List Prefixes 9-14
- List Files 9-15
- Move Files 9-17
- Delete Files 9-18
- Request Upload PAR 9-19
- Request Download PAR 9-20

File Transfer Service Troubleshooting 9-21
Test FTS API using Postman 9-22
- Step 1: Get Client Access Token 9-22
- Step 2: Call FTS Endpoints 9-22

A Appendix: Report Formats

Reports A-1

Report Formats A-2

B Appendix: Batch File Layout Specifications

Clearance Import File Specification B-1
Filename Format B-1
File Layout B-1
Inventory Extract B-2
Filename Format B-2
File Layout B-2
POS Transaction Import File Specification B-3
Filename Format B-3
Zip File Format B-4
File Layout B-4
Price Change Import File Specification B-5
Filename Format B-5
File Layout B-6
Retail Sale Audit Import File Specification B-7
Filename Format B-7
File Format B-7
File Layout B-7
Stock Count Results Export File Specification B-10
Store Sequence Data Import File Specification B-11
Filename Format B-11
Zip Filename Format B-12
File Format B-12
File Layout B-12
Third Party RFID File Specification B-13
Filename Format B-13
File Layout B-13
Third Party Price File Layout B-14
Filename Format B-14
Third Party Initial Inventory File Layout B-16
Filename Format B-17
DataFilename format B-17
File Layout B-17
Third Party Stock Count Import File Layout B-18
Filename Format B-18
Data Filename Format B-18
File Layout B-18
Warehouse Available Inventory Import Specification B-19
Filename Format B-19

File Layout B-19

C Appendix: Auto-Authorized Third-Party Stock Count Process Overview

Third Party Processing C-3
Third Party Recovery C-3

D Appendix: Unit and Amount Stock Counts Export

Export File Layout D-1
Export File Location D-1
Export File Name D-1

E Appendix: UPC Barcode

Differences between UPC-A and UPC-E E-1
Conversion between UPC-A and UPC-E E-2

F Appendix: EICS Provided URLs

EICS web-client URL F-1
SOCS (connections config) URL F-1
EICS Web Service URLs F-1
EICS-RICS Integration URLs F-3
EICS-RICS Message Publishing F-3
Setting Up RIB Message Injector F-4
BI Related URL F-4
ORDS (Apex Data Viewer) URL F-4

G Appendix: MPS Message Types

DCS Message Types G-1
DCS Message Type Source G-2

Send Us Your Comments

Oracle Retail Enterprise Inventory Cloud Service Administration Guide, Release 24.0.101.0

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 Help Center (OHC) website. It contains the most current Documentation Library plus all documents revised or released recently.

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

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

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

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

Preface

This document describes the administration tasks for Oracle Retail Enterprise Inventory Cloud Service.

Audience

This document is intended for administrators.

Documentation Accessibility

For information about Oracle's commitment to accessibility, visit the Oracle Accessibility Program website at https://docs.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 https://docs.oracle.com/pls/topic/lookup?ctx=acc&id=info or visit https://docs.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 Store Inventory Operations Cloud Services Release 24.0.101.0 documentation set:

- *Oracle Retail Store Inventory Operations Cloud Services Release Notes*
- *Oracle Retail Store Inventory Operations Cloud Services Implementation Guide*
- *Oracle Retail Store Inventory Operations Cloud Services Data Model*
- *Oracle Retail Enterprise Inventory Cloud Service Security Guide*
- *Oracle Retail Enterprise Inventory Cloud Service User Guide*
- *Oracle Retail Store Operations Cloud Service User Guide*
- *Oracle Retail Store Operations Cloud Service Mobile Guide*

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 at the Oracle Help Center (OHC) website, 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 at the Oracle Help Center at the following URL:

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

An updated version of the applicable Oracle Retail document is indicated by Oracle part number, as well as print date (month and year). An updated version uses the same part number, with a higher-numbered suffix. For example, part number F123456-02 is an updated version of a document with part number F123456-01.

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

Oracle Retail Documentation at the Oracle Help Center

Oracle Retail product documentation is available on the following website:

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

(Data Model documents are not available through Oracle Help Center. You can obtain them through My Oracle Support.)

Conventions

The following text conventions are used in this document:

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

1

Technical Architecture

This chapter describes the overall software architecture, offering a high-level discussion of the general structure of the system.

There could be underlying version updates to the technical stack (DB 19c, Web Logic, updated versions of UI libraries, Fusion middle ware libraries and so on.)

Multiple Products

EICS (Enterprise Inventory Cloud Service) and SOCS (Store Operations Cloud Service) are two separately licensed products.

EICS includes:

- EICS Browser Client
- EICS Web Services
- EICS Server Tier
- EICS Database tier with data access code, batches, reports

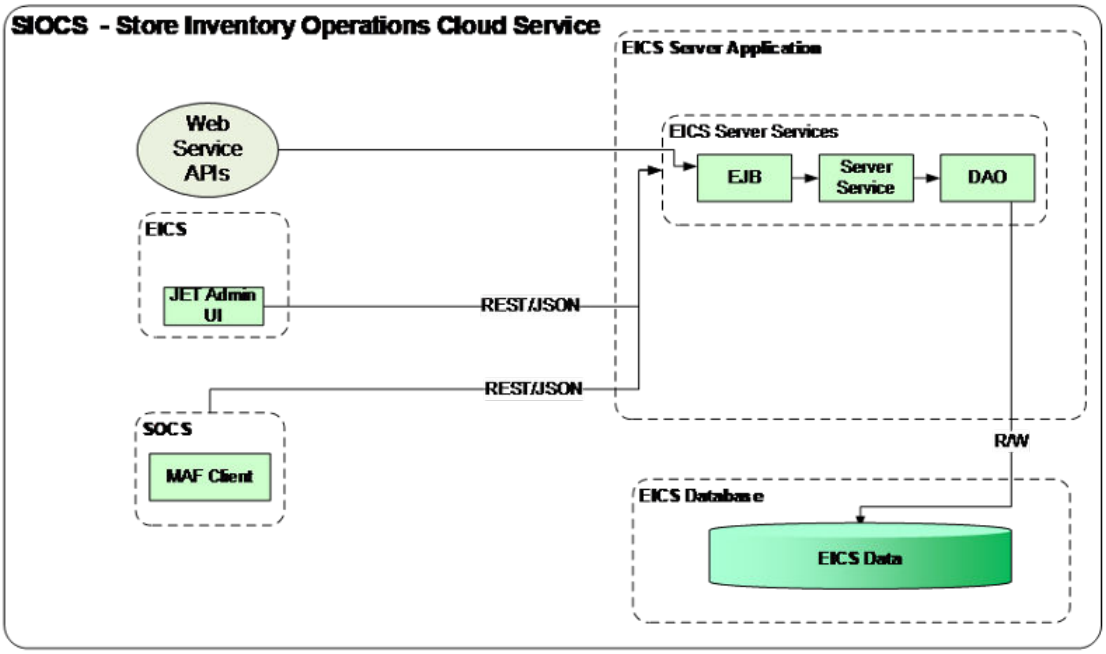
SOCS includes:

- Oracle MAF Client

To use SOCS, EICS needs to be deployed.

Logical Model

Figure 1-1 Logical Model



Cloud Deployment

EICS Client

Oracle JET based browser application that allows the user to perform a wide range of administrative functions.

SOCS Mobile Client

The mobile client provides all day-to-day transactional workflows within an Oracle Mobile Application Framework (MAF) platform. MAF is a hybrid-mobile platform that supports both iOS and Android devices. For more details, please see *Oracle Retail Store Operations Cloud Service Mobile Guide*.

Web Services

There is no GUI for the SOAP web services APIs that are provided by EICS. These APIs allow customers to create or develop applications or add-ons that can replicate some or all the steps of a transaction workflow.

We have started providing REST web service APIs as well.

These REST APIs will be in addition to existing SOAP APIs.

Please note that once a comparable REST API is added, SOAP service deprecation clock will start ticking.

WTSS / IDCS or OCI IAM

WTSS: Web Traffic Security Service

Integration Cloud Services uses Oracle Identity Cloud Service (IDCS) as its identity provider (IDP) or Oracle Cloud Infrastructure Identity and Access Management (OCI IAM) as its identify provider (IDP).

EICS Application Server(s)

Server deployed as a J2EE application inside the WebLogic Application Server.

Oracle DB Server (DBaaS)

Contains EICS schema. Uses JDBC to access data from the database.

WebLogic application server provides a connection pool to use database resources in an efficient fashion.

PL/SQL stored procedures are also used for high volume batch processing.

Client-Server Communication

Client(s) use REST service calls to access the server.

External systems may use SOAP service calls to access the server.

All transactions are container managed.

Performance is sensitive to network latency (hence compression from client to server).

Integration

Oracle Retail Integration Cloud Server (RICS) is used for integration between multiple systems within the cloud as well as deployment of public web services by EICS and other systems.

Oracle Retail Integration Bus (RIB)

The RIB is a near-real time, message based communication queue. Payloads are delivered in an asynchronous fashion between multiple systems on the enterprise in a non-blocking (fire and forget) manner. This broadcast of notifications is subscribed to by each application interested in an event notification.

Deployment

EICS and SOCS have a distributed deployment model with browser and mobile devices running at stores, connecting with server and database hosted at corporate. The central server deployment allows real-time inventory queries for stock-on-hand positions across the enterprise but requires a fairly robust network connection between store and corporate environments.

Deployment - Performance: Bandwidth, Scaling

Bandwidth Requirements for Browser Clients

Installations with less than 128 KB bandwidth available between the device containing the browser or the mobile application and the data center are not recommended or supported. Limiting the client to less than 128 KB total available bandwidth causes unpredictable network utilization spikes, and the performance of the client degrades below requirements established for the product.

Network Latency Constraints

EICS is also sensitive to the network latency between the browser or mobile device and the data center. Oracle Retail does not recommend or support installations with more than 100 ms total round-trip network latency between the client device and the data center. Latency beyond the 100 ms limit causes unpredictable network utilization spikes, and the performance of the client degrades below requirements established for the product. The 100 ms limitation provides reasonable, predictable performance and network utilization for transactions.

Application Server Clustering

EICS supports application server clustering with a load balancer for Oracle WebLogic.

The EICS Server can run inside an application server cluster, but the load must be distributed among the nodes in the cluster.

It is also possible to configure a single application server instance with multiple JVMs inside the application server instance. When doing this, an upper limit of 1.5 to 2 JVMs per server CPU is recommended.

Clustering for Oracle WebLogic Server is also supported through a load balancer.

Database Clustering with RAC

EICS supports a RAC-enabled database for performance.

2

Data Seeding

SIOCS needs merchandising foundation data (stores, items, initial inventory positions, suppliers, and so on) to function.

Initial inventory data seeding is applicable for new or fresh full SIOCS installation. After the initial set of data is seeded into SIOCS, subsequent inventory changes are communicated via Oracle Retail Integration Cloud Service.

Data Seeding from Merchandising Foundation Cloud Service

When SIOCS and MFCS (Merchandising Foundation Cloud Service) are co-deployed in the same database Container, the data seeding process imports data from the MFCS database into the SIOCS database, and this seeding is an MDI-Based data seeding.

Standalone Data Seeding

In SIOCS standalone installation, SIOCS provides standalone data seeding to seed external data into SIOCS. See Standalone Data Seeding for details.

Transactional Data Seeding

See Transactional Data Seeding for more details.

Data Seeding from Merchandising Foundation Cloud Service

This section contains the following:

- Overview
- When to Run Data Seeding
- Data Seeding Modules
- Data Seeding Steps
- Data Seeding MFCS-SIOCS View Mappings

Overview

Merchandising Data Integration (MDI) Based Data Seeding process is seeding foundation data from a co-deployed MFCS (Merchandising Foundation Cloud Service) database into SIOCS.

Data Seeding contains 39 modules and are grouped into nine (9) groups based on dependency and functionality.

Data seeding can be used for a variety of use cases:

- Loading just the Foundation data from Merchandising
- Loading all stores data
- Loading a single store data

Table 2-1 Initial Data Loading Groups

Group Number	Data Group	Module	Description
2	Item	Item Header	Initial Data Loading Groups
2		Item CFA	Import item custom flexible attribute data.
2		Item Translation	Import item description translation data.
3		Item Image	Import item image URL data.
3		Item UDA	Import item User Defined Attribute data.
1		Item Hierarchy	Import item merchandise hierarchy data, for example, department, class and subclasses.
3		Pack Item	Import item pack item component data.
3		Related Item	Import related item type data.
4		Related Item Detail	Import related item detail data.
2	Miscellaneous	Differentiator	Import item differentiation data.
1		Differentiator Type	Import item differentiator type data, for example, color, size, and so on.
1		Transfer Zone	Import transfer zone data.
1		UDA	Import User Defined Attribute data.
2		UDA Values	Import User Defined Attribute Value data.
1		UOM Class	Import Unit Of Measure class data.
2		UOM Conversion	Import Unit Of Measure conversion data.
	Store	Store Item Stock	Import store item stock record data. Can be run by a store, or list of stores.
5		Store	Import store data. Can be run by a store, or list of stores.
5		Store Address	Import store address data. Can be run by a store, or list of stores.
6		Store Item	Import store item data. Can be run by a store, or list of stores.
7		Store Item CFA	Import store item custom defined attributes.
7		Store Item Stock	Import store item stock data.
8		Store Item Price	Import store item price data. Can be run by a store, or list of stores.
9		Store Item Price History	Import store item price history data. Can be run by a store, or list of stores.
9		Store UIN Admin Item	Import UIN admin item foundation data. Only applicable if UIN is enabled for the store. Can be run by a store, or list of stores.
4	Supplier	Item Supplier Country Dimension	Import item supplier country dimension data.
4		Item Supplier Manufacturer Country	Import item supplier manufacture country data.
4		Item Supplier Country	Import item supplier country data.
4		Item Supplier UOM	Import item supplier UOM data.

Table 2-1 (Cont.) Initial Data Loading Groups

Group Number	Data Group	Module	Description
4		Item Supplier	Import item supplier data.
1		Partner	Importer partner data.
1		Partner Address	Import partner address data.
		Partner Item	Import partner item data.
2		Supplier Organization Unit	Import supplier organization unit data.
1		Supplier	Import supplier data.
1		Supplier Address	Import supplier address data.
1		Supplier CFA	Import supplier custom flex attributes
1	Warehouse	Warehouse	Import warehouse data.
1		Warehouse Address	Import warehouse address data.
3		Warehouse Item	Import warehouse item data.

When to Run Data Seeding

Typically, data seeding on fresh installed SIOCS environment.

Data Seeding Modules

Data seeding modules are grouped into 9 data groups:

See Table 2-3 Data Seeding MFCS-SIOCS View Mappings for additional information.

Data Seeding Steps

- Pre-requisites for Seeding from Co-Deployed MFCS
- Assign Application Roles for Initial Data Loading
- Assign Security Permissions for Initial Data Loading
- Initial Data Loading System Configuration
- Submit Seed
- View Selected Module Executions
- View Selected Module Execution Details
- Re-run Initial Data Loading
- Initial Data Loading Post Steps

Pre-requisites for Seeding from Co-Deployed MFCS

Prior to running data seeding, the following requirements must be met:

- MFCS database is installed
- MFCS foundation data setup is completed

- MFCS and SIOCS are installed in the same pluggable database with different schemas

Assign Application Roles for Initial Data Loading

Users need to have the following Application roles assigned in IDCS or OCI IAM:

{SIOCS Primary APP}.admin_users for example,

RGBU_SIOCS_CFS_EICS.admin_users

{SIOCS Primary APP}.batch_users for example,

RGBU_SIOCS_CFS_EICS.batch_users

Assign Security Permissions for Initial Data Loading

Table 2-2 Security Permissions for Initial Data Loading

Name	Description
Access Initial Data Load	With this permission the user will have access to the Initial Data Load screen. Without this permission the user will not have access to the Initial Data Load screen.
Submit Initial Data Load	With this permission the user will have the permission to submit seed. Without this permission, the Submit Seed button will be disabled for the user.
Delete Initial Data Load	With this permission the user will have the permission to delete seeded data. Without this permission, the Delete Seed button will be disabled for the user.

Initial Data Loading System Configuration

To seed initial inventory foundation data from sourcing system directly into destination application tables, an application implementation consultant must perform the following configuration steps:

Login SIOCS Application as a user who are assigned proper app roles and security permissions, see App Roles and Security Permission Section for details.

To seed data from co-deployed Merchandising data integration shared database, set system configuration values as shown below:

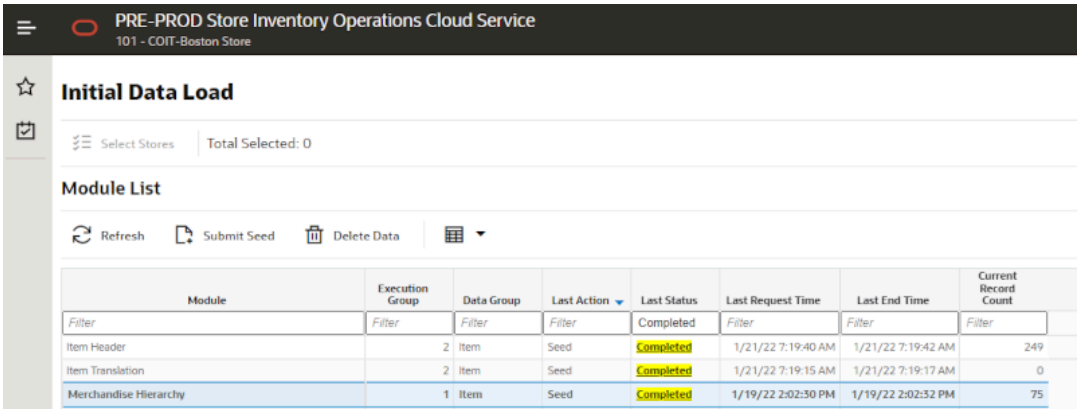
1. Set **Initial Data Load Seed** to **Yes**.
2. Set **Initial Seed Foundation** to **Yes**.
3. Set **Initial Data Load Seed Foundation Data** to **Yes**.

Submit Seed

To start the initial data loading, perform following steps:

1. Login SIOCS Application as app admin user.
2. Navigate to Admin - Technical Maintenance - Initial Data Loading Screen.

Figure 2-1 Initial Data Loading Screen



3. Filter the modules by execution group, start with group 1.
4. Select the module group, then click **Submit Seed** button.

> **Note:**
>
> To run data seeding for store related groups, user will need to select sourcing stores using **Select Store** button.

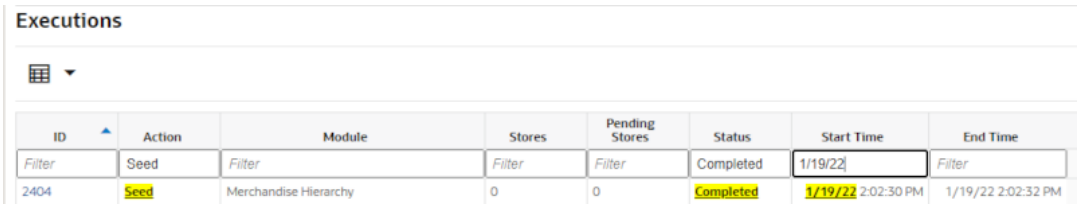
5. Once modules for selected group are completed, then proceed to the next group.

View Selected Module Executions

To view data loading log for the selected module:

1. Click the executed module record from the module list panel.
2. Scroll down to the **Executions** panel to view execution details for the selected module.

Figure 2-2 Initial Data Loading Execution Panel

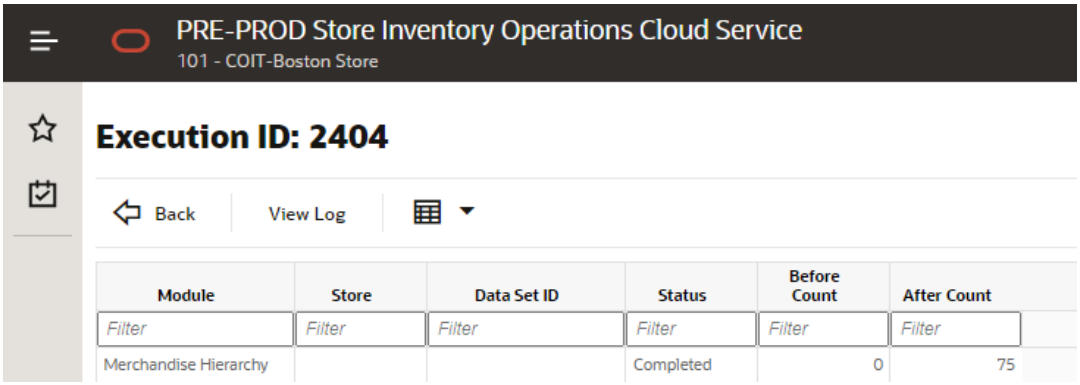


View Selected Module Execution Details

To view data loading execution details:

1. Click the executed module record from the module list panel.
2. Select record in the **Execution** panel.
3. Click the ID link to navigate to the **Execution Detail** screen.

Figure 2-3 Initial Data Loading Execution Details Screen



Re-run Initial Data Loading

In the event of failures, you may need to re-run the data seeding after correcting the errors.

To re-run data seeding:

1. Select the module, then click the **Delete Data** button.
2. After delete process to complete, select the module, then click **Submit Seed** button.

Initial Data Loading Post Steps

1. Verify data seeded into SIOCS application tables without error.
2. Set **Initial Data Load Seed** to **No**.
3. Set **Initial Seed Foundation** to **No**.
4. Set **Initial Data Load Seed Foundation Data** to **No**.

> **Note:**
>
> For stores which need to be rolled out by phases, the value can be set back to **Yes** before loading another set of stores and set to **No** after all stores are seeded from sourcing system.

Data Seeding MFCS-SIOCS View Mappings

Table 2-3 Data Seeding MFCS-SIOCS View Mappings

Seeding Module	SIOCS Target Table	SIOCS View	MFCS[1]
Differentiator Type	DIFFERENTIATOR_TYPE	IDLV_DIFFERENTIATOR_TYPE	V_RMS_SIM_DIFF_TYPE
Differentiator	DIFFERENTIATOR	IDLV_DIFFERENTIATOR	V_RMS_SIM_DIFF
Item	ITEM	IDLV_ITEM	V_RMS_SIM_ITEM_MASTER
Item CFA	ITEM_CFA	IDLV_ITEM_CFA	V_RMS_SIM_ITEM_MASTER_CFA_EXT

Table 2-3 (Cont.) Data Seeding MFCS-SIOCS View Mappings

Seeding Module	SIOCS Target Table	SIOCS View	MFCS[1]
Item Description Translation	ITEM_DESCRIPTION	IDLV_ITEM_DESCRIPTION	V_RMS_SIM_ITEM_MASTER_TL
Item Image	ITEM_IMAGE	IDLV_ITEM_IMAGE	V_RMS_SIM_ITEM_IMAGE
Item Supp Country Dim	SUPPLIER_ITEM_COUNTRY_DIM	IDLV_SUPPLIER_ITEM_COUNTRY_DIM	V_RMS_SIM_ITEM_SUPP_CTRY_DIM
Item Supp Man. Country	SUPPLIER_ITEM_MANUFACTURE	IDLV_SUPPLIER_ITEM_MANUFACTURE	V_RMS_SIM_ITEM_SUPP_MANU_CTRY
Item Supp Country	SUPPLIER_ITEM_COUNTRY	IDLV_SUPPLIER_ITEM_COUNTRY	V_RMS_SIM_ITEM_SUPP_CTRY
Item Supplier	SUPPLIER_ITEM	IDLV_SUPPLIER_ITEM	V_RMS_SIM_ITEM_SUPPLIER
Item Supplier UOM	SUPPLIER_ITEM_UOM	IDLV_SUPPLIER_ITEM_UOM	V_RMS_SIM_ITEM_SUPP_UOM
Item UDA	ITEM_UDA	IDLV_ITEM_UDA	V_RMS_SIM_UDA_ITEM_DATE V_RMS_SIM_UDA_ITEM_FF V_RMS_SIM_UDA_ITEM_LOV
Merch Hier	ITEM_HIERARCHY	IDLV_ITEM_HIER	V_RMS_SIM_MERCH_HIER
Pack Item	ITEM_COMPONENT	IDLV_ITEM_COMPONENT	V_RMS_SIM_PACKITEM
Partner	PARTNER	IDLV_PARTNER	V_RMS_SIM_EXTERNAL_FINISHER
Partner Address	ADDRESS	IDLV_ADDRESS	V_RMS_SIM_ADDR
Partner Item	PARTNER_ITEM	IDLV_PARTNER_ITEM	V_RMS_SIM_ITEM_LOC (loc_type = 'E' --external finisher)
Partner Org Unit	SUPPLIER_ORGANIZATION	IDLV_SUPPLIER_ORGANIZATION	V_RMS_SIM_PARTNER_ORG_UNIT
Price History	ITEM_PRICE_HISTORY	IDLV_STORE_ITEM_PRICE_HIST	V_RMS_SIM_PRICE_HIST
Related Item	RELATED_ITEM_TYPE	IDLV_RELATED_ITEM_TYPE	V_RMS_SIM_RELATED_ITEM_HEAD
Related Item Detail	RELATED_ITEM	IDLV_RELATED_ITEM	V_RMS_SIM_RELATED_ITEM_DETAIL
Store	STORE	IDLV_STORE	V_RMS_SIM_STORE
Store Address	ADDRESS	IDLV_ADDRESS	V_RMS_SIM_ADDR
Store Item	STORE_ITEM	IDLV_STORE_ITEM	V_RMS_SIM_STORE_ITEM V_RMS_SIM_REPL_ITEM_LOC
Store Uin Admin Item	STORE_UIN_ADMIN_ITEM	IDLV_STORE_UIN_ADMIN_ITEM	V_RMS_SIM_STORE_ITEM
Store Item CFA	STORE_ITEM_CFA	IDLV_STORE_ITEM_CFA	V_RMS_SIM_ITEM_LOC_CFA_EXT

Table 2-3 (Cont.) Data Seeding MFCS-SIOCS View Mappings

Seeding Module	SIOCS Target Table	SIOCS View	MFCS[1]
Store Item Price	ITEM_PRICE	IDLV_STORE_ITEM_PRICE	V_RMS_SIM_STORE_ITEM
Store Item Stock	STORE_ITEM_STOCK STORE_ITEM_STOCK_NONSELL	IDLV_STORE_ITEM_STOCK IDLV_STORE_ITEM_STOCK_NONSELL	V_RMS_SIM_STORE_ITEM_SOH
Supplier	SUPPLIER	IDLV_SUPPLIER	V_RMS_SIM_SUPS
Supplier CFA	SUPPLIER_CFA	IDLV_SUPPLIER_CFA	V_RMS_SIM_SUPS_CFA_EXT
Supplier Address	ADDRESS	IDLV_ADDRESS	V_RMS_SIM_ADDR
Transfer Zone	STORE_TRANSFER_ZONE	IDLV_TRANSFER_ZONE	V_RMS_SIM_TSFZONE
UDA	UDA	IDLV_UDA	V_RMS_SIM_UDA
UDA LOV	UDA LOV	IDLV_UDA_LOV	V_RMS_SIM_UDA_VALUES
UOM Class	UOM_CLASS	IDLV_UOM_CLASS	V_RMS_SIM_UOM_CLASS
UOM Conversion	UOM_CONVERSION	IDLV_UOM_CONVERSION	V_RMS_SIM_UOM_CONVERSION
Warehouse	WAREHOUSEWAREHOUSE_VIRTUAL	IDLV_WAREHOUSE IDLV_WAREHOUSE_VIRTUAL	V_RMS_SIM_WH
Warehouse Address	ADDRESS	IDLV_ADDRESS	V_RMS_SIM_ADDR
Warehouse Item	WAREHOUSE_ITEM	IDLV_WAREHOUSE_ITEM	V_RMS_SIM_ITEM_LOC (loc_type = 'W')

[1]MFCS view: only applicable for data seeding source is MFCS on a co-deployed Oracle PDB.

Standalone Data Seeding

This section contains the following:

- Overview
- System Admin Parameters
- Initial Data Loading Process
- Data Seeding Modules
- File Layouts
- Supported Locales

Overview

Data seeding in a SIOCS Standalone installation is achieved by uploading data in CSV (comma-separated values) files to Object Storage via FTS (File Transfer Service). The Initial Foundation Data File Import and Initial Store Data File Import batch jobs then download the relevant files from Object Storage (see Data Seeding Modules) and import the data into SIOCS.

The Initial Data Load UI (see Data Seeding from Merchandising Foundation Cloud Service) can be re-used in a SIOCS Standalone installation to view the status of each Data Seeding Module (see below), view any errors associated with the processing of files for that module, and to Delete Data for that module.

> **Note:**
>
> The **Submit Seed** button will be disabled in a SIOCS Standalone installation: data seeding will be initiated by running the **Initial Foundation Data File Import** and **Initial Store Data File Import** batch jobs from the Job Admin UI (see Batches).

System Admin Parameters

Table 2-4 System Admin Parameters

Option	Description	Default Value	Topic	Type
Initial Data Load Seed	Determines if data seeding is enabled.	No	Admin	Boolean
Initial Data Load Seed Foundation Data	Determines if data seeding of foundation data is enabled.	No	Admin	Boolean
Initial Data Load Seed Store Data	Determines if data seeding of store data is enabled. Yes: Store Data will be available for data seeding. No: Store Data will not be available for data seeding.	No	Admin	Boolean
Initial Data Load Fail Limit	The maximum number of errors to ignore before processing of a file is terminated.	0	Batch	Integer
Initial Data Load Chunk Log Limit	The maximum number of errors to log when processing a file - this value should be greater than the Initial Data Load Fail Limit.	10	Batch	Integer
Initial Data Load Chunk Limit	The maximum number of records to insert into the DB in a single batch update.	1000	Batch	Integer

Initial Data Loading Process

1. Set the **Initial Data Load Seed** and **Initial Data Load Seed Foundation Data** options to **Yes**.
2. Upload the relevant foundation data files to the **imports** folder in Object Storage via FTS.
3. Run the **Initial Foundation Data File Import** batch job: the batch job will download the foundation data files from Object Storage, parse the files and insert the data into the staging tables, merge/upsert the data from the staging tables into the SIOCS master tables, upload any failed files/records to the **rejects** folder, and any successful files/records to the **archives** folder, in Object Storage.

4. Wait for the batch job to finish then check the Job Execution (Job Admin UI) and Execution Detail (Initial Data Load UI) screens for any errors. If the number of errors exceeds the **Initial Data Load Fail Limit** the entire file will be rejected and uploaded to the **rejects** folder in Object Storage. If the number of errors does not exceed the **Initial Data Load Fail Limit**, the erroneous records will be uploaded to the **rejects** folder, and the successful records to the **archives** folder in Object Storage.

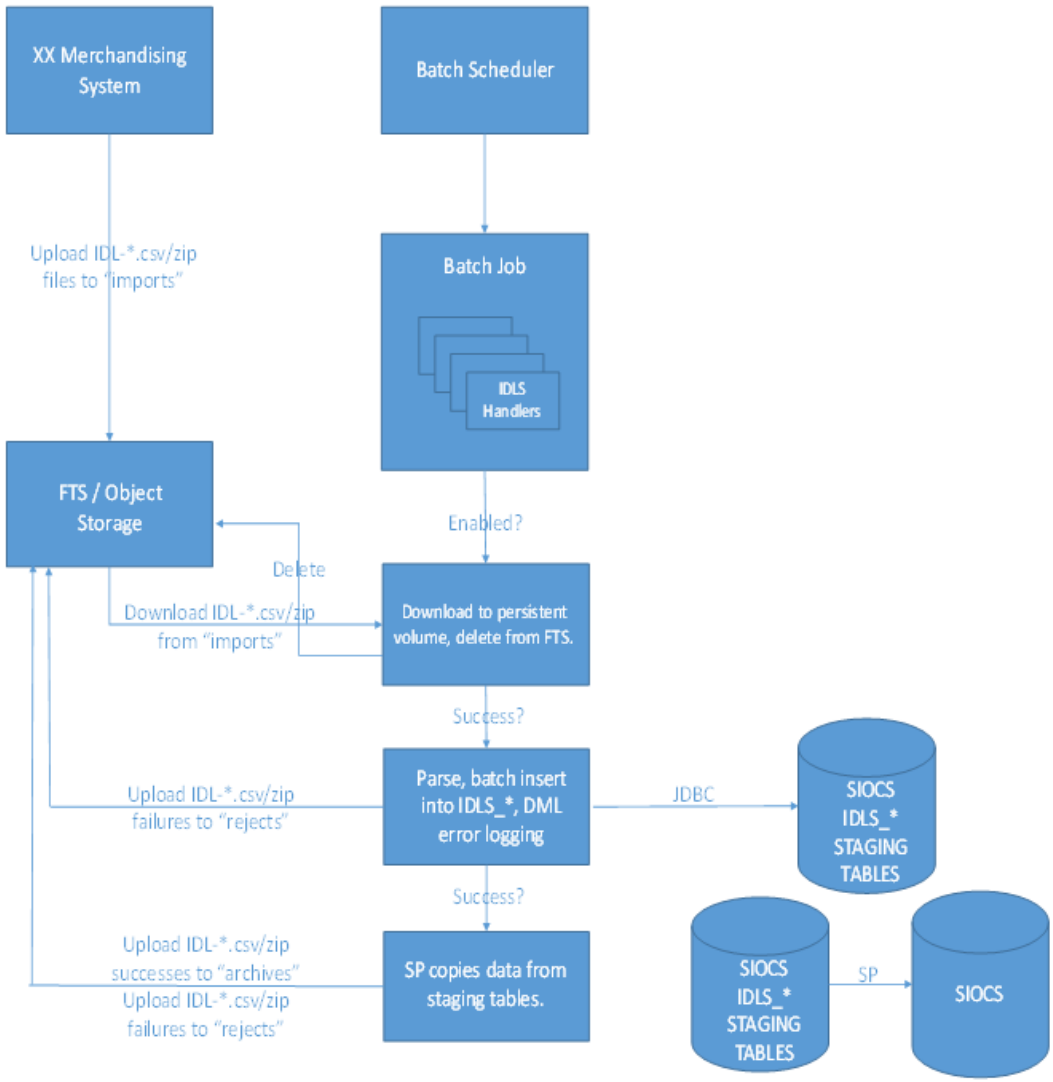
   > **Note:**
   >
   > For performance reasons, the batch job will zip any files > 10 MB before uploading to Object Storage.

5. Correct any errors and repeat steps 2 to 4 until there are no errors and all the foundation data has been imported.

   > **Note:**
   >
   > It is not necessary to delete data for a module before re-importing data for that module: the batch job uses a merge/upsert when copying data from the staging tables to the SIOCS master tables.

6. Set the **Initial Data Load Seed Foundation Data** option to **No** and the **Initial Data Load Seed Store Data** option to **Yes**.
7. Upload the relevant store data files to the **imports** folder in Object Storage via FTS.
8. Run the **Initial Store Data File Import** batch job: the batch job follows the same flow as the **Initial Foundation Data File Import** batch job but for store data.
9. Wait for the batch job to finish then check the Job Execution (Job Admin UI) and Execution Detail (Initial Data Load UI) screens for any errors.
10. Correct any errors and repeat steps 7 to 9 until there are no errors and all the store data has been imported.
11. Set the **Initial Data Load Seed** and **Initial Data Load Seed Store Data** options to **No**.

Figure 2-4 High Level Flow



Data Seeding Modules

Data seeding modules are grouped into 5 data groups:

Table 2-5 Initial Data Loading Groups

Data Group	Module	Description
Item	Item	Item data.
	Item CFA	Item custom flexible attribute data.
	Item Component	Pack item component data.
	Item Description	Item description data.
	Item Hierarchy	Item merchandise hierarchy data, for example, department, class and subclasses.
	Item Image	Item image URL data.
	Item UDA	Item user defined attribute data.

Table 2-5 (Cont.) Initial Data Loading Groups

Data Group	Module	Description
	Related Item	Related item detail data.
	Related Item Type	Related item type data.
Miscellaneous	Differentiator	Item differentiation data.
	Differentiator Type	Item differentiation type data, such as style, color, size, and so on.
	Transfer Zone	Transfer zone data.
	UDA	User defined attribute data.
	UDA LOV	User defined attribute list of values data.
	UOM Class	Unit of measure class data.
	UOM Conversion	Unit of measure conversion data.
Store	Store	Store data.
	Store Address	Store address data.
	Store Item	Store item data.
	Store Item CFA	Store item custom flexible attribute data.
	Store Item Price	Store item price data.
	Store Item Price History	Store item price history data.
	Store Item Stock	Store item stock record data.
	Store UIN Admin Item	Store UIN (Unique Identification Number) admin item data.
Supplier	Partner	Partner data.
	Partner Address	Partner address data.
	Partner Item	Partner item data.
	Supplier	Supplier data.
	Supplier Address	Supplier address data.
	Supplier CFA	Supplier custom flexible attribute data.
	Supplier Item	Supplier item data.
	Supplier Item Country	Supplier item country data.
	Supplier Item Country Dimension	Supplier item country dimension data.
	Supplier Item Manufacturer	Supplier item country manufacture data.
	Supplier Organization	Supplier organization unit data.
	Supplier UOM	Supplier UOM data.
Warehouse	Warehouse	Warehouse data.
	Warehouse Address	Warehouse address data.
	Warehouse Item	Warehouse item data.

The data for the Miscellaneous, Item, Supplier and Warehouse data group modules are imported by the Initial Foundation Data File Import batch job. The data for the Store data group modules are imported by the Initial Store Data File Import batch job. All foundation data should be imported prior to importing any store data. Due to referential integrity constraints (see File Layouts) the batch jobs process the data in the order shown above (for example, the Supplier Item module cannot be imported prior to the Supplier and Item modules, and the Item module

cannot be imported prior to the Item Hierarchy module); as such the files for each module should be uploaded to Object Storage and imported in a similar order (or all at the same time). To import data for a group of Stores, upload all the data for those Stores to Object Storage, then run the Initial Store Data File Import batch job. To import data for a single Store, upload all the data for that Store to Object Storage, then run the Initial Store Data File Import batch job. Alternatively, the Initial Store Data File Import batch job can be run for a single Store by entering the Store ID in the Job Admin UI and adding the Store ID to the corresponding filename(s) (vide infra).

File Layouts

All files should be in CSV (comma-separated values) format, with either a ".csv" or ".dat" filename extension. The batch jobs also support zipped files which will be extracted upon download and processed individually. Empty or blank fields within a record will be considered null. String fields containing a comma or double quote must be quoted (with double quotes), a double quote in a field must be represented by 2 double quote characters. Line breaks within quoted fields are not supported. The filename format is IDL-[MODULENAME]-XXXX.csv(/dat/zip). Files contained within .zip files must adhere to the same filename format. To run the Initial Store Data File Import batch job for a particular Store, the filename format is IDL-[MODULENAME]-[STOREID]-XXXX.csv(/dat/zip). Any files which do not adhere to the filename format will not be downloaded or processed. Files > 500MB will be rejected: the file should be split into smaller files and uploaded as a .zip file. It is recommended to not edit the .csv files in Excel as this can lead to formatting issues.

The file layout for each module is described below:

Differentiator File

Table 2-6 Differentiator File Layout

Field Name	Description	Required	Type
ID	The unique identifier of the differentiator.	Yes	VARCHAR2 (10)
DESCRIPTION	The description of the differentiator.	Yes	VARCHAR2 (255)
DIFF_TYPE_ID	The unique identifier of the differentiator type - this references the ID column in the DIFFERENTIATOR_TYPE table.	No	VARCHAR2 (10)

Example CSV File

IDL-DIFFERENTIATOR-*.csv

1,DESCRIPTION FOR DIFFERENTIATOR 1,1

Differentiator Type File

Table 2-7 Differentiator Type File Layout

Field Name	Description	Required	Type
ID	The unique identifier of the differentiator type.	Yes	VARCHAR2 (10)
DESCRIPTION	The description of the differentiator type.	Yes	VARCHAR2 (255)

Example CSV File

IDL-DIFFERENTIATORTYPE-*.csv

1,DESCRIPTION FOR DIFFERENTIATOR TYPE 1

Item CFA File

Table 2-8 Item CFA File Layout

Field Name	Description	Required	Type
ITEM_ID	The unique identifier of the item - this references the ITEM_ID column in the ITEM table.	Yes	VARCHAR2 (25)
NAME	The name of the custom flex attribute - forms the primary key together with the ITEM_ID field.	Yes	VARCHAR2 (30)
VALUE	The value of the custom flex attribute.	No	VARCHAR2 (250)
VALUE_DATE	The date value of the custom flex attribute in "yyyy-MM-dd" format.	No	DATE

Example CSV File

IDL-ITEMCFA-*.csv

2,Name 2,Value 2,

3,Name 3,,2021-10-06

Item Component File

Table 2-9 Item Component File Layout

Field Name	Description	Required	Type
ITEM_ID	The unique identifier of the pack item - references the ITEM_ID column in the ITEM table.	Yes	VARCHAR2 (25)
COMPONENT_ITEM_ID	The unique identifier of the component item - references the ITEM_ID column in the ITEM table and forms the primary key together with the ITEM_ID field.	Yes	VARCHAR2 (25)
QUANTITY	The quantity of the component item in the pack item.	Yes	NUMBER (12, 4)

Example CSV File

IDL-ITEMCOMPONENT-*.csv

1,11,1.11

Item Description File

Table 2-10 Item Description File Layout

Field Name	Description	Required	Type
ITEM_ID	The unique identifier of the pack item - references the ITEM_ID column in the ITEM table.	Yes	VARCHAR2 (25)
LOCALE_ID	The unique identifier of the locale - references the ID column in the TRANSLATION_LOCALE table and forms the primary key together with the ITEM_ID field - see Supported Locales.	Yes	NUMBER (12, 0)
DESCRIPTION	The description of the item.	Yes	VARCHAR2 (255)
SHORT_DESCRIPTION	The short description of the item.	Yes	VARCHAR2 (250)
SECONDARY_DESCRIPTION	The secondary description of the item.	No	VARCHAR2 (250)
LOCALE lANGUAGE	The ISO 3166 language code - references the LANUGAGE column in the TRANSLATION_LOCALE table - see Supported Locales.	Yes	VARCHAR2 (6)
LOCALE_DESCRIPTION	The description of the locale.	No	VARCHAR2 (120)

Example CSV File

IDL-ITEMDESCRIPTION-*.csv

1,1,Description 1,Short Description 1,Secondary Description 1,en,English

Item File

Table 2-11 Item File Layout

Field Name	Description	Required	Type
ITEM_ID	The unique identifier of the item	Yes	VARCHAR2 (25)
ITEM_TYPE	The type of item - 0 (Item), 15 (Simple Pack), 20 (Complex Pack), 25 (Simple Breakable Pack) or 30 (Complex Breakable Pack).	Yes	NUMBER (2, 0)
DEPARTMENT_ID	The department identifier - references the DEPARTMENT_ID column in the ITEM_HIERARCHY table.	No	NUMBER (12, 0)
CLASS_ID	The class identifier - references the CLASS_ID column in the ITEM_HIERARCHY table.	No	NUMBER (12, 0)
SUBCLASS_ID	The subclass identifier - references the SUBCLASS_ID column in the ITEM_HIERARCHY table.	No	NUMBER (12, 0)
SHORT_DESCRIPTION	The short description of the item.	No	VARCHAR2 (255)

Table 2-11 (Cont.) Item File Layout

Field Name	Description	Required	Type
LONG_DESCRIPTION	The long description of the item.	No	VARCHAR2 (400)
DIFFERENTIATOR_1	The identifier of the first differentiator of the item.	No	VARCHAR2 (10)
DIFFERENTIATOR_2	The identifier of the second differentiator of the item.	No	VARCHAR2 (10)
DIFFERENTIATOR_3	The identifier of the third differentiator of the item.	No	VARCHAR2 (10)
DIFFERENTIATOR_4	The identifier of the fourth differentiator of the item.	No	VARCHAR2 (10)
STATUS	The status of the item - ' ' (None), A (Active), C (Discontinued), I (Inactive), D (Deleted), Q (Auto-stocked) or N (Non-ranged).	No	VARCHAR2 (1)
ORDER_AS_TYPE	Indicates if a pack item is receivable at the component level or at the pack level (for a buyer pack only).	No	VARCHAR2 (1)
PARENT_ITEM_ID	The unique identifier of the parent item.	No	VARCHAR2 (25)
TRANSACTION_LEVEL	Number indicating which of the three levels transactions occur for the item's group.	No	NUMBER
ITEM_LEVEL	Number indicating which of the three levels the item resides.	No	NUMBER
SELLABLE	Flag indicating if the item may be sold as a unit - Y or N.	Yes	VARCHAR2 (1)
ORDERABLE	Flag indicating if the item may be ordered from a supplier - Y or N.	Yes	VARCHAR2 (1)
PACKAGE_UNIT_OF_MEASURE	The unit of measure associated with the package size.	No	VARCHAR2 (4)
PACKAGE_SIZE	The size of the product printed on any packaging.	No	NUMBER (12, 4)
UNIT_OF_MEASURE	The unit of measure.	Yes	VARCHAR2 (4)
CASE_SIZE	The default number of items that are contained in a case.	No	NUMBER (12, 4)
BARCODE_FORMAT	The barcode format for the item.	No	VARCHAR2 (4)
BARCODE_PREFIX	The barcode prefix for the item.	No	NUMBER (9, 0)
TICKET_TYPE_CODE	The ticket type code for the item.	No	VARCHAR2 (6)
EACH_TO_UOM_FACTOR	The conversion factor between an "Each" and the standard unit of measure.	No	NUMBER (20, 10)
WASTE_TYPE	Identifies the wastage type as either sales or spoilage wastage - SL (sales) or SP (spoilage).	No	VARCHAR2 (6)
WASTE_PERCENT	Average percent of wastage for the item over its shelf life.	No	NUMBER (12, 4)
WASTE_PERCENT_DEFAULT	Default daily wastage percent for spoilage type wastage items.	No	NUMBER (12, 4)
ESTIMATE_SOH_FOR_PACK	Indicates if a notional simple pack item's inventory should be displayed in packs - Y or N.	Yes	VARCHAR2 (1)

Table 2-11 (Cont.) Item File Layout

Field Name	Description	Required	Type
RETAIL_ZONE_ID	The unique identifier of the retail pricing strategy associated with the item.	No	VARCHAR2 (128)
IS_PRIMARY	Flag indicating if the sub-transaction level item is designated as the primary sub-transaction level item - Y or N.	No	VARCHAR2 (1)
BRAND	The brand associated with the item.	No	VARCHAR2 (30)
MANU_SUGGESTED_RETAIL_PRICE	The manufacturer's recommended retail price for the item.	No	NUMBER (12, 4)
MANU_SUGGESTED_RETAIL_CURRENCY	The ISO 4217 currency code of the manufacturer's retail price.	No	VARCHAR2 (3)
INVENTORIABLE	Flag indicating if the item is inventoriable - Y or N.	Yes	VARCHAR2 (1)
SHIP_ALONE	Flag indicating if the item should be shipped to the customer as a separate package - Y or N.	No	VARCHAR2 (1)
BRAND_DESCRIPTION	The description of the brand associated with the item.	No	VARCHAR2 (120)

Example CSV File

IDL-ITEM-*.csv

1,0,1,1,1,SHORT_DESC,LONG_DESC,1,2,3,4,A,N,4,1,3,Y,N,kg,12345678.1234,kg,1,UPCA,22,TT,1,SL,33.33,16.66,Y,RETAIL_ZONE_ID,N,BRAND,4.99,GBP,Y,N,BRAND_DESC

Item Hierarchy File

Table 2-12 Item Hierarchy File Layout

Field Name	Description	Required	Type
DEPARTMENT_ID	The department identifier.	No	NUMBER (12, 0)
DEPARTMENT_NAME	The name of the department.	No	VARCHAR2 (360)
CLASS_ID	The class identifier.	No	NUMBER (12, 0)
CLASS_NAME	The name of the class.	No	VARCHAR2 (360)
SUBCLASS_ID	The subclass identifier.	No	NUMBER (12, 0)
SUBCLASS_NAME	The name of the subclass.	No	VARCHAR2 (360)
STATUS	The status of the item hierarchy - A (Active) or D (Deleted).	Yes	VARCHAR2 (1)

The unique key comprises the DEPARTMENT_ID, CLASS_ID and SUBCLASS_ID fields.

Example CSV File

IDL-ITEMHIERARCHY-*.csv

1,Department 1,1,Class 1,1,Subclass 1,A

Item Image File

Table 2-13 Item Image File Layout

Field Name	Description	Required	Type
ITEM_ID	The unique identifier of the pack item - references the ITEM_ID column in the ITEM table.	Yes	VARCHAR2 (25)
DISPLAY_SEQUENCE	The display sequence order of images associated to the item.	Yes	NUMBER (2, 0)
IMAGE_URL	The URL of the item image.	Yes	VARCHAR2 (1000)
IMAGE_NAME	The name of the item image - forms a unique key together with the ITEM_ID field.	Yes	VARCHAR2 (120)
IMAGE_SIZE_CODE	The type of item image. Valid values are defined as members of IITD code type - T (Thumbnail), H (High), M (Medium) or L (Low).	Yes	VARCHAR2 (6)

Example CSV File

IDL-ITEMIMAGE-*.csv

1,99,http://somewhere.com/someimage1.gif,Image1.gif,T

Item UDA File

Table 2-14 Item UDA File Layout

Field Name	Description	Required	Type
ITEM_ID	The unique identifier of the item - references the ITEM_ID column in the ITEM table.	Yes	VARCHAR2 (25)
UDA_ID	The unique identifier of the user defined attribute - references the ID column in the UDA table and forms a unique key along with the ITEM_ID field.	Yes	NUMBER (5, 0)
UDA_DATE	The value, in 'yyyy-MM-dd HH:mm:ss' format, for DT (Date) user defined attributes.	No	DATE
UDA_TEXT	The value for FF (Text) user defined attributes.	No	VARCHAR2 (250)
UDA_VALUE	The value for LOV (List of Values) user defined attributes.	No	VARCHAR2 (25)

Example CSV File

IDL-ITEMUDA-*.csv

1,1,2021-10-01 12:34:56,FF1,LOV1

Partner Address File

Table 2-15 Partner Address File Layout

Field Name	Description	Required	Type
EXTERNAL_ID	The external identifier of the address.	Yes	VARCHAR2 (25)
PARTNER_ID	The unique identifier of the partner - forms a unique key together with the EXTERNAL_ID field.	Yes	NUMBER (10, 0)
ADDRESS_TYPE	The type of address - 01 (Business), 02 (Postal), 03 (Returns), 04 (Order), 05 (Invoice), 06 (Remittance), 07 (Billing), 08 (Delivery) or 09 (External).	Yes	VARCHAR2 (2)
IS_PRIMARY	Flag indicating if this is the primary address - Y or N.	Yes	VARCHAR2 (1)
ADDRESS_LINE_1	The first line of the address.	No	VARCHAR2 (240)
ADDRESS_LINE_2	The second line of the address.	No	VARCHAR2 (240)
ADDRESS_LINE_3	The third line of the address.	No	VARCHAR2 (240)
CITY	The city.	No	VARCHAR2 (120)
STATE	The state.	No	VARCHAR2 (3)
COUNTRY_ID	The ISO 3166 2- (or 3-) letter country code.	No	VARCHAR2 (3)
POSTAL_CODE	The postal code.	No	VARCHAR2 (30)
CONTACT_NAME	The contact name.	No	VARCHAR2 (120)
CONTACT_PHONE	The contact phone number.	No	VARCHAR2 (20)
CONTACT_FAX	The contact fax number.	No	VARCHAR2 (20)
CONTACT_EMAIL	The contact email address.	No	VARCHAR2 (100)
COUNTY	The county.	No	VARCHAR2 (250)

Example CSV File

IDL-PARTNERADDR-*.csv

1,1,01,Y,Line 1,Line 2,Line 3,City,MN,USA,Postcode,Contact Name,Contact_Phone,Contact_Fax,Contact_Email,County

Partner File

Table 2-16 Partner File Layout

Field Name	Description	Required	Type
ID	The unique identifier of the partner.	Yes	NUMBER (10, 0)
NAME	The name of the partner.	No	VARCHAR2 (240)
CURRENCY_CODE	The ISO 4217 currency code of the partner.	No	VARCHAR2 (3)

Table 2-16 (Cont.) Partner File Layout

Field Name	Description	Required	Type
LOCALE_ID	The locale identifier of the partner - see Supported Locales.	No	NUMBER (6)
STATUS	The status of the partner - A (Active) or I (Inactive).	No	VARCHAR2 (1)
CONTACT_NAME	The contact name.	No	VARCHAR2 (120)
CONTACT_PHONE	The contact phone number.	No	VARCHAR2 (20)
CONTACT_FAX	The contact fax number.	No	VARCHAR2 (20)
CONTACT_TELEX	The contact telex number.	No	VARCHAR2 (20)
CONTACT_EMAIL	The contact email address.	No	VARCHAR2 (100)
MANUFACTURER_ID	The manufacturer's tax identification number.	No	VARCHAR2 (18)
PRINCIPAL_COUNTRY_ID	The ISO 3166 2- (or 3-) letter country code to which the partner is assigned.	No	VARCHAR2 (3)
TAX_ID	The unique tax identification number of the partner.	No	VARCHAR2 (18)
PAYMENT_TERMS	The payment terms of the partner.	No	VARCHAR2 (20)
IMPORT_COUNTRY_ID	The ISO 3166 2- (or 3-) letter country code of the Import Authority.	No	VARCHAR2 (3)
IMPORT_PRIMARY	Flag that indicates if an Import Authority is the primary Import Authority for an import country - Y or N.	No	VARCHAR2 (1)
ORGANIZATION_UNIT_ID	The organization unit identifier of the partner.	No	VARCHAR2 (15)
VALUE_ADDED_TAX_REGION	The VAT region of the partner.	No	VARCHAR2 (20)
TRANSFER_ENTITY_ID	The transfer entity identifier of the partner.	No	VARCHAR2 (20)

Example CSV File

IDL-PARTNER-*.csv

1,Partner 1,GBP,1,A,Contact Name 1,Contact Phone 1,Contact Fax 1,Contact Telex 1,Contact Email 1,Manufacturer ID 1,GB,123456789012345678,Payment Terms 1,US,Y,Org Unit ID 1,VAT Region 1,Transfer Entity ID 1

Partner Item File

Table 2-17 Partner Item File Layout

Field Name	Description	Required	Type
ITEM_ID	The unique identifier of the item - references the ITEM_ID column in the ITEM table.	Yes	VARCHAR2 (25)

Table 2-17 (Cont.) Partner Item File Layout

Field Name	Description	Required	Type
PARTNER_ID	The unique identifier of the partner - this references the ID column in the PARTNER table and forms the primary key together with the ITEM_ID field.	Yes	NUMBER (10, 0)
STATUS	The status of the item - ' ' (None), A (Active), C (Discontinued), I (Inactive), D (Deleted), Q (Auto-stocked) or N (Non-ranged).	No	VARCHAR2 (2)

Example CSV File

IDL-PARTNERITEM-*.csv

1,1,A

Related Item File

Table 2-18 Related Item File Layout

Field Name	Description	Required	Type
ITEM_ID	The unique identifier of the item - references the ITEM_ID column in the ITEM table.	Yes	VARCHAR(25)
RELATIONSHIP_ID_EXTERNAL	The external identifier of the relationship type.	Yes	NUMBER (20, 0)
RELATIONSHIP_NAME	The name of the relationship type.	No	VARCHAR2 (120)
RELATIONSHIP_TYPE	The relationship type - RLTD (Related), SUBS (Substitute), UPSL (Upsell) or CSSL (Crosssell).	Yes	VARCHAR2 (6)
MANDATORY_IND	Flag indicating if the relationship is mandatory - Y or N.	Yes	VARCHAR2 (1)
RELATED_ITEM_ID	The unique identifier of the related item - references the ITEM_ID column in the ITEM table.	Yes	VARCHAR2 (25)
PRIORITY_NUMBER	The priority when there are multiple relationships.	No	NUMBER (4, 0)
EFFECTIVE_DATE	The effective date of the relationship in "yyyy-MM-dd HH:mm:ss" format.	No	DATE
END_DATE	The end date of the relationship in "yyyy-MM-dd HH:mm:ss" format.	No	DATE

The primary key comprises the ITEM_ID, RELATIONSHIP_ID_EXTERNAL and RELATED_ITEM_ID fields.

Example CSV File

IDL-RELATEDITEM-*.csv

1,1,Related,RLTD,N,11,9999,2021-10-01 12:34:56,2021-11-01 12:34:56

Related Item Type File

Table 2-19 Related Item Type File Layout

Field Name	Description	Required	Type
ITEM_ID	The unique identifier of the item - references the ITEM_ID column in the ITEM table.	Yes	VARCHAR2 (25)
RELATIONSHIP_ID_EXTERNAL	The external identifier of the relationship type - forms the primary key together with the ITEM_ID field.	Yes	NUMBER (20, 0)
RELATIONSHIP_NAME	The name of the relationship type.	No	VARCHAR2 (120)
RELATIONSHIP_TYPE	The relationship type - RLTD (Related), SUBS (Substitute), UPSL (Upsell) or CSSL (Crosssell).	Yes	VARCHAR2 (6)
MANDATORY_IND	Flag indicating if the relationship is mandatory - Y or N.	Yes	VARCHAR2 (1)

Example CSV File

IDL-RELATEDITEMTYPE-*.csv

1,1,Related,RLTD,N

Store Address File

Table 2-20 Store Address File Layout

Field Name	Description	Required	Type
EXTERNAL_ID	The external identifier of the address.	Yes	VARCHAR2 (25)
STORE_ID	The unique identifier of the store - forms a unique key together with the EXTERNAL_ID field.	Yes	NUMBER (10, 0)
ADDRESS_TYPE	The type of address - 01 (Business), 02 (Postal), 03 (Returns), 04 (Order), 05 (Invoice), 06 (Remittance), 07 (Billing), 08 (Delivery) or 09 (External).	Yes	VARCHAR2 (2)
IS_PRIMARY	Flag indicating if this is the primary address - Y or N.	Yes	VARCHAR2 (1)
ADDRESS_LINE_1	The first line of the address.	No	VARCHAR2 (240)
ADDRESS_LINE_2	The second line of the address.	No	VARCHAR2 (240)
ADDRESS_LINE_3	The third line of the address.	No	VARCHAR2 (240)
CITY	The city.	No	VARCHAR2 (120)
STATE	The state.	No	VARCHAR2 (3)
COUNTRY_ID	The ISO 3166 2- (or 3-) letter country code.	No	VARCHAR2 (3)
POSTAL_CODE	The postal code.	No	VARCHAR2 (30)

Table 2-20 (Cont.) Store Address File Layout

Field Name	Description	Required	Type
CONTACT_NAME	The contact name.	No	VARCHAR2 (120)
CONTACT_PHONE	The contact phone number.	No	VARCHAR2 (20)
CONTACT_FAX	The contact fax number.	No	VARCHAR2 (20)
CONTACT_EMAIL	The contact email address.	No	VARCHAR2 (100)
COUNTY	The county.	No	VARCHAR2 (250)

Example CSV File

IDL-STOREADDR-*.csv

1,1,01,Y,Line 1,Line 2,Line 3,City,MN,USA,Postcode,Contact Name,Contact_Phone,Contact_Fax,Contact_Email,County

Store File

Table 2-21 Store File Layout

Field Name	Description	Required	Type
ID	The unique identifier of the store.	Yes	NUMBER (10,0)
NAME	The name of the store.	Yes	VARCHAR2 (150)
ORGANIZATION_UNIT_ID	The organization unit identifier of the store.	No	VARCHAR2 (15)
LOCALE_LANGUAGE	The ISO 3166 language to which the store is assigned - see Supported Locales.	No	VARCHAR2 (3)
LOCALE_COUNTRY	The ISO 3166 2- (or 3-) letter country code to which the store is assigned.	No	VARCHAR2 (3)
OPEN_DATE	The date on which the store opened in 'yyyy-MM-dd' format.	No	DATE
CLOSE_DATE	The date on which the store closed in 'yyyy-MM-dd' format.	No	DATE
TOTAL_SQUARE_FEET	The total square footage of the store.	No	NUMBER (9,2)
SELLING_SQUARE_FEET	The total square footage of the store's selling area.	No	NUMBER (9,2)
CURRENCY_CODE	The ISO 4217 currency code of the store.	No	VARCHAR2 (40)
TRANSFER_ZONE_ID	The transfer zone identifier.	No	VARCHAR2 (128)
SIM_STORE	Flag indicating if the store is using the SIM application - Y or N.	No	VARCHAR2 (1)
TIMEZONE	The time zone of the store.	Yes	VARCHAR2 (80)
CUSTOMER_ORDER_LOC_IND	Flag indicating if the store is a customer order location - Y or N.	Yes	VARCHAR2 (1)

Example CSV File

IDL-STORE-*.csv

1,Store 1,Org Unit ID,en,GB,2001-01-01,2030-12-31,20,10,GBP,1,Y,GMT,Y

Store Item CFA File

Table 2-22 Store Item CFA File Layout

Field Name	Description	Required	Type
ITEM_ID	The unique identifier of the item - this references the ITEM_ID column in the ITEM table.	Yes	VARCHAR2 (25)
STORE_ID	The unique identifier of the store - this references the ID column in the STORE table.	Yes	NUMBER (10, 0)
NAME	The name of the custom flex attribute.	Yes	VARCHAR2 (30)
VALUE	The value of the custom flex attribute.	No	VARCHAR2 (250)
VALUE_DATE	The date value of the custom flex attribute in "yyyy-MM-dd" format.	No	DATE

The primary key comprises the ITEM_ID, STORE_ID and NAME fields.

Example CSV File

IDL-STOREITEMCFA-*.csv

2,1,Name 2,Value 2,

3,1,Name 3, ,2021-10-06

Store Item File

Table 2-23 Store Item File Layout

Field Name	Description	Required	Type
ITEM_ID	The unique identifier of the item - references the ITEM_ID column in the ITEM table.	Yes	VARCHAR2 (25)
STORE_ID	The unique identifier of the store - this references the ID column in the STORE table and forms the primary key together with the ITEM_ID field.	Yes	NUMBER (10, 0)
ITEM_TYPE	The type of store item - 0 (Item), 15 (Simple Pack), 20 (Complex Pack), 25 (Simple Breakable Pack) or 30 (Complex Breakable Pack).	Yes	VARCHAR2 (255)
SHORT_DESCRIPTION	The short description of the store item.	No	VARCHAR2 (255)
LONG_DESCRIPTION	The long description of the store item.	No	VARCHAR2 (400)

Table 2-23 (Cont.) Store Item File Layout

Field Name	Description	Required	Type
STATUS	The status of the store item - ' ' (None), A (Active), C (Discontinued), I (Inactive), D (Deleted), Q (Auto-stocked) or N (Non-ranged).	No	VARCHAR2 (20)
STATUS_DATE	The date that the status of the store item was updated in 'yyyy-MM-dd' format.	No	DATE
DEFAULT_CURRENCY	The default ISO 4217 currency code of the store item.	Yes	VARCHAR2 (3)
PRIMARY_SUPPLIER_ID	The identifier of the primary supplier of the store item - this references the ID column in the SUPPLIER table.	No	NUMBER (10, 0
NEXT_DELIVERY_DATE	The next delivery date of the store item in 'yyyy-MM-dd' format.	No	DATE
UIN_REQUIRED	Flag to indicate if a UIN (unique identification number) is required for the store item - Y or N.	No	VARCHAR2 (1)
REPLENISHMENT_TYPE	The replenishment method for the store item - SO (Store Order).	No	VARCHAR2 (6)
REJECT_STORE_ORDER	Flag indicating if uploaded store orders should be rejected for the store item - Y or N.	No	VARCHAR2 (1)
STORE_CONTROL_PRICING	Flag indicating if the store can modify the item's price - Y or N.	No	VARCHAR2 (1)
MULTIPLE_DELIVERY_PER_DAY	Flag indicating if the store item is replenished multiple times per day - Y or N.	No	VARCHAR2 (1)
RFID	Flag indicating if the store item is RFID tagged - Y or N.	Yes	VARCHAR2 (1)
CONSIGNMENT_TYPE	The consignment type of the store item - 5 (Consignment) or 10 (Concession).	No	NUMBER (2, 0)

Example CSV File

IDL-STOREITEM-*.csv

1,1,0,Short Desc 1,Long Desc 1,A,2022-01-14,GBP,1,2022-01-31,N,SO,N,Y,Y,N,10

Store Item Price File

Table 2-24 Store Item Price File Layout

Field Name	Description	Required	Type
ITEM_ID	The unique identifier of the item - this references the ITEM_ID column in the STORE_ITEM table.	Yes	VARCHAR2 (25)
STORE_ID	The unique identifier of the store - this references the STORE_ID column in the STORE_ITEM table.	Yes	NUMBER (10, 0)
EFFECTIVE_DATE	The date that the item price becomes effective in 'yyyy-MM-dd HH:mm:ss' format.	No	DATE

Table 2-24 (Cont.) Store Item Price File Layout

Field Name	Description	Required	Type
END_DATE	The date that the item price is no longer valid in 'yyyy-MM-dd HH:mm:ss' format.	No	DATE
PRICE_TYPE	The item price type - 202 (Permanent/ Regular), 201 (Promotional) or 200 (Clearance).	Yes	NUMBER (3, 0)
STORE_REQUESTED	Flag indicating if the item price was requested by the store - Y or N.	Yes	VARCHAR2 (1)
STATUS	The status of the item price - 0 (New), 1 (Pending), 2 (Approved), 3 (Completed), 4 (Rejected), 5 (Ticket List), 6 (Active), 7 (Extract Failed), 9 (Deleted) or 99 (Default).	Yes	NUMBER (2, 0)
PROMOTION_ID	The identifier of the promotion.	No	NUMBER (10, 0)
PROMOTION_COMP_ID	The identifier of the promotion component.	No	NUMBER (10, 0)
MULTI_UNITS	The number of units involved in the multi-unit pricing of the item price.	No	NUMBER (12, 4)
MULTI_UNIT_RETAIL_CURRENCY	The ISO 4217 currency code of the multi-unit price.	No	VARCHAR2 (3)
MULTI_UNIT_RETAIL	The value of the multi-unit price.	No	NUMBER (20, 4)
MULTI_UNIT_UOM	The unit of measure of the multi-unit price.	No	VARCHAR2 (4)
MULTI_UNIT_CHANGE	Flag indicating if the multi-unit price has changed - Y or N.	Yes	VARCHAR2 (1)
SELLING_UNIT_CHANGE	Flag indicating if the item price has changed - Y or N.	Yes	VARCHAR2 (1)
PROMOTION_NAME	The name of the promotion.	No	VARCHAR2 (160)
PROMOTION_DESCRIPTION	The description of the promotion.	No	VARCHAR2 (640)
PROMOTION_COMP_NAME	The name of the promotion component.	No	VARCHAR2 (160)
RESET_CLEARANCE_ID	The clearance reset identifier.	No	NUMBER (15, 0)
PROMO_COMP_TYPE	The promotion component type - 0 (Complex), 1 (Simple), 2 (Threshold), 3 (Credit) or 4 (Threshold).	No	NUMBER (2, 0)
REGULAR_PRICE_CHANGE_ID	The identifier of the regular price change.	No	NUMBER (15, 0)
CLEARANCE_ID	The identifier of the clearance price change.	No	NUMBER (15, 0)
PROMO_COMP_DTL_ID	The identifier of the promotion component detail.	No	NUMBER (15, 0)
PROMO_DURATION_TYPE	The promotion duration type - 1 (All Day), 2 (Partial Day) or 3 (Multiple Day).	No	NUMBER (2, 0)
PRICE_VALUE	The value of the item price.	Yes	NUMBER (20, 4)

Table 2-24 (Cont.) Store Item Price File Layout

Field Name	Description	Required	Type
PRICE_CURRENCY	The ISO 4217 currency code of the item price.	No	VARCHAR2 (3)
PRICE_UNIT_OF_MEASURE	The unit of measure of the item price.	No	VARCHAR2 (4)
EXT_PRICE_EVENT_ID	The external price event identifier.	No	NUMBER (12, 0)

For Permanent/Regular (202) Item Prices the unique key comprises the ITEM_ID, STORE_ID, PRICE_TYPE and REGULAR_PRICE_CHANGE_ID fields. For Promotional (201) Item Prices the unique key comprises the ITEM_ID, STORE_ID, PRICE_TYPE, PROMOTION_ID, PROMOTION_COMP_ID and PROMO_COMP_DTL_ID fields. For Clearance (200) Item Prices the unique key comprises the ITEM_ID, STORE_ID, PRICE_TYPE and CLEARANCE_ID fields.

Example CSV File

IDL-STOREITEMPRICE-*.csv

1,1,2021-10-06 12:34:56,2021-10-06 12:34:56,202,N,6,,,1,GBP,2469,kg,Y,N,,,,,,1,,,,1234.5678,GBP,kg,1111

1,1,2021-10-07 12:34:56,2021-10-07 12:34:56,201,N,6,1,1,1,GBP,2469,g,Y,N,Promo Name,Promo Desc,Promo Comp Name,,1,,,1,3,1234.5678,GBP,g,3333

1,1,2021-10-08 12:34:56,2021-10-08 12:34:56,200,N,6,,,1,GBP,2469,lb,Y,N,,,,1,,,1,,,1234.5678,GBP,lb,5555

Store Item Price History File

Table 2-25 Store Item Price History File Layout

Field Name	Description	Required	Type
ITEM_PRICE_ID	The identifier of the item price.	No	NUMBER (12, 0)
ITEM_ID	The unique identifier of the item.	Yes	VARCHAR2 (25)
STORE_ID	The unique identifier of the store.	Yes	NUMBER (10, 0)
EFFECTIVE_DATE	The date that the item price becomes effective in 'yyyy-MM-dd HH:mm:ss' format.	No	DATE
END_DATE	The date that the item price is no longer valid in 'yyyy-MM-dd HH:mm:ss' format.	No	DATE
PRICE_TYPE	The item price type - 202 (Permanent/ Regular), 201 (Promotional) or 200 (Clearance).	Yes	NUMBER (3, 0)
STORE_REQUESTED	Flag indicating if the item price was requested by the store - Y or N.	Yes	VARCHAR2 (1)
PROMOTION_ID	The identifier of the promotion.	No	NUMBER (10, 0)
PROMOTION_COMP_ID	The identifier of the promotion component.	No	NUMBER (10, 0)

Table 2-25 (Cont.) Store Item Price History File Layout

Field Name	Description	Required	Type
MULTI_UNITS	The number of units involved in the multi-unit pricing of the item price.	No	NUMBER (12, 4)
MULTI_UNIT_RETAIL_CURRENCY	The ISO 4217 currency code of the multi-unit price.	No	VARCHAR2 (3)
MULTI_UNIT_RETAIL	The value of the multi-unit price.	No	NUMBER (20, 4)
MULTI_UNIT_UOM	The unit of measure of the multi-unit price.	No	VARCHAR2 (4)
MULTI_UNIT_CHANGE	Flag indicating if the multi-unit price has changed - Y or N.	Yes	VARCHAR2 (1)
SELLING_UNIT_CHANGE	Flag indicating if the item price has changed - Y or N.	Yes	VARCHAR2 (1)
PROMOTION_NAME	The name of the promotion.	No	VARCHAR2 (160)
PROMOTION_DESCRIPTION	The description of the promotion.	No	VARCHAR2 (640)
PROMOTION_COMP_NAME	The name of the promotion component.	No	VARCHAR2 (160)
RESET_CLEARANCE_ID	The clearance reset identifier.	No	NUMBER (15, 0)
PROMO_COMP_TYPE	The promotion component type - 0 (Complex), 1 (Simple), 2 (Threshold), 3 (Credit) or 4 (Threshold).	No	NUMBER (2, 0)
REGULAR_PRICE_CHANGE_ID	The identifier of the regular price change.	No	NUMBER (15, 0)
CLEARANCE_ID	The identifier of the clearance price change.	No	NUMBER (15, 0)
PROMO_COMP_DTL_ID	The identifier of the promotion component detail.	No	NUMBER (15, 0)
PROMO_DURATION_TYPE	The promotion duration type - 1 (All Day), 2 (Partial Day) or 3 (Multiple Day).	No	NUMBER (2, 0)
PRICE_VALUE	The value of the item price.	Yes	NUMBER (20, 4)
PRICE_CURRENCY	The ISO 4217 currency code of the item price.	No	VARCHAR2 (3)
PRICE_UNIT_OF_MEASURE	The unit of measure of the item price.	No	VARCHAR2 (4)

For Permanent/Regular (202) Item Prices the unique key comprises the ITEM_ID, STORE_ID, PRICE_TYPE and REGULAR_PRICE_CHANGE_ID fields. For Promotional (201) Item Prices the unique key comprises the ITEM_ID, STORE_ID, PRICE_TYPE, PROMOTION_ID, PROMOTION_COMP_ID and PROMO_COMP_DTL_ID fields. For Clearance (200) Item Prices the unique key comprises the ITEM_ID, STORE_ID, PRICE_TYPE and CLEARANCE_ID fields.

Example CSV File

IDL-STOREITEMPRICEHIST-*.csv

1,1,1,2021-10-06 12:34:56,2021-10-06 12:34:56,202,N,,,1,GBP,2469,kg,Y,N,,,,,1,,,,1234.5678,GBP,kg

3,1,1,2021-10-07 12:34:56,2021-10-07 12:34:56,201,N,1,1,1,GBP,2469,g,Y,N,Promo Name,Promo Desc,Promo Comp Name,,1,,,1,3,1234.5678,GBP,g

5,1,1,2021-10-08 12:34:56,2021-10-08 12:34:56,200,N,,,1,GBP,2469,lb,Y,N,,,,1,,,1,,,1234.5678,GBP,lb

Store Item Stock File

Table 2-26 Store Item Stock File Layout

Field Name	Description	Required	Type
ITEM_ID	The unique identifier of the item - this references the ITEM_ID column in the STORE_ITEM table.	Yes	VARCHAR2 (25)
STORE_ID	The unique identifier of the store - this references the STORE_ID column in the STORE_ITEM table and forms the primary key together with the ITEM_ID field.	Yes	NUMBER (10, 0)
QUANTITY_TOTAL	The total quantity of the item that is sellable.	Yes	NUMBER (12, 4)
QUANTITY_RESERVED	The reserved quantity of the item.	Yes	NUMBER (12, 4)
QUANTITY_CUSTOMER_RESERVE	The quantity of the item reserved for customers.	Yes	NUMBER (12, 4)
QUANTITY_IN_TRANSIT	The in transit quantity of the item.	Yes	NUMBER (12, 4)
QUANTITY_VENDOR_RETURN	The vendor return quantity of the item.	Yes	NUMBER (12, 4)
QUANTITY_NON_SELLABLE	The non-sellable quantity of the item.	Yes	NUMBER (12, 4)

All records in this file will be used to populate the STORE_ITEM_STOCK table. Records where the QUANTITY_NON_SELLABLE field is non-zero will be used to populate the STORE_ITEM_STOCK_NONSELL table.

Example CSV File

IDL-STOREITEMSTOCK-*.csv

1,1,1.1,1.2,1.3,1.4,1.5,1.6

Store UIN Admin Item File

Table 2-27 Store UIN Admin Item File Layout

Field Name	Description	Required	Type
ITEM_ID	The unique identifier of the item - this references the ITEM_ID column in the ITEM table.	Yes	VARCHAR2 (25)
STORE_ID	The unique identifier of the store - this references the ID column in the STORE table and forms the primary key together with the ITEM_ID field.	Yes	NUMBER (10, 0)

Table 2-27 (Cont.) Store UIN Admin Item File Layout

Field Name	Description	Required	Type
UIN_TYPE	The UIN (Unique Identification Number) type - 1 (Serial Number) or 2 (Auto-generated Serial Number).	Yes	NUMBER (2, 0)
UIN_LABEL_ID	The UIN label identifier - SN (Serial Number), IM (IMEI), LN (License Number), PN (Plate Number) or SIN (SIN).	Yes	VARCHAR2 (3)
CAPTURE_TIME_ID	The time to capture the UIN - 1 (Sales) or 2 (Store Receiving).	No	NUMBER (2, 0)
EXTERNAL_CREATE_ALLOWED	Flag to indicate if the UIN can be created externally - Y or N.	No	VARCHAR2 (1)
TICKET_FORMAT_ID	The ticket format identifier.	No	NUMBER (10, 0)

Example CSV File

IDL-STOREUINADMINITEM-*.csv

1,1,1,SN,1,N,1

Supplier Address File

Table 2-28 Supplier Address File Layout

Field Name	Description	Required	Type
EXTERNAL_ID	The external identifier of the address.	Yes	VARCHAR2 (25)
SUPPLIER_ID	The unique identifier of the supplier - forms a unique key together with the EXTERNAL_ID field.	Yes	NUMBER (10, 0)
ADDRESS_TYPE	The type of address - 01 (Business), 02 (Postal), 03 (Returns), 04 (Order), 05 (Invoice), 06 (Remittance), 07 (Billing), 08 (Delivery) or 09 (External).	Yes	VARCHAR2 (2)
IS_PRIMARY	Flag indicating if this is the primary address - Y or N.	Yes	VARCHAR2 (1)
ADDRESS_LINE_1	The first line of the address.	No	VARCHAR2 (240)
ADDRESS_LINE_2	The second line of the address.	No	VARCHAR2 (240)
ADDRESS_LINE_3	The third line of the address.	No	VARCHAR2 (240)
CITY	The city.	No	VARCHAR2 (120)
STATE	The state.	No	VARCHAR2 (3)
COUNTRY_ID	The ISO 3166 2- (or 3-) letter country code.	No	VARCHAR2 (3)
POSTAL_CODE	The postal code.	No	VARCHAR2 (30)
CONTACT_NAME	The contact name.	No	VARCHAR2 (120)
CONTACT_PHONE	The contact phone number.	No	VARCHAR2 (20)

Table 2-28 (Cont.) Supplier Address File Layout

Field Name	Description	Required	Type
CONTACT_FAX	The contact fax number.	No	VARCHAR2 (20)
CONTACT_EMAIL	The contact email address.	No	VARCHAR2 (100)
COUNTY	The county.	No	VARCHAR2 (250)

Example CSV File

IDL-SUPPLIERADDR-*.csv

1,1,01,Y,Line 1,Line 2,Line 3,City,MN,USA,Postcode,Contact Name,Contact_Phone,Contact_Fax,Contact_Email,County

Supplier CFA File

Table 2-29 Supplier CFA File Layout

Field Name	Description	Required	Type
SUPPLIER_ID	The unique identifier of the supplier - this references the ID column in the SUPPLIER table.	Yes	NUMBER (10, 0)
NAME	The name of the custom flex attribute - forms the primary key together with the SUPPLIER_ID field.	Yes	VARCHAR2 (30)
VALUE	The value of the custom flex attribute.	No	VARCHAR2 (250)
VALUE_DATE	The date value of the custom flex attribute in "yyyy-MM-dd" format.	No	DATE

Example CSV File

IDL-SUPPLIERCFA-*.csv

2,Name 2,Value 2,

3,Name 3,,2021-10-06

Supplier File

Table 2-30 Supplier File Layout

Field Name	Description	Required	Type
ID	The unique identifier of the supplier.	Yes	NUMBER (10, 0)
DUNS_NUMBER	The Dun and Bradstreet number to identify the supplier.	No	VARCHAR2 (9)
NAME	The name of the supplier.	No	VARCHAR2 (240)
STATUS	The status of the supplier - A (Active) or I (Inactive).	No	VARCHAR2 (1)

Table 2-30 (Cont.) Supplier File Layout

Field Name	Description	Required	Type
LOCALE_LANGUAGE	The ISO 3166 language to which the supplier is assigned - see Supported Locales.	No	VARCHAR2 (3)
LOCALE_COUNTRY	The ISO 3166 2- (or 3-) letter country code to which the supplier is assigned.	No	VARCHAR2 (3)
CURRENCY_CODE	The ISO 4217 currency code of the supplier.	No	VARCHAR2 (3)
RETURN_ALLOWED	Flag indicating if the supplier will accept returns - Y or N.	No	VARCHAR2 (1)
AUTHORIZATION_REQUIRED	Flag indicating if returns must be accompanied by an authorization number - Y or N.	No	VARCHAR2 (1)
PO_CREATE_ALLOWED	Flag indicating if purchase orders can be created - Y or N.	No	VARCHAR2 (1)
VENDOR_CHECK	Flag indicating if orders from this supplier will require vendor control - Y or N.	No	VARCHAR2 (1)
VENDOR_CHECK_PERCENT	The percentage of items per receipt that will be marked for vendor checking.	No	NUMBER (12, 4)
PARENT_ID	The identifier of the parent supplier.	No	VARCHAR2 (128)
QUANTITY_LEVEL	The supplier order quantity level - CA (Case) or EA (Each).	Yes	VARCHAR2 (6)
TAX_ID	The unique tax identification number of the supplier.	No	VARCHAR2 (18)
DELIVERY_DISCREPANCY_TYPE	The delivery discrepancy type - 0 (Allow), 1 (Overage) or 2 (Restricted).	No	NUMBER (2, 0)

Example CSV File

IDL-SUPPLIER-*.csv

1,1111,Supplier 1,A,en,GB,GBP,Y,N,Y,Y,12345678.1234,Parent Of 1,CA,1234,0

Supplier Item Country File

Table 2-31 Supplier Item Country File Layout

Field Name	Description	Required	Type
ITEM_ID	The unique identifier of the item.	Yes	VARCHAR2 (25)
SUPPLIER_ID	The unique identifier of the supplier.	Yes	NUMBER (10, 0)
COUNTRY_ID	The ISO 3166 2- (or 3-) letter country code.	Yes	VARCHAR2 (3)
CASE_SIZE	The default number of items within a case from the supplier.	No	NUMBER (12, 4)
UNIT_COST_CURRENCY	The unit cost currency of the item for that supplier in that country.	No	VARCHAR2 (3)

Table 2-31 (Cont.) Supplier Item Country File Layout

Field Name	Description	Required	Type
UNIT_COST_VALUE	The unit cost of the item for that supplier in that country.	No	NUMBER (12, 4)

The primary key comprises the ITEM_ID, SUPPLIER_ID and COUNTRY_ID fields.

Example CSV File

IDL-SUPPLIERITEMCOUNTRY-*.csv

1,1,GB,12345678.9012,GBP,11111111.1111

Supplier Item Country Dimension File

Table 2-32 Supplier Item Country Dimension File Layout

Field Name	Description	Required	Type
ITEM_ID	The unique identifier of the item - references the ITEM_ID column in the SUPPLIER_ITEM_COUNTRY table.	Yes	VARCHAR2 (25)
SUPPLIER_ID	The unique identifier of the supplier - references the SUPPLIER_ID column in the SUPPLIER_ITEM_COUNTRY table.	Yes	NUMBER (10, 0)
COUNTRY_ID	The ISO 3166 2- (or 3-) letter country code - references the COUNTRY_ID column in the SUPPLIER_ITEM_COUNTRY table.	Yes	VARCHAR2 (3)
DIMENSION_OBJECT	The dimension object.	Yes	VARCHAR2 (6)
PRESENTATION_METHOD	The packaging (if any) being taken into consideration in the specified dimensions.	No	VARCHAR2 (6)
LENGTH	The length of the dimension object.	No	NUMBER (12, 4)
WIDTH	The width of the dimension object.	No	NUMBER (12, 4)
HEIGHT	The height of the dimension object.	No	NUMBER (12, 4)
DIMENSION_UOM	The unit of measurement for length, width and height.	No	VARCHAR2 (4)
WEIGHT	The weight of the dimension object.	No	NUMBER (12, 4)
NET_WEIGHT	The net weight of the dimension object.	No	NUMBER (12, 4)
WEIGHT_UOM	The unit of measurement for weight.	No	VARCHAR2 (4)
LIQUID_VOLUME	The liquid volume or capacity of the dimension object.	No	NUMBER (12, 4)
LIQUID_VOLUME_UOM	The unit of measurement for liquid volume.	No	VARCHAR2 (4)
STATISTICAL_CUBE	The statistical value of the dimension object's dimensions to be used for loading purposed.	No	NUMBER (12, 4)

The primary key comprises the ITEM_ID, SUPPLIER_ID, COUNTRY_ID and DIMENSION_OBJECT fields.

Example CSV File

IDL-SUPPLIERITEMCOUNTRYDIM-*.csv

1,1,GB,CASE,BARE,1,1,1,M,1.1,1.01,KG,0.1,ML,1

Supplier Item File

Table 2-33 Supplier Item File Layout

Field Name	Description	Required	Type
ITEM_ID	The unique identifier of the item - references the ITEM_ID column in the ITEM table.	Yes	VARCHAR2 (25)
SUPPLIER_ID	The unique identifier of the supplier - this references the ID column in the SUPPLIER table and forms the primary key together with the ITEM_ID field.	Yes	NUMBER (10, 0)
VENDOR_PRODUCT_NUMBER	The vendor product number.	No	VARCHAR2 (256)
IS_PRIMARY	Flag indicating if the supplier is the primary supplier for this item - Y or N.	No	VARCHAR2 (3)

Example CSV File

IDL-SUPPLIERITEM-*.csv

1,1,1,Y

Supplier Item Manufacture File

Table 2-34 Supplier Item Manufacture File Layout

Field Name	Description	Required	Type
ITEM_ID	The unique identifier of the item - references the ITEM_ID column in the ITEM table.	Yes	VARCHAR2 (25)
SUPPLIER_ID	The unique identifier of the supplier - references the ID column in the SUPPLIER table.	Yes	NUMBER (10, 0)
COUNTRY_ID	The ISO 3166 2- (or 3-) letter country code.	Yes	VARCHAR2 (3)
IS_PRIMARY	Flag indicating if this is the primary country of manufacture - Y or N.	No	VARCHAR2 (1)

The primary key comprises the ITEM_ID, SUPPLIER_ID and COUNTRY_ID fields.

Example CSV File

IDL-SUPPLIERITEMMANUFACTURE-*.csv

1,1,GB,Y

Supplier Item UOM File

Table 2-35 Supplier Item UOM File Layout

Field Name	Description	Required	Type
ITEM_ID	The unique identifier of the item - this references the ITEM_ID column in the SUPPLIER_ITEM table.	Yes	VARCHAR2 (25)
SUPPLIER_ID	The unique identifier of the supplier - this references the SUPPLIER_ID column in the SUPPLIER_ITEM table.	Yes	NUMBER (10, 0)
UNIT_OF_MEASURE	The unit of measure - this references the UOM column in the UOM_CLASS table.	Yes	VARCHAR2 (4)
VALUE	The equivalent value of the item/suppliers shipping carton in the associated unit of measure.	Yes	NUMBER (20, 4)

The primary key comprises the ITEM_ID, SUPPLIER_ID and UNIT_OF_MEASURE fields.

Example CSV File

IDL-SUPPLIERITEMUOM-*.csv

1,1,g,1234567890123456.7890

Supplier Organization File

Table 2-36 Supplier Organization File Layout

Field Name	Description	Required	Type
SUPPLIER_ID	The unique identifier of the supplier - this references the ID column in the SUPPLIER table.	Yes	NUMBER (10, 0)
ORGANIZATION_UNIT_ID	The organization unit identifier - forms a unique key together with the SUPPLIER_ID field.	Yes	VARCHAR2 (15)

Example CSV File

IDL-SUPPLIERORGANIZATION-*.csv

1,Org 1

Transfer Zone File

Table 2-37 Transfer Zone File Layout

Field Name	Description	Required	Type
TRANSFER_ZONE	The unique identifier of the transfer zone.	Yes	VARCHAR2 (128)
DESCRIPTION	The description of the transfer zone.	Yes	VARCHAR2 (255)

Example CSV File

IDL-TRANSFERZONE-*.csv

1,DESCRIPTION FOR TRANSFER ZONE 1

UDA File

Table 2-38 UDA File Layout

Field Name	Description	Required	Type
ID	The unique identifier of the user defined attribute.	Yes	NUMBER (5, 0)
TYPE	The type of user defined attribute - FF, DT or LV.	Yes	VARCHAR2 (2)
DESCRIPTION	The description of the user defined attribute.	Yes	VARCHAR2 (120)
PRINT_TICKET	Flag indicating if item tickets should be printed for this user defined attribute - Y or N.	Yes	VARCHAR2 (1)
PRINT_LABEL	Flag indicating if item labels should be printed for this user defined attribute - Y or N.	Yes	VARCHAR2 (1)

Example CSV File

IDL-UDA-*.csv

1,FF,DESCRIPTION FOR 1,Y,Y

UDA LOV File

Table 2-39 UDA LOV File Layout

Field Name	Description	Required	Type
UDA_ID	The unique identifier of the user defined attribute - this references the ID column in the UDA table.	Yes	NUMBER (5, 0)
LOV_ID	The identifier for the LV (List of Values) user defined attribute value - forms the primary key together with the UDA_ID field.	Yes	VARCHAR2 (25)
DESCRIPTION	The description of the user defined attribute value.	Yes	VARCHAR2 (250)

Example CSV File

IDL-UDALOV-*.csv

3,LOV_1,DESCRIPTION FOR LOV_1

UOM Class File

Table 2-40 UOM Class File Layout

Field Name	Description	Required	Type
UOM	The unique identifier of the unit of measure.	Yes	VARCHAR2 (4)
UOM_CLASS	The type of unit of measure - AREA, DIMEN, LVOL, MASS, MISC, PACK, QTY or VOL.	Yes	VARCHAR2 (6)
DESCRIPTION	The description of the unit of measure.	Yes	VARCHAR2 (120)

Example CSV File

IDL-UOMCLASS-*.csv

g,MASS,DESCRIPTION FOR 'GRAM'

UOM Conversion File

Table 2-41 UOM Conversion File Layout

Field Name	Description	Required	Type
FROM_UOM	The unit of measure to convert from - this references the UOM column in the UOM_CLASS table.	Yes	VARCHAR2 (4)
TO_UOM	The unit of measure to convert to - this references the UOM column in the UOM_CLASS table and forms the primary key together with the FROM_UOM field.	Yes	VARCHAR2 (4)
FACTOR	The factor to apply when converting the unit of measure.	Yes	NUMBER (20, 10)

Example CSV File

IDL-UOMCONVERSION-*.csv

g,lb,453.592

Warehouse Address File

Table 2-42 Warehouse Address File Layout

Field Name	Description	Required	Type
EXTERNAL_ID	The external identifier of the address.	Yes	VARCHAR2 (25)
SUPPLIER_ID	The unique identifier of the warehouse - forms a unique key together with the EXTERNAL_ID field.	Yes	NUMBER (10, 0)
ADDRESS_TYPE	The type of address - 01 (Business), 02 (Postal), 03 (Returns), 04 (Order), 05 (Invoice), 06 (Remittance), 07 (Billing), 08 (Delivery) or 09 (External).	Yes	VARCHAR2 (2)

Table 2-42 (Cont.) Warehouse Address File Layout

Field Name	Description	Required	Type
IS_PRIMARY	Flag indicating if this is the primary address - Y or N.	Yes	VARCHAR2 (1)
ADDRESS_LINE_1	The first line of the address.	No	VARCHAR2 (240)
ADDRESS_LINE_2	The second line of the address.	No	VARCHAR2 (240)
ADDRESS_LINE_3	The third line of the address.	No	VARCHAR2 (240)
CITY	The city.	No	VARCHAR2 (120)
STATE	The state.	No	VARCHAR2 (3)
COUNTRY_ID	The ISO 3166 2- (or 3-) letter country code.	No	VARCHAR2 (3)
POSTAL_CODE	The postal code.	No	VARCHAR2 (30)
CONTACT_NAME	The contact name.	No	VARCHAR2 (120)
CONTACT_PHONE	The contact phone number.	No	VARCHAR2 (20)
CONTACT_FAX	The contact fax number.	No	VARCHAR2 (20)
CONTACT_EMAIL	The contact email address.	No	VARCHAR2 (100)
COUNTY	The county.	No	VARCHAR2 (250)

Example CSV File

IDL-WAREHOUSEADDR-*.csv

1,1,01,Y,Line 1,Line 2,Line 3,City,MN,USA,Postcode,Contact Name,Contact_Phone,Contact_Fax,Contact_Email,County

Warehouse Class File

Table 2-43 Warehouse File Layout

Field Name	Description	Required	Type
ID	The unique identifier of the warehouse.	Yes	NUMBER (10, 0)
NAME	The name of the warehouse.	Yes	VARCHAR2 (150)
ORGANIZATION_UNIT_ID	The organization unit identifier of the warehouse.	No	VARCHAR2 (15)
LOCALE_COUNTRY	The ISO 3166 2- (or 3-) letter country code.	No	VARCHAR2 (3)
CURRENCY_CODE	The ISO 4217 currency code of the warehouse.	No	VARCHAR2 (40)
PHYSICAL_WH	The identifier of the physical warehouse corresponding to the warehouse.	Yes	NUMBER (10, 0)
PRIMARY_VWH	The identifier of the primary virtual warehouse corresponding to the warehouse.	No	NUMBER (10, 0)
NAME_SECONDARY	The secondary name of the warehouse.	No	VARCHAR2 (150)

Table 2-43 (Cont.) Warehouse File Layout

Field Name	Description	Required	Type
STOCKHOLDING_IND	Flag indicating if the warehouse is a stock holding location.	No	VARCHAR2 (1)
DUNS_NUMBER	The Dun and Bradstreet number to identify the location.	No	VARCHAR2 (9)
DUNS_LOC	The Dun and Bradstreet number to identify the location.	No	VARCHAR2(4)
TSF_ENTITY_ID	The transfer entity identifier of the warehouse.	No	NUMBER (10, 0)
INBOUND_HANDLING_DAYS	The number of days that the warehouse requires to receive any item and get it to the shelf so that it is ready to pick.	No	NUMBER (2, 0)
CHANNEL_ID	The channel identifier of the warehouse.	No	NUMBER (4, 0)
CHANNEL_NAME	The name of the channel.	No	VARCHAR2 (120)
FINISHER_IND	Flag indicating if the warehouse is a finisher - Y or N.	No	VARCHAR2 (1)
EMAIL	The email address of the warehouse.	No	VARCHAR2 (100)

All records in this file will be used to populate the WAREHOUSE_VIRTUAL (Virtual Warehouse) table. Records where the ID and PHYSICAL_WH match will be used to populate the WAREHOUSE (Physical Warehouse) table with a subset of the fields: ID, NAME, ORGANIZATION_UNIT_ID, LOCALE_COUNTRY and CURRENCY_CODE.

Example CSV File

IDL-WAREHOUSE-*.csv

1,Virtual Warehouse 1,Org Unit ID 1,GB,GBP,1,11,Secondary Name 1,Y,D&B NUM 1,LOC1,1234567890,96,1234,CHANNEL 1234,N,warehouse1@abc.com

Warehouse Item File

Table 2-44 Warehouse Item File Layout

Field Name	Description	Required	Type
ITEM_ID	The unique identifier of the item - references the ITEM_ID column in the ITEM table.	Yes	VARCHAR2 (25)
WAREHOUSE_ID	The unique identifier of the warehouse - this references the ID column in the WAREHOUSE table and forms the primary key together with the ITEM_ID field.	Yes	NUMBER (10, 0)
STATUS	The status of the warehouse item - ' ' (None), A (Active), C (Discontinued), I (Inactive), D (Deleted), Q (Auto-stocked) or N (Non-ranged).	Yes	VARCHAR2 (2)
QUANTITY_TOTAL	The total quantity of the warehouse item.	Yes	NUMBER (12, 4)
QUANTITY_RESERVED	The reserved quantity of the warehouse item.	Yes	NUMBER (12, 4)

Table 2-44 (Cont.) Warehouse Item File Layout

Field Name	Description	Required	Type
QUANTITY_UNAVAILABLE	The unavailable quantity of the warehouse item.	Yes	NUMBER (12, 4)
QUANTITY_IN_TRANSIT	The in transit quantity of the warehouse item.	Yes	NUMBER (12, 4)
STANDARD_UOM	The standard unit of measure of the warehouse item.	No	VARCHAR2 (4)

Example CSV File

IDL-WAREHOUSEITEM-*.csv

1,1,A,12345678.9012,34567890.1234,56789012.3456,78901234.5678,kg

Transactional Data Seeding

Transaction data seeding in a SIOCS installation is achieved by uploading data in CSV (comma-separated values) files to Object Storage via FTS (File Transfer Service).

After files are uploaded to Object Storage, The Initial Foundation Data File Import and Initial Store Data File Import batch jobs download the relevant files from Object Storage and import the data into SIOCS. For transaction data modules which require store, the customer admin user would need to run Store Data File Import batch and provide store id as input. For modules which do not require store id, run Initial Foundation Data File Import job.

Process Flow

This is a general overview of the process flow.

- Each file that loaded is broken into groupings of transactions (1000 per group). Each grouping is given a processing number and a request to process the data is places in MPS.
- The MPS messages system will grab processing requests off the queue and process each group (of 1000) at a time committing transactions that are successful and failing transactions that have problems.
- The user can monitor the process both through the MPS Staged Message screen and the Integration Dashboard screen.
- The user can then export the errors, make corrections, and reload a file. (See Errors and Reprocessing.)

Process Ordering

The processing of sets of data needs to be in order with one set of data being completed before the next begins.

> **Note:**
>
> Data seeding of foundation data and data setup should be completed prior to data seeding transactional data.

Purchase Order Group

Purchase orders and DSDs are loaded for each individual store. The purchase orders for a single store must be fully loaded and finished processing, along with error corrections, prior to loading DSD (Deliveries from Vendors) information for the same store.

Transfer Group

Transfers are not loaded for each individual store. Transfers must be fully loaded and finish processing, along with desired error corrections, prior to loading additional transfer information. Once transfers are loaded, you load transfer information in the following sequence per store: allocation, transfer shipment, transfer delivery.

UIN

The UIN file is loaded by store. Loading in stock UINs is dependent only on the foundation data.

Errors and Reprocessing

When errors occur, they must be manually dealt with by the user.

- During processing, transactions that fail at any level (header, carton, detail) will fail the entire transaction.
- The number of failures for a particular data type can be seen in the Integration Dashboard.
- You can load the same file for different stores without issue (such as loading DSDs for Store 1 and Store 2) prior to dealing with errors, however, you should not load the same data type file for the same store without first clearing out the errors. For example, do not load DSDs for Store 1 and again for Store 1 without first dealing with errors.
- The integration dashboard will allow a user to export error data back out for examination and correction. If more than one store worth of errors exists, it will create one file for each store on export.
- Once the data is exported, the errors for the data type should be deleted prior to re-loading additional information. For example, load transfer shipments for store 1, export the failures, delete the failures, correct the failures, reload ONLY the corrected data from transfer shipments for store 1 again. Removal of erroneous data can be done through the integration dashboard.

> **Note:**
>
> Once a transaction is successfully processed and reaches the transaction tables without error, it cannot be loaded again. Additional attempts to load the data will fail with duplicate data errors. Data seeding will not perform updates on currently existing data.

Volume Considerations

This is a general overview of the process flow for basic understanding.

- To prevent system overload, file sizes should remain under or around 100,000 transactions (that is 100,000 header rows or overall transactions, not rows in file).
- It is recommended that only 1 or 2 files be loaded at a time and that the processing is completed on these files prior to loading more files.

Integration Dashboard

The integration dashboard screen (see Technical Maintenance Screens – Integration Dashboard) displays the current state of integration processing, which includes the transaction data seeding. This screen can be used to do the following:

- See the number of records currently processing for a data type
- See the number of failed records for a data type
- Export error data back out to file
- Clear error data out so that reprocessing of a file can occur

Transaction Data Seeding Modules

Table 2-45 Transaction Data Seeding Modules

Module	Description
Allocation	Loads warehouse to store transfer allocation information by store.
DSD	Loads direct store delivery information by store.
Purchase Order	Load purchase order information by store.
Transfer	Load transfer document information
Transfer Shipment	Loads transfer shipment information by shipping store.
Transfer Delivery	Load transfer delivery information by receiving store.
UIN	Load basic UIN information for a limited set of statuses.

File Layouts

- All files should be in CSV (comma-separated values) format, with either a ".csv" or ".dat" filename extension. The batch jobs also support zipped files which will be extracted upon download and processed individually.
- Empty or blank fields within a record will be considered null. Every column must be present even if it is empty or null.
- String fields containing a comma or double quote must be quoted (with double quotes), a double quote in a field must be represented by 2 double quote characters. Line breaks within quoted fields are not supported.

- The filename format is IDL-[MODULENAME]-XXXX.csv(/dat/zip). Files contained within .zip files must adhere to the same filename format.
- To run the Initial Store Data File Import batch job for a particular Store, the filename format should be IDL-{moduleName}-{StoreId]-{fileNum}.csv. Any file which does not adhere to the filename format will not be downloaded or processed. Also, files > 500MB will be rejected: the file should be split into smaller files and uploaded as a .zip file.
- It is recommended to not edit the .csv files in Excel as this can lead to formatting issues.
- Rows within the file can have different layouts. The row type column located first in any row defines what kind of row it is and the format it must follow.

File Date Requirements

- All columns noted as required must have values within the file or the entire file will be failed.
- The data within the file for a particular column must match the data type of the column or the entire file will be failed.
- Dates must be entered in the format YYYY-MM-DD HH:MM:SS (examples: 2022-12-06 14:34:21).
- Dates must be GMT as the file will parse and process the dates as GMT dates.

Allocation File

Allocation files must contain information for a single store only.

Table 2-46 Allocation File Row Layout

Field Name	Description	Required	Type
IMPORT_ALLOC_ID	A unique identifier of this imported allocation.	Yes	VARCHAR2(128)
EXTERNAL_ID	The unique allocation identifier from an external system.	Yes	NUMBER(12)
ITEM_ID	The unique identifier of the item to be delivered.	Yes	VARCHAR2(25)
STORE_ID	The unique identifier of the store receiving the allocation.	Yes	NUMBER(10)
WAREHOUSE_ID	The unique identifier of the warehouse shipping the item.	Yes	NUMBER(10)
STATUS	The status of the allocation	Yes	NUMBER(2)
DISTRIBUTION_PARENT_ID	The unique identifier of the parent transfer document.	No	VARCHAR2(25)
DELIVERY_DATE	The date the allocation is expected to be delivered.	No	DATE
CONTEXT_ID	The identifier of a context associated to the allocation.	No	NUMBER(18)
CONTEXT_VALUE	A value associated to the context	No	VARCHAR2(25)

Table 2-46 (Cont.) Allocation File Row Layout

Field Name	Description	Required	Type
DELIVERY_SLOT_ID	The unique identifier of the delivery slot of expected delivery time.	No	NUMBER(15)
QUANTITY_EXPECTED	The quantity expected to be delivered.	No	NUMBER(20,4)
QUANTITY_RECEIVED	The quantity that has been received.	No	NUMBER(20,4)
QUANTITY_DAMAGED	The quantity that has been received as damaged.	No	NUMBER(20,4)

Data Definition

Status: (1) Approved, (2) Completed, (3) Canceled

Example CSV File

For a store-based transaction import, the file name must have the fileNum, IDL-ALLOCATION-<storeId>-<fileNum>.csv

Example:

IDL-ALLOCATION-1111-1.csv

1,5001,100637113,5000,9000,1,1234,2022-10-14 10:40:21,145,CV145,4523026194,100,0,0

DSD File

- Direct Store Delivery files must contain information for a single store only.
- Purchase order must be loaded and complete processing prior to loading direct store deliveries.
- Each delivery must have at least one carton associated to it.
- Each carton must have at least one item associated to it.
- The status of the delivery is not uploaded, but rather calculated from the status of the cartons.
- UINs are not loaded as part of this transfer delivery data seeding file upload.

Table 2-47 DSD File Row Layout (H – Header)

Field Name	Description	Required	Type
ROW_TYPE	The type of row that is represented.	Yes	“H”
IMPORT_DSD_ID	The unique identifier of the delivery record.	Yes	VARCHAR2(128)
IMPORT_PO_ID	The purchase order that the delivery is associated to.	Yes	VARCHAR2(128)
STORE_ID	The unique identifier of the store receiving the inventory.	Yes	NUMBER(10)

Table 2-47 (Cont.) DSD File Row Layout (H – Header)

Field Name	Description	Required	Type
SUPPLIER_ID	The unique identifier of the supplier shipping the inventory.	Yes	NUMBER(10)
ORIGIN_TYPE	The origin type of the delivery.	Yes	NUMBER(2)
RECEIPT_NO		Yes	NUMBER(12)
ASN_ID	The advanced shipping notification of the delivery.	No	VARCHAR2(128)
INVOICE_ID	A unique identifier of an invoice associated to this delivery.	No	VARCHAR2(128)
INVOICE_DATE	The date of the delivery invoice.	No	DATE
CURRENCY_CODE	A currency code identifying the type of currency.	No	VARCHAR2(3)
INVOICE_COST_VALUE	The cost of the invoice.	No	NUMBER(12,4)
CARRIER_ENTITY	The name of the carrier.	No	VARCHAR2(128)
CARRIER_TYPE	The type of the carrier.	No	NUMBER(2)
CARRIER_CODE	Unique code that identifies the carrier.	No	VARCHAR2(4)
COUNTRY_CODE	A country code.	No	VARCHAR(3)
SOURCE_ADDRESS	The address of the source sending the delivery to the store.	No	VARCHAR2(1000)
LICENSE_PLATE	The license plate of the delivery vehicle.	No	VARCHAR2(128)
FREIGHT_ID	A freight identifier associated to the delivery.	No	VARCHAR2(128)
BOL_EXTERNAL_ID	An external identifier of a bill of lading record.	No	VARCHAR2(128)
FISCAL_DOCUMENT_ID	The Fiscal Document Number from a fiscal document system.	No	VARCHAR2(128)
EXPECTED_DATE	The expected date of the delivery.	No	DATE
RECEIVED_DATE	The date the delivery was received.	No	DATE
RECEIVED_USER	The user who received the delivery record.	No	VARCHAR2(128)
CREATE_DATE	The date the delivery record was created.	Yes	DATE
CREATE_USER	The user that created the delivery record.	No	VARCHAR2(128)
UPDATE_DATE	The date the delivery record was last updated.	No	DATE
UPDATE_USER	The user who last updated the delivery record.	No	VARCHAR2(128)

Table 2-48 DSD Row Layout (C – Carton)

Field Name	Description	Required	Type
ROW_TYPE	The type of row that is represented.	Yes	“C”
IMPORT_DSD_ID	An identifier assigned to the DSD header to tie it to cartons and items within the file.	Yes	VARCHAR2(128)
IMPORT_CARTON_ID	An import identifier assigned to the carton to tie it to the items.	Yes	VARCHAR2(128)
EXTERNAL_ID	An external identifier of the carton used during integration publication.	Yes	VARCHAR2(128)
STATUS	The carton status (see Index).	Yes	NUMBER(4)
REFERENCE_ID	A Reference identifier.	No	VARCHAR2(128)
DAMAGED_REASON	The reason for container damage.	No	VARCHAR2(128)
TRACKING_NUMBER	The tracking number for the container.	No	VARCHAR2(128)
SERIAL_CODE	The serial code.	No	NUMBER(18)
DAMAGE_REMAINING	Indicates all remaining quantities should be damaged on final receipt.	No	VARCHAR2(1)
UIN_REQUIRED	The item UIN, Y if UIN item exists in container, else No	Yes	VARCHAR2(1)
RECEIVE_AT_SHOP_FLOOR	Y if receive at shop floor, else No	Yes	VARCHAR2(1)
QUALITY_CONTROL	Y indicates the container is flagged for detailed receiving.	Y	VARCHAR2(1)
EXTERNAL_CREATE	Yes indicates whether the delivery is external created. Valid values: Y or N.	Yes	VARCHAR2(1)
ADJUSTED	Indicates whether the delivery is adjusted. Valid values: Y or N	Yes	VARCHAR2(1)
RECEIVE_DATE	The date when the vendor delivery carton was received.	No	DATE
RECEIVE_USER	The user who received the vendor delivery carton.	No	VARCHAR2(128)
CREATE_DATE	The date when the vendor delivery carton was created.	Yes	DATE
CREATE_USER	The user who created the vendor delivery carton.	No	VARCHAR2(128)
UPDATE_DATE	The date when the vendor delivery carton was updated.	No	DATE
UPDATE_USER	The user who last updated the vendor delivery carton.	No	VARCHAR2(128)

Table 2-49 DSD Row Layout (D – Detail)

Field Name	Description	Required	Type
ROW_TYPE	The type of row that is represented.	Yes	“D”
IMPORT_DSD_ID	Identifier assigned to the DSD header to tie it to cartons and items within the file.	Yes	VARCHAR2(128)
IMPORT_CARTON_ID	Import identifier assigned to the carton to tie it to the items.	Yes	VARCHAR2(128)
ITEM_ID	The unique identifier of the item that is contained in this carton.	Yes	VARCHAR2(25)
CASE_SIZE	The number of units in the case that this item was shipped in.	Yes	NUMBER(10,2)
QUANTITY_EXPECTED	The total number of units expected in this direct delivery.	No	NUMBER(20,4)
QUANTITY_RECEIVED	The total number of units received in this direct delivery.	No	NUMBER(20,4)
QUANTITY_DAMAGED	The total number of units that were damaged when the direct delivery was received.	No	NUMBER(20,4)
QUANTITY_RECEIVED_OVER	Amount of received inventory over the expected quantities.	No	NUMBER(20,4)
QUANTITY_DAMAGED_OVER	Amount of damaged inventory over the expected quantities.	No	NUMBER(20,4)
PREVIOUS_RECEIVED	Units previous received when container is reopened for adjustment.	No	NUMBER(20,4)

Table 2-49 (Cont.) DSD Row Layout (D – Detail)

Field Name	Description	Required	Type
PREVIOUS_DAMAGED	Units previous received as damaged when container is reopened for adjustment.	No	NUMBER(20,4)
UNIT_COST_CURRENCY	The unit cost currency code for the line item.	No	VARCHAR2(3)
UNIT_COST_VALUE	The unit cost value for the line item.	No	NUMBER(12,4)
OVERRIDE_UNIT_COST_CURRENCY	The override unit cost currency.	No	VARCHAR2(3)
OVERRIDE_UNIT_COST_VALUE	The override unit cost value.	No	NUMBER(12,4)
IMPORT_PO_ID	The import identifier used in the PO upload file that this DSD item is associated to.	Yes	VARCHAR2(128)

Data Definition

Delivery Origin Type: (0) Asn, (1) PO, (2) DexNex, (3) Manual

Delivery Carrier Type: (0) Corporate, (1) Third Party

Carton Status: (1), New, (2) In Progress, (3) Submitted, (4) Received, 5 (Damaged), 6 (Missing), (7) Canceled

Example CSV File

For a store-based transaction import, the file name must have the fileNum, IDL-VENDORDELIVERY-<storeId>-<fileNum>.csv

Example:

IDL-VENDORDELIVERY-1111-1.csv

H,4,PO 02 TRY,5000,5115,1,3,POASN1,1,2022-10-10,USD,100.00,CI1,1,0,US,SA,LP1,FR1,BOLEID1,FDID1,2022-10-10,2022-10-10,1500,2022-10-10,15000,2022-10-10,15000

C,4,-1,EID1,1,REFID1,NO DAMAGED,TN1,0599123645,N,N,Y,Y,Y,N,2022-10-10 17:12:21,15000,2022-10-10 17:12:21,1500,2022-10-10 17:12:21,1500

D,4,-1,100668163,1,30,30,0,0,0,0,0,USD,50.00,USD,50.00,PO 02 TRY

Purchase Order File

Purchase order files must contain information for a single store only. For each “H” header record, there must be at least one “D” detail record.

Table 2-50 Purchase Order File Row Layout (H – Header)

Field Name	Description	Required	Type
ROW_TYPE	The type of row that is represented.	Yes	“H” (Header)
IMPORT_PO_ID	A unique identifier of this imported purchase order.	Yes	VARCHAR2(128)
EXTERNAL_ID	An identifier of this purchase order from an external system.	Yes	VARCHAR2(128)
STORE_ID	The identifier of the store this purchase order is for.	Yes	NUMBER(10)
SUPPLIER_ID	The identifier of the supplier this purchase order is from.	Yes	VARCHAR2(128)
STATUS	The status or the purchase order.	Yes	NUMBER(4)
EXTERNAL_STATUS	The status of the purchase order in the originating system.	Yes	NUMBER(4)
NOT_BEFORE_DATE	Earliest date that the inventory should arrive at the store.	No	DATE
NOT_AFTER_DATE	Latest date that the inventory should arrive at the store.	No	DATE
USER_ID	User who originated the purchase order.	No	VARCHAR2(128)
COMMENTS	Comments associated to the purchase order.	No	VARCHAR2(2000)
CUST_ORDER_ID	The external identifier of a customer order associated to the purchase order.	No	VARCHAR2(128)
FUL_ORD_EXTERNAL_ID	The external identifier of the fulfilment order associated to the order.	No	VARCHAR2(128)
SOURCE	The originating source of the purchase order.	Yes	VARCHAR2(25)
CREATE_DATE	The date the purchase order was created.	Yes	DATE
UPDATE_DATE	The date the purchase order was updated.	No	DATE
COMPLETE_DATE	The date the purchase order was completed.	No	DATE

Table 2-51 Purchase Order File Row Layout (D – Detail)

Field Name	Description	Required	Type
ROW_TYPE	The type of row that is represented.	Yes	“D” (Detail)

Table 2-51 (Cont.) Purchase Order File Row Layout (D – Detail)

Field Name	Description	Required	Type
IMPORT_PO_ID	The unique identifier from the (H)eader row this detail is associated to.	Yes	VARCHAR2(128)
ITEM_ID	The unique sku number.	Yes	VARCHAR2(25)
SUPPLIER_COUNTRY	The supplier country of origin	Yes	VARCHAR(3)
CASE_SIZE	The case size coming from the supplier.	Yes	NUMBER(10,2)
QUANTITY_EXPECTED	The number of units expected to be delivered to the store.	Yes	NUMBER(20,4)
QUANTITY_RECEIVED	The number of units received to date against the order.	No	NUMBER(20,4)
UNIT_COST_CURRENCY	The unit cost ISO currency code.	No	VARCHAR2(3)
UNIT_COST_VALUE	The unit cost value of the item.	No	NUMBER(12,4)
PREFERRED_UOM	The preferred unit of measure of this item on the order.	No	VARCHAR(4)

Data Definition

Purchase Order Status: (1) New, (2) In Progress, (3) Canceled, (4) Completed

Purchase Order External Status: (1) Worksheet, 2() Submitted, (3) Approved, (4) Closed

Example CSV File

For a store-based transaction import, the file name must have the fileNum, IDL-PURCHASEORDER-<storeId>-<fileNum>.csv

Example:

IDL-PURCHASEORDER-1111-1.csv

H,abcde,EXTID1,5000,5100,1,2,2022-10-06 00:00:00,2022-10-06 00:00:00,15000,NO COMMENTS,1000,POIDSLFILE1,SIOCS,2022-10-06 12:07:01,2022-10-06 12:07:02,2022-10-06 12:07:10

D,abcde,100654087,US,1,100,100,USD,1.55,EA

Transfer File

- There must be a least one detail row for each header row.
- Reserved quantities will be incremented by any remaining quantities for the item at the source location.
- If unavailable inventor is used, the unavailable inventory will be decremented at the source location.
- The transfer quantities are considered final and correct. Therefore, shipments and deliveries referencing the transfer and loaded later will not update the transfer information.

Table 2-52 Transfer File Row Layout (H – Header)

Field Name	Description	Required	Type
ROW_TYPE	The type of row that is represented.	Yes	“H”
IMPORT_TSF_ID	An import identifier to tie this header with line items.	Yes	VARCHAR2(128)
EXTERNAL_ID	An external identifier supplied from an external system.	No	VARCHAR2(128)
DISTRO_NUMBER		Yes	VARCHAR2(128)
SOURCE_TYPE	Source location type.	Yes	NUMBER(2)
SOURCE_ID	The identifier of the source location of the transfer.	Yes	NUMBER(10)
DESTINATION_TYPE	Destination location type.	Yes	NUMBER(2)
DESTINATION_ID	The identifier of the destination location of the transfer.	Yes	NUMBER(10)
STATUS	The transfer status.	Yes	NUMBER(2)
ORIGIN_TYPE	The origin type of the transfer.	Yes	NUMBER(2)
CONTEXT_ID	Unique identifier of a context associated to the transfer.	No	NUMBER(18)
CONTEXT_VALUE	A value or some information related to the context associated to the transfer.	No	VARCHAR2(25)
FUL_ORD_EXTERNAL_ID	External system identifier of the fulfillment order.	No	VARCHAR2(128)
CUST_ORD_EXTERNAL_ID	External system identifier of the customer order.	No	VARCHAR2(128)
USE_AVAILABLE	The Use Available, Y indicates the transfer must use available stock, N indicates it uses unavailable stock.	Yes	VARCHAR2(1)
ALLOW_PARTIAL_DELIVERY	Y indicates that the partial delivery is allowed for the transfer, N indicates it is not.	Yes	VARCHAR2(1)
AUTHORIZATION_CODE	An authorization code required for the transfer.	No	VARCHAR2(12)
NOT_AFTER_DATE	Date after which the transfer is no longer valid.	No	DATE
REQUEST_DATE	The date the transfer was requested.	No	DATE

Table 2-52 (Cont.) Transfer File Row Layout (H – Header)

Field Name	Description	Required	Type
REQUEST_USER	The user that requested the transfer.	No	VARCHAR2(128)
APPROVAL_DATE	The date the transfer was approved.	No	DATE
APPROVAL_USER	The user that approved the transfer.	No	VARCHAR2(128)
CREATE_DATE	The date this record was created.	Yes	DATE
CREATE_USER	The user that created this record.	No	VARCHAR2(128)
UPDATE_DATE	The date this record was last updated.	No	DATE
UPDATE_USER	The user that last updated this record.	No	VARCHAR2(128)

Table 2-53 Transfer File Row Layout (D – Detail)

Field Name	Description	Required	Type
ROW_TYPE	The type of row that is represented.	Yes	“D”
IMPORT_TSF_ID	The unique transfer identifier.	Yes	VARCHAR2(128)
ITEM_ID	The item identifier.	Yes	VARCHAR2(25)
CASE_SIZE	The case size associated to this line item.	Yes	NUMBER(10,2)
QUANTITY_REQUESTED	The quantity that was requested.	No	NUMBER(20,4)
QUANTITY_APPROVED	The quantity that was approved.	No	NUMBER(20,4)
QUANTITY_SHIPPING	The quantity that is currently in shipping.	No	NUMBER(20,4)
QUANTITY_SHIPPED	The quantity that has currently shipped.	No	NUMBER(20,4)
QUANTITY_RECEIVED	The quantity that has been received into stock.	No	NUMBER(20,4)
QUANTITY_DAMAGED	The quantity that has been received as damaged.	No	NUMBER(20,4)
PREFERRED_UOM	The preferred unit of measure of the transfer line item.	No	VARCHAR2(4)

Data Definition

Source Type: (1) Store, (3) Warehouse, (4) Finisher

Destination Type: (1) Store, (3) Warehouse, (4) Finisher

Transfer Status: (1) New Request, (2) Requested, (3) Request In Progress, (4) Rejected, (5) Canceled Request, (6) Transfer In Progress, (7) Approved, (8) In Shipping, (9) Completed, (10) Canceled

Transfer Origin Type: (0 External, (1) Internal, (2) Adhoc

Example CSV File

IDL-TRANSFER.csv

H,TSFID1,EXTID1,DN1,1,5000,1,5001,7,1,1,364155194,MOBCO5,MOBFO5,Y,N,AUTHCODE
1,2022-10-30 00:00:01,2022-10-22 09:28:01,1500,2022-10-22 09:28:01,1500,2022-10-22 09:28:02,1500,2022-10-22 09:28:03,1500

D,TSFID1,100701234,1,1,1,1,1,1,0,EA

Transfer Shipment File

- Transfer shipment files must contain information for a single store only.
- Duplicate cartons are not allowed on the shipments
- Cartons not in new status are required to have line items.
- Transfer shipment status will be calculated from the status of the various cartons on the shipment. Transfer shipments should not be imported if they are in submitted status. The complexities of communication and synchronization with third party systems responsible for manifesting or other fiscal documentation makes this not feasible. Transfer shipments that are currently in progress prior to dispatch should be submitted after the import.
- UINs are not loaded as part of this transfer delivery data seeding file upload.

Table 2-54 Transfer Shipment File Row Layout (H – Header)

Field Name	Description	Required	Type
ROW_TYPE	The type of row that is represented.	Yes	“H”
IMPORT_SHIP_ID	An import identifier used to associate the shipment with its cartons and items.	Yes	VARCHAR2(128)
STORE_ID	The unique store identifier that is the source of the shipment.	Yes	NUMER(10)
DESTINATION_TYPE	Destination location type.	Yes	NUMBER(2)
DESTINATION_ID	The unique identifier of the destination.	Yes	NUMBER(10)
ASN	The advance shipment notification number.	Yes	VARCHAR2(128)
NOT_AFTER_DATE	A date that the goods should not be shipped after.	No	DATE
AUTHORIZATION_CODE	An authorization code	No	VARCHAR2(128)

Table 2-54 (Cont.) Transfer Shipment File Row Layout (H – Header)

Field Name	Description	Required	Type
IMPORT_TSF_ID	The original import identifier of a transfer loaded from the transfer file.	No	NUMBER(15)
TRACKING_NUMBER	Holds the tracking number for the transaction.	No	VARCHAR2(128)
SHIP_CARRIER_ID	Identifier representing the carrier for the shipment.	No	NUMBER(10)
SHIP_CARRIER_SERVICE_ID	Identifier representing the carrier service for the shipment.	No	NUMBER(10)
SHIPMENT_CARTON_DIM_ID	The shipment carton dimension Id.	No	NUMBER(12)
SHIP_WEIGHT	The weight of the carton.	No	NUMBER(12,4)
SHIP_WEIGHT_UOM	The weight UOM of the Carton.	No	VARCHAR2(4)
REQUESTED_PICKUP_DATE	The field contains the requested pickup date.	No	DATE
SHIP_TO_ADDRESS_TYPE	The address type for the ship to address.	No	VARCHAR2(2)
ALT_DESTINATION_ADDRESS	This field contains the alternate destination address.	No	VARCHAR2(2000)
CARRIER_ROLE	The carrier type for a Bill of Lading.	Yes	NUMBER(2)
THIRD_PARTY_NAME	This field contains the name of the third party.	No	VARCHAR2(240)
THIRD_PARTY_ADDRESS	This field contains the address of the third party.	No	VARCHAR2(240)
MOTIVE	This field contains the motive.	No	VARCHAR2(120)
TAX_ID	This field contains the tax id of the supplier.	No	VARCHAR2(18)
FISCAL_DOCUMENT_ID	Fiscal Document Number.	No	VARCHAR2(255)
FISCAL_DOCUMENT_URL	Fiscal Document printing URL provided by external system.	No	VARCHAR2(255)
SUBMIT_USER	The user that submitted the shipment record.	No	VARCHAR2(128)

Table 2-54 (Cont.) Transfer Shipment File Row Layout (H – Header)

Field Name	Description	Required	Type
SUBMIT_DATE	The date the shipment was submitted within EICS.	No	DATE
DISPATCH_USER	The user that dispatched the shipment.	No	VARCHAR2(128)
DISPATCH_DATE	The date the shipment was dispatched within EICS.	No	DATE
CREATE_USER	The user that created the shipment record.	No	VARCHAR2(128)
CREATE_DATE	The date the shipment record was created.	Yes	DATE
UPDATE_USER	The user that last updated the shipment.	No	VARCHAR2(128)
UPDATE_DATE	The last date the shipment was updated.	No	DATE

Table 2-55 Transfer Shipment File Row Layout (C – Carton)

Field Name	Description	Required	Type
ROW_TYPE	The type of row that is represented.	Yes	“C”
IMPORT_SHIP_ID	An import shipment identifier to tie the shipment, carton, and items together.	Yes	VARCHAR2(128)
IMPORT_CARTON_ID	A unique import carton identifier to tie the carton to its items.	Yes	VARCHAR2(128)
EXTERNAL_ID	The external identifier.	Yes	VARCHAR2(128)
STATUS	The status of the shipment carton.	Yes	NUMBER(4)
CARTON_DIMENSION_ID	The shipment container dimension id.	No	NUMBER(10)
WEIGHT	The weight of the container.	No	NUMBER(12,4)
WEIGHT_UOM	The UOM of the container.	No	VARCHAR2(4)
TRACKING_NUMBER	The tracking number for the container.	No	VARCHAR2(128)
USE_AVAILABLE	Value of Y indicates carton will use only available inventory, N means carton will use unavailable inventory.	Yes	VARCHAR2(1)
RESTRICTION_LEVEL	The hierarchy restriction level for items in a container.	Yes	NUMBER(4)
APPROVAL_USER	The user who approved the shipment.	No	VARCHAR2(128)

Table 2-55 (Cont.) Transfer Shipment File Row Layout (C – Carton)

Field Name	Description	Required	Type
APPROVAL_DATE	The date when the shipment was approved.	No	DATE
CREATE_USER	The user who created the shipment carton.	No	VARCHAR2(128)
CREATE_DATE	The create date of the shipment carton.	Yes	DATE
UPDATE_USER	The user who last updated the shipment carton.	No	VARCHAR2(128)
UPDATE_DATE	The date when the shipment carton was updated.	No	DATE

Table 2-56 Transfer Shipment File Row Layout (D – Detail)

Field Name	Description	Required	Type
ROW_TYPE	The type of row that is represented.	Yes	“D”
IMPORT_SHIP_ID	An import shipment identifier to tie the shipment, carton, and items together.	Yes	VARCHAR2(128)
IMPORT_CARTON_ID	A unique import carton identifier to tie the carton to its items.	Yes	VARCHAR2(128)
ITEM_ID	The item identifier.	Yes	VARCHAR2(25)
IMPORT_TSF_ID	The import identifier assigned to the transfer in the import file that imported this item in the transfer upload.	Yes	VARCHAR2(128)
SHIPMENT_REASON_ID	The shipment reason identifier.	No	NUMBER(15)
CASE_SIZE	Pack size of the item.	Yes	NUMBER(10,2)
QUANTITY	Quantity to be shipped.	Yes	NUMBER(20,4)

Data Definition

Shipment Destination Type: (1) Store, (3) Warehouse, (4) Finisher

Shipment Ship To Address Type: (01) Business, (02) Postal, (03) Returns, (04) Order, (05) Invoice, (06) Remittance, (07) Billing, (08) Delivery, (09) External

Shipment Carrier Role: (1) Sender, (2) Receiver, (3) Third Party

Carton Status: (1) New, (2) In Progress, (3) Completed, (4) Shipped, (5) Canceled

Carton Restriction Level: (1) Department, (2) Class, (3) Subclass, (4) None

Example CSV File

For a store-based transaction import, the file name must have the fileNum, IDL-TRANSFERSHIP-<storeId>-<fileNum>.csv

Example:

IDL-TRANSFERSHIP-1111-1.csv

H,100000,5000,4,8000,4,2022-10-24 16:12:32,AUTCODE1,TSFID1,4,1,1,1,100,KG,2022-10-25 00:12:32,1,ALTDESTADDRESS,1,3RDPARTYNAME,3RDPARTYADDRESS,MOTIVE,TAXID1, FDOC1,FDOCURL1,1500,2022-10-24 16:12:32,1500,2022-10-24 16:12:32,1500,2022-10-24 16:12:32,15000,2022-10-24 16:12:32

C,100000,5,2,1,1,100,KG,1234,Y,4,1500,2022-10-23 11:32:12,15000,2022-10-24 16:12:32,15000,2022-10-24 16:12:32

D,100000,5,100701234,1,1,100,1

Transfer Delivery File

- Transfer delivery files must contain information for a single store only.
- Each delivery must contain at least one container.
- Each container must container at least one item.
- Duplicate cartons are not allowed on the delivery.
- If the container is open, the in-transit quantity will be incremented for the items at the destination store for the remaining expected quantity.
- The status of the delivery will be calculated from the status of the containers.
- UINs are not loaded as part of this transfer delivery data seeding file upload.

Table 2-57 Transfer Delivery File Row Layout (H – Header)

Field Name	Description	Required	Type
ROW_TYPE	The type of row that is represented.	Yes	“H”
IMPORT_DELV_ID	An import identifier to tie the delivery to its cartons and items.	Yes	VARCHAR2(128)
STORE_ID	The receiving store identifier.	Yes	NUMBER(10)
SOURCE_TYPE	The source type.	Yes	NUMBER(4)
SOURCE_ID	The source location identifier.	Yes	NUMBER(10)
ASN_ID	The Advance Shipment Notification number.	Yes	VARCHAR2(30)
RECEIPT_NO	The receipt number.	Yes	VARCHAR2(30)
CARRIER_ENTITY	The carrier entity.	No	VARCHAR2(128)
CARRIER_TYPE	The carrier type.	No	NUMBER(2)
CARRIER_CODE	Unique code identifier for a carrier.	No	VARCHAR2(4)
SOURCE_ADDRESS	The address of source sending delivery.	No	VARCHAR2(1000)
LICENSE_PLATE	A license plate number.	No	VARCHAR2(128)
FREIGHT_ID	The freight identifier.	No	VARCHAR2(128)

Table 2-57 (Cont.) Transfer Delivery File Row Layout (H – Header)

Field Name	Description	Required	Type
BOL_EXTERNAL_ID	Delivery Bill Of Lading from external system or entered by SIOCS user.	No	VARCHAR2(128)
FISCAL_DOCUMENT_ID	Fiscal Document Number.	No	VARCHAR2(128)
EXPECTED_DATE	The expected date of the Transfer Delivery.	No	DATE
RECEIVED_DATE	The received date of the Transfer Delivery.	No	DATE
RECEIVED_USER	The user who received the Transfer Delivery.	No	VARCHAR2(128)
CREATE_DATE	The create date of the Transfer Delivery.	Yes	DATE
CREATE_USER	The user who created the Transfer Delivery.	No	VARCHAR2(128)
UPDATE_DATE	The date when the Transfer Delivery was updated.	No	DATE
UPDATE_USER	The user who last updated the Transfer Delivery.	No	VARCHAR2(128)

Table 2-58 Transfer Delivery File Row Layout (C – Carton)

Field Name	Description	Required	Type
ROW_TYPE	The type of row that is represented.	Yes	“C”
IMPORT_DELV_ID	The identifier of the legacy information when the record was imported.	Yes	VARCHAR2(128)
IMPORT_CARTON_ID	The unique identifier for the transfer delivery carton/container.	Yes	VARCHAR2(128)
EXTERNAL_ID	An external carton identifier, often used to communicate with external systems.	No	VARCHAR2(128)
REFERENCE_ID	A reference identifier.	No	VARCHAR2(128)
STATUS	The status of the transfer delivery carton.	Yes	NUMBER(4)
SERIAL_CODE	A serial code.	No	NUMBER(18)
TRACKING_NUMBER	A tracking number for the container.	No	VARCHAR2(128)
DAMAGED_REASON	The reason for container damage.	No	VARCHAR2(128)

Table 2-58 (Cont.) Transfer Delivery File Row Layout (C – Carton)

Field Name	Description	Required	Type
DAMAGE_REMAINING	Y indicates all remaining quantities should be damaged on final receipt. Y/N value.	Yes	VARCHAR2(1)
RECEIVE_AT_SHOP_FLOOR	Indicates if the stock would be received at shop-floor or not."Y" if stock is to be received at shop-floor "N" otherwise.	Yes	VARCHAR2(1)
QUALITY_CONTROL	A quality control indicator. Y indicates that the carton must be manually received. Y/N value.	Yes	VARCHAR2(1)
EXTERNAL_CREATE	Indicates it was external created. Y indicates it was. Y/N value.	Yes	VARCHAR2(1)
ADJUSTED	Y indicates the container has been adjusted after receipt. Y/N value.	Yes	VARCHAR2(1)
COPIED	Y means has been copied as a misdirected container, N means it has not.	Yes	VARCHAR2(1)
RECEIVE_DATE	The date when the carton was received.	No	DATE
RECEIVE_USER	The user who received the carton.	No	VARCHAR2(128)
CREATE_DATE	The date when the carton was created.	Yes	DATE
CREATE_USER	The user who created the carton.	No	VARCHAR2(128)
UPDATE_DATE	The date when the carton was updated.	No	DATE
UPDATE_USER	The user who last updated the carton.	No	VARCHAR2(128)

Table 2-59 Transfer Delivery File Row Layout (D – Detail)

Field Name	Description	Required	Type
ROW_TYPE	The type of row that is represented.	Yes	"D"
IMPORT_DELV_ID	An import identifier to tie the delivery to its cartons and items.	Yes	VARCHAR(128)
IMPORT_CARTON_ID	Import identifier to tie the cartons to its items.	Yes	VARCHAR(128)

Table 2-59 (Cont.) Transfer Delivery File Row Layout (D – Detail)

Field Name	Description	Required	Type
ITEM_ID	The item identifier.	Yes	VARCHAR(25)
DOCUMENT_TYPE	Transfer delivery document type.	Yes	NUMBER(2)
DOCUMENT_DATE	The date when document was created.	Yes	DATE
IMPORT_ALLOC_ID	The original legacy import identifier from the allocation file when it was uploaded.	No	VARCHAR(128)
IMPORT_TSF_ID	The original legacy import identifier from the transfer file when it was uploaded.	No	VARCHAR(128)
CUST_ORD_EXTERNAL_ID	Customer order external identifier.	No	VARCHAR(128)
FUL_ORD_EXTERNAL_ID	Fulfillment order external identifier.	No	VARCHAR(128)
USE_AVAILABLE	Value of Y indicates item will be received as available inventory, N means received as unavailable inventory.	Yes	VARCHAR(1)
CASE_SIZE	Pack size of the item.	Yes	NUMBER(10,2)
QUANTITY_EXPECTED	The total number of units expected in this transfer delivery.	No	NUMBER(20,4)
QUANTITY_RECEIVED	The total number of units received in this transfer delivery.	No	NUMBER(20,4)
QUANTITY_DAMAGED	The total number of units that were damaged when the transfer delivery was received.	No	NUMBER(20,4)
PREVIOUS_RECEIVED	Units previous received when container is reopened for adjustment.	No	NUMBER(20,4)
PREVIOUS_DAMAGED	Units previous received as damaged when container is reopened for adjustment.	No	NUMBER(20,4)

Data Definition

Delivery Source Type: (1) Store, (3) Warehouse, (4) Finisher

Delivery Carrier Type: (0) Corporate, (1) Third Party

Carton Status: (1) New, (2) In Progress, (3) Damaged, (4) Received, (5) Missing

Detail Document Type: (0) Transfer, (1) Allocation

Example CSV File

For a store-based transaction import, the file name must have the fileNum, IDL-TRANSFERDELV-<storeId>-<fileNum>.csv

Example:

IDL-TRANSFERDELV-1111-1.csv

H,TSF-DELV-x6,5000,1,5001,ASN-1,Receipt-1,CARRENTITY1,0,CCD1,SRCADDR,LP1,FR1,BOLEXTID1,FDOC1,2022-12-08 09:00:00,2022-12-08 09:00:00,1500,2022-12-07 09:00:00,1500,2022-12-07 09:00:00,1500

C,TSF-DELV-x6,CARTON-3,EXTID1,REFID1,4,0599123645,TKNUM1,NO DAMAGE,N,Y,Y,Y,N,N,2022-12-07 09:00:00,1500,2022-12-07 09:00:00,1500,2022-12-07 09:00:00,1500

D,TSF-DELV-x6,CARTON-3,100637121,0,2022-12-07 09:00:00,IMPALLOCID1,778,CUSTORDID1,FCUSTORDID1,Y,1,1,1,0,0,0

UIN File

- This file must contain only information for a single store.
- Only the following statues will be allowed for the UIN: In Stock, Sold, Shipped To Warehouse, Shipped To vendor, Shipped to Finisher, Remove From Inventory, Missing, and Customer Fulfilled.
- The current functional area will be defaulted to MANUAL.
- The current functional identifier is not allowed on the import.
- Note that a UIN history record will be created for each imported UIN.
- Note that this loads UINs into the base UIN table and does not associate or attach any UINs to ongoing transactions.

Table 2-60 UIN File Row Layout (H – Header)

Field Name	Description	Required	Type
ITEM_ID	The identifier of the item.	Yes	VARCHAR2(25)
UIN	The universal identification number.	Yes	VARCHAR2(128)
STORE_ID	The store identifier.	Yes	NUMBER(10)
STATUS	The current status of the UIN.	Yes	NUMBER(2)
PREVIOUS_STATUS	The previous status of the UIN.	No	NUMBER(2)
PREVIOUS_FUNCTIONAL_AREA	The previous business area that contained the UIN for that previous status.	No	NUMBER(2)

Table 2-60 (Cont.) UIN File Row Layout (H – Header)

Field Name	Description	Required	Type
PREVIOUS_FUNCTIONAL_ID	The transaction id of the transaction that previously contained the UIN for that previous status.	No	VARCHAR2(128)
PREVIOUS_STORE_ID	The previous store identifier associated with the previous status.	No	NUMBER(10)
PREVIOUS_NONSELLABLE_TYPE_ID	A non-sellable inventory bucket the UIN was last within for that previous status.	No	NUMBER(12)
PREVIOUS_CARTON_ID	The identifier of the carton that previously contained the UIN for that previous status.	No	VARCHAR2(128)
CREATE_DATE	The date the UIN was first inserted into the system.	Yes	DATE
UPDATE_DATE	The last date the UIN was updated.	No	DATE
CREATE_USER	The user that first inserted the UIN into the system.	No	VARCHAR2(128)
UPDATE_USER	The user that last updated the UIN in the system.	No	VARCHAR2(128)

Data Definition

Status: (0) In Stock, (1) Sold, (2) Shipped To Warehouse, (5) Shipped To Vendor, ((6) Remove From Inventory, (8) Missing, (11) Customer Fulfilled, (12) Shipped to Finisher,

Functional Area: (0) Warehouse Delivery Receipt, (1) Direct Delivery Receipt, (2) Create Transfer, (3) Dispatch Transfer, (4) Receive Transfer, (5) Receipt Adjustment, (6) Crete Return, (7) Dispatch Return, (8) Inventory Adjustment, (9), Stock Count, (10) Stock Recount, (11) Stock Count Authorization, (12) Manual, (13) POS Sale, (14) POS return, (15) POS Sales Void, (16) POS Return Void, (17) UIN Web Service, (18) Customer Order, (20), Direct Delivery ASN, (21) Transfer ASN, (22) Transfer Shipment

Example CSV File

For a store-based transaction import, the file name must have the fileNum, IDL-ITEMUIN-<storeId>-<fileNum>.csv

Example:

IDL-ITEMUIN-1111-1.csv

100665085,testuinCsv0,5000,0,0,1,prev function area id,5000,5001,0,2022-10-24 14:23:00,2022-10-24 14:23:01,15000,15000

Vendor Return

- Vendor returns only allow one store per file.

- Must have at least one detail row per vendor return.
- For a Vendor Return in Approved or In Shipping status, the RTV Reserved bucket of the item's inventory at the source store will be updated with the approved quantity of the vendor return.
- External Locked attribute will be calculated. If External Id has a value and the status is Approved, In Shipping, or Completed, then External Locked = Y, otherwise.

Table 2-61 Vendor Return File Row Layout (H - Header)

Field Name	Description	Required	Type
ROW_TYPE	Defines the type of row content.	Yes	“H”
IMPORT_RTV_ID	An import identifier from external system to tie the return to its item detail within the file	Yes	NUMBER(10)
STORE_ID	The identifier of the store shipping the goods	Yes	NUMBER(10)
SUPPLIER_ID	The identifier of the supplier receiving the goods	Yes	VARCHAR(128)
EXTERNAL_ID_STATUS	An identifier to communicate to external systems when publishing information about this return	No	NUMBER(2)
NOT_AFTER_DATE	The date after which the return is no longer allowed	YES	DATE
AUTHORIZATION_NUMBER	The supplier authorization number	NO	VARCHAR(12)
ORIGIN_TYPE	The origin type of the return	YES	NUMBER(2)
ADDRESS_LINE_1	The first line of the return address	NO	VARCHAR(240)
ADDRESS_LINE_2	The second line of the return address	NO	VARCHAR(240)
ADDRESS_LINE_3	The third line of the return address	NO	VARCHAR(240)
ADDRESS_CITY	The city of the return address	NO	VARCHAR(120)
ADDRESS_COUNTRY	The country of the return address	NO	VARCHAR(3)
ADDRESS_POSTAL_CODE	The postal code of the return address	NO	VARCHAR(30)
APPROVED_USER	The user who approved the return	NO	VARCHAR(128)
APPROVED_DATE	The date the return was approved	NO	DATE
CLOSED_USER	The user who closed the return	NO	VARCHAR(128)
CLOSED_DATE	The date the return was closed	NO	DATE
CREATE_USER	The user who created the return	NO	VARCHAR(128)
CREATE_DATE	The date the return was created	YES	DATE
UPDATE_USER	The user who last updated the return	NO	VARCHAR(128)

Table 2-61 (Cont.) Vendor Return File Row Layout (H - Header)

UPDATE_DATE	The date the return was last updated	NO	DATE

Table 2-62 Vendor Return File Layout (D — Detail)

Field Name	Description	Required	Type
ROW_TYPE	Defines the type of row	YES	“D”
IMPORT_RTV_ID	An import identifier from external system to tie the return to its item detail within the file	YES	VARCHAR(128)
ITEM_ID	The unique identifier of the item/sku	YES	VARCHAR(25)
CASE_SIZE	The case size of this item on this return	NO	NUMER(10,2)
EXTERNAL_ID	An external identifier to this particular line item on the return	NO	NUMBER(15)
SHIPMENT_REASON_ID	A unique identifier to a reason code associated to this line item	YES	NUMBER(15)
QUANTITY_REQUESTED	The amount requested to return	NO	NUMBER(20,4)
QUANTITY_APPROVED	The amount approved to return	NO	NUMBER(20,4)
QUANTITY_SHIPPING	The amount prepared to ship on the return	NO	NUMBER(20,4)
QUANTITY_SHIPPED	The amount shipped on the return	NO	NUMBER(20,4)

Example CSV File

IDL-RTV-5000.csv

H,1298,5000,6100,800,6,2023-04-07 00:00:00,1276,2,,,,,,DEV,2023-04-02 00:00:00,DEV,2023-03-28 00:00:00,,2023-03-21 00:00:00,,

D,1298,6100,100000024,1,7,2,2,,0,2

Data Definition

Valid Return Status Quantity:

(1) Requested, (2) Requested In Progress, (3) RTV In Progress, (4) Approved, (5) In Shipping, (6) Completed, (7) Rejected, (8) Cancel Request, (9) Cancel RTV.

Vendor Origin Type:

(1) External, (2) Internal, (3) Shipment.

Vendor Shipment

- Vendor shipments only allow one store per file.

- Must have at least one carton row per header row.
- Must have at least one detail row per carton row.
- A carton in New status may have no items in it.
- Shipment status will be calculated from the container status.
- If any of the containers are in New, In Progress, or Completed status, the shipment status is In Progress.
- If all of the containers are in Canceled status, the shipment status is Canceled.
- If at least one container is Shipped and all other containers are Shipped or Canceled, the shipment status should be Shipped.
- If none of these conditions are met, the shipment should fail with status error

Prerequisite

Vendor returns must be loaded prior to vendor shipments that reference them.

Table 2-63 Vendor Shipment File Row Layout (H — Header)

Field Name	Description	Required	Type
ROW_TYPE	Defines the type of row content.	Yes	"H"
IMPORT_SHIP_ID	The import identifier from external system to tie the return to its item detail within the file.	Yes	VARCHAR(128)
STORE_ID	The identifier of the store shipping the return.	Yes	NUMBER(10)
SUPPLIER_ID	The identifier of the supplier receiving the return.	Yes	NUMBER(10)
IMPORT_RTV_ID	The import vendor return identifier of the previous uploaded return document file.	Yes	VARCHAR(128)
STATUS	The status of the shipment	Yes	NUMBER(2)
NOT_AFTER_DATE	A date after which the shipment should not be shipped.	No	DATE
AUTHORIZATION_CODE	A vendor authorization code.	No	VARCHAR(12)
TRACKING_NUMBER	Tracking number of the shipment.	No	VARCHAR(128)
SHIP_CARRIER_ID	Identifier of the carrier of the shipment.	No	NUMBER(10
SHIP_CARRIER_SERVICE_ID	Identifier of the carrier service of the shipment.	No	NUMBER(10
SHIPMENT_CARTON_DIM_ID	The shipment carton dimension identifier.	No	NUMBER(12)
SHIP_WEIGHT	The weight of the carton	No	NUMBER(12,3)
SHIP_WEIGHT_UOM	The unit of measure of the carton.	No	VARCHAR(4)
REQUESTED_PICKUP_DATE	The date requested for pickup.	No	DATE
SHIP_TO_ADDRESS_TYPE	The address type of the shipment.	No	VARCHAR(2)

Table 2-63 (Cont.) Vendor Shipment File Row Layout (H — Header)

ALT_DESTINATION_ADDRESS	An alternate destination address.	No	VARCHAR(2000)
CARRIER_ROLE	The carrier type of the shipment.	No	NUMBER(2)
THIRD_PARTY_NAME	The name of the a third party shipper.	No	VARCHAR(240)
THIRD_PARTY_ADDRESS	The address of a third party shipper.	No	VARCHAR(240)
MOTIVE	A motive for the shipment.	No	VARCHAR(120)
TAX_ID	A tax identifier.	No	VARCHAR(18)
CONTEXT_ID	An identifier of a context associated to the return.	No	NUMBER(18)
CONTEXT_VALUE	A value that goes with the context.	No	VARCHAR(25)
FISCAL_DOCUMENT_ID	The identifying number of a fiscal document associated to the return.	No	VARCHAR(255)
FISCAL_DOCUMENT_URL	A URL to the fiscal document.	No	VARCHAR(255)
SUBMIT_USER	The user that submitted the shipment.	No	VARCHAR(128)
SUBMIT_DATE	The date the shipment was submitted.	No	DATE
DISPATCH_USER	The user that dispatched the shipment.	No	VARCHAR(128)
DISPATCH_DATE	The date the shipment was dispatched.	No	DATE
CREATE_USER	The user that created the shipment.	No	VARCHAR(128)
CREATE_DATE	The date the shipment was created.	Yes	DATE
UPDATE_USER	The user that last updated the shipment.	No	VARCHAR(128)
UPDATE_DATE	The date the shipment was last updated.	No	DATE

Table 2-64 Vendor Shipment File Row Layout (C – Carton)

Field Name	Description	Required	Type
ROW_TYPE	Defines the type of row content.	Yes	“C”
IMPORT_SHIP_ID	An import identifier from external system to tie the shipment to its carton and items within the file.	Yes	VARCHAR(128)
IMPORT_CARTON_ID	Import identifier from external system to tie the carton to its items.	Yes	VARCHAR(128)
EXTERNAL_CARTON_ID	An external identifier associated to the carton.	No	VARCHAR(128)
STATUS	The status of the carton.	Yes	NUMBER(2)

Table 2-64 (Cont.) Vendor Shipment File Row Layout (C – Carton)

SHIPMENT_CARTON_DIM_ID	An identifier of the shipment carton dimension.	No	NUMBER(10)
WEIGHT	The weight of the carton.	No	NUMBER(12,4)
WEIGHT_UOM	The unit of measure of the weight of the carton.	No	VARCHAR(4)
TRACKING_NUMBER	A tracking number associated to the carton.	No	VARCHAR(128)
RESTRICTION_LEVEL	A restriction level associated to the carton.	Yes	NUMBER(4)
APPROVAL_USER	The user that approved the carton.	No	VARCHAR(128)
APPROVAL_DATE	The date the carton was approved.	No	DATE
CREATE_USER	The user that created the carton.	No	VARCHAR(128)
CREATE_DATE	The date the carton was created.	No	DATE
UPDATE_USER	The user that last updated the carton.	No	VARCHAR(128)
UPDATE_DATE	The date the carton was last updated.	No	DATE

Table 2-65 Vendor Shipment File Row Layout (D — Detail)

Field Name	Description	Required	Type
ROW_TYPE	Defines the type of row content.	Yes	“D”
IMPORT_SHIP_ID	An import identifier from external system to tie the shipment to its carton and items within the file.	Yes	VARCHAR(128)
IMPORT_CARTON_ID	An import identifier from external system to tie the carton to its items.	Yes	VARCHAR(128)
ITEM_ID	The identifier of the item.	Yes	VARCHAR(25)
SHIPMENT_REASON_ID	The identifier of a return reason associated to the item being returned.	Yes	NUMBER(15)
CASE_SIZE	The case size of this item on this return.	No	NUMBER(10,2)
QUANTITY	The quantity that was shipped.	Yes	NUMBER(20,4)

Example CSV File

IDL-RTVSHIP-5000.csv

H,2276,5000,6100,1276,4,,,,,,,,,,,3,,,,,,,,DEV,2023-04-01 00:00:00,DEV,2023-04-01 00:00:00,,2023-03-20 00:00:00,,

C,2276,1076,,4,,,,,4,DEV,2023-03-28 00:00:00,,2023-03-19 00:00:00,,

D,2276,1076,100000024,7,1,2

Data Definition

Valid Shipment Status Description: (1) New, (2) In Progress, (4) Shipped, (5) Canceled.

Vendor Shipment Carrier Role: (1) Sender, (2) Receiver, (3) Third Party.

Vendor Shipment Ship To Address Type: (01) Business, (02) Postal, (03) Returns, (04) Order, (05) Invoice, (06) Remittance, (07) Billing, (08) Delivery, (09) External.

Vendor Shipment Carton Status: (1) New, (2) In Progress, (3) Completed, (4) Shipped, (5) Canceled.

Vendor Shipment Carton Restriction Level: (1) Department, (2) Class, (3) Subclass, (4) None.

Vendor Delivery UIN

- Vendor delivery UIN will be loaded one store per file.
- DSD_CARTON needs to have IMPORT_ID added to it.
- VendorDeliveryImportDcsConsumer needs to capture the carton import identifier in the DSD_CARTON table.
- If a UIN does not exist at the store, create the UIN at the store in an In Stock status.
- Upon processing, the UIN itself will be updated with the information from the shipment.
- If the count of UINs is different than the line quantities counts, this will be an error and the transaction will be rejected.

Prerequisite

- Purchase orders must be loaded prior to DSD/Vendor deliveries that reference them.
- Vendor deliveries must be loaded prior to UINs that reference them.
- Optionally, item UINs may be loaded prior ot the UINs that reference them.

Table 2-66 Vendor Delivery File Row Layout

Field Name	Description	Required	Type
STORE_ID	The unique store identifier.	Yes	NUMBER (10, 0)
IMPORT_DELIVERY_ID	The import identifier of the delivery from the original DSD file upload.	Yes	VARCHAR(128)
IMPORT_CARTON_ID	The import identifier of the carton from the original DSD file upload.	Yes	VARCHAR(128)
ITEM_ID	The identifier of the item.	Yes	VARCHAR(25)
UIN	The UIN associated to the item.	Yes	VARCHAR(128)
SHIPPED	Y/N Indicator. Y indicates the UIN was shipped and is ready to be received.	Yes	VARCHAR(1)
RECEIVED	Y/N Indicator. Y indicates the UIN was received.	Yes	VARCHAR(1)

Table 2-66 (Cont.) Vendor Delivery File Row Layout

DAMAGED	Y/N Indicator. Y indicates the UIN was received as damaged. If a UIN is marked damaged ("Y"), then the carton status cannot be in "Received" status and an error should prevent this delivery from uploading.	Yes	VARCHAR(1)

Example CSV File

IDL-VENDORDELIVERYITEMUIN-5000.csv

5000,30000,30000,12345678901233,testuin2,Y,N,N

Data Definition

Valid Status For Pre-Existing UIN: (0) In Stock, (1) Sold, (2) Shipped To Warehouse, , (5) Shipped To Vendor, (12) Shipped To Finisher, (6) Removed From Inventory, (8) Missing, (11) Customer Fulfilled

Vendor Shipment UIN

- Vendor Shipment UIN will be loaded one store per file.
- RTV_SHIPMENT_CARTON needs to have IMPORT_ID added to it.
- VendorShipmentImportDcsConsumer needs to capture the carton import identifier in the RTV_SHIPMENT_CARTON table.
- Upon processing, the UIN itself will be updated with the information from the shipment.
- If the count of UINs is different than the line quantities counts, this will be an error and the transaction will be rejected.

Prerequisite

- Vendor returns must be loaded prior to vendor shipments that reference them.
- Vendor shipments must be loaded prior to the vendor shipment UINs that reference them.
- Item UINs must be loaded prior to the vendor shipment UIns that reference them.

Table 2-67 Vendor Shipment UIN File Row Layout

Field Name	Description	Required	Type
STORE_ID	The unique store identifier.	Yes	NUMBER (15)
IMPORT_SHIPMENT_ID	The import identifier of the shipment from the original vendor return shipment file upload.	Yes	VARCHAR(128)
IMPORT_CARTON_ID	The import identifier of the carton from the original DSD file upload.	Yes	VARCHAR(128)
ITEM_ID	The identifier of the item.	Yes	VARCHAR(25
UIN	The UIN associated to the item.	Yes	VARCHAR(128)

Example CSV File

IDL-VENDORDELIVERYUIN-5000.csv

5000,30000,30000,12345678901233,testuin2

Data Definition

Valid Status For Pre-Existing UIN: (0) In Stock, (1) Sold, (2) Shipped To Warehouse, , (5) Shipped To Vendor, (12) Shipped To Finisher, (6) Removed From Inventory, (8) Missing, (11) Customer Fulfilled

Transfer Shipment UIN

- Transfer Shipment UIN will be loaded one store per file.
- TSF_SHIPMENT_CARTON needs to have IMPORT_ID added to it.
- TransferShipmentImportDcsConsumer needs to capture the carton import identifier in the TSF_SHIPMENT_CARTON table.
- Upon processing, the UIN itself will be updated with the information from the shipment.
- If the count of UINs is different than the line quantities counts, this will be an error and the transaction will be rejected.

Prerequisite

- Transfers must be loaded prior to transfer shipments that reference them.
- Transfer shipments must be loaded prior to transfer shipment UINs that reference them.
- Item UINS must be loaded prior to the transfer shipment UINs that reference them.

Table 2-68 Transfer Shipment UIN File Row Layout

Field Name	Description	Required	Type
STORE_ID	The unique store identifier.	Yes	NUMBER (10, 0)
IMPORT_SHIPMENT_ID	The import identifier of the shipment from the original shipment file upload.	Yes	VARCHAR(128)
IMPORT_CARTON_ID	The import identifier of the carton from the original shipment file upload.	Yes	VARCHAR(128)
ITEM_ID	The identifier of the item.	Yes	VARCHAR(25)
UIN	The identifier of the item. The UIN associated to the item.	Yes	VARCHAR(128)

Example CSV File

IDL-TRANSFERSHIPMENTUIN-5000.csv

5000,10001,10001,12345678901233,testuin2

Data Definition

Valid Status For Pre-Existing UIN: (0) In Stock, (1) Sold, (2) Shipped To Warehouse, (5) Shipped To Vendor, (12) Shipped To Finisher, (6) Removed From Inventory, (8) Missing, (11) Customer Fulfilled

Transfer Delivery UIN

- Transfer delivery UIN will be loaded one store per file.
- TSf_DELV_CARTON needs to have IMPORT_ID added to it.
- TransferDeliveryImportDcsConsumer needs to capture the carton import identifier in the TSF_DELV_CARTON table.
- Transfer Shipment UINs must be loaded prior to transfer delivery UINs.
- The received quantity of the transfer delivery line item should be set to the total of the received UINs for that line item.
- The damaged quantity of the transfer delivery line item should be set to the total of the damaged UINs for that line item
- Upon processing, the UIN itself will be updated with the information from the delivery.
- If the count of UINs is different than the line quantities counts, this will be an error and the transaction will be rejected.

Prerequisite

- Transfer must be loaded prior to the transfer delivery that references them.
- Allocations must be loaded prior to the transfer delivery that references them (optional if testing allocations).
- Transfer deliveries must be loaded prior to the transfer delivery UINS that references them.
- Optionally, Item UINs may be loaded prior to the transfer delivery UINs that reference them.

Table 2-69 Transfer Delivery Fields

Field Name	Description	Required	Type
STORE_ID	The unique store identifier.	Yes	NUMBER (10, 0)
IMPORT_DELIVERY_ID	The import identifier of the delivery from the original delivery file upload.	Yes	VARCHAR(128)
IMPORT_CARTON_ID	The import identifier of the carton from the original delivery file upload.	Yes	VARCHAR(128)
ITEM_ID	The identifier of the item.	Yes	VARCHAR(25
UIN	The UIN associated to the item.	Yes	VARCHAR(128)
SHIPPED	Y/N Indicator. Y indicates the UIN was shipped and is ready to be received.	Yes	VARCHAR(1)
RECEIVED	Y/N Indicator. Y indicates the UIN was received.	Yes	VARCHAR(1)
DAMAGED	Y/N Indicator. Y indicates the UIN was received as damaged. If a UIN is marked damaged ("Y"), then the carton status cannot be in "Received" status and an error should prevent this delivery from uploading.	Yes	VARCHAR(1)

Example CSV File

IDL-TRANSFERDELIVERYUIN-5000.csv

5000,20000,20000,12345678901233,testuin2,Y,N,N

Data Definition

Valid Status For Pre-Existing UIN: (0) In Stock, (1) Sold, (2) Shipped To Warehouse, (5) Shipped To Vendor, (12) Shipped To Finisher, (6) Removed From Inventory, (8) Missing, (11) Customer Fulfilled

Supported Locales

Table 2-70 Locale ID Values

LOCALE_ID	LOCALE_LANGUAGE	LOCALE_DESCRIPTION
1	en	English
2	de	German
3	fr	French
4	es	Spanish
5	ja	Japanese
6	ko	Korean
7	ru	Russian
8	zh	Chinese
9	tr	Turkish
10	hu	Hungarian
11	zh	Traditional Chinese
12	pt	Brazilian Portuguese
13	ar	Arabic
15	hr	Croatian
18	nl	Dutch
20	el	Greek
22	it	Italian
26	pl	Polish
31	sv	Swedish
32	sq	Albanian
33	hy	Armenian
34	az	Azerbaijani
35	be	Belarusian
36	bn	Bengali
37	bs	Bosnian
38	bg	Bulgarian
39	my	Burmese
40	cs	Czech
41	da	Danish

Table 2-70 (Cont.) Locale ID Values

LOCALE_ID	LOCALE_LANGUAGE	LOCALE_DESCRIPTION
42	et	Estonian
43	fil	Filipino
44	fi	Finnish
45	ka	Georgian
46	he	Hebrew
47	hi	Hindi
48	id	Indonesian
49	kk	Indonesian
50	km	Khmer
51	lo	Lao
52	lv	Latvian
53	lt	Lithuanian
54	ms	Malay
55	no	Norwegian
56	ro	Romanian
57	sr	Serbian
58	sk	Slovak
59	sl	Slovene
60	th	Thai
61	uk	Ukrainian
62	ur	Urdu
63	uz	Uzbek
64	vi	Vietnamese

3
Reporting

EICS can produce reports for retails to view.

Reports are generated from within the functional areas of EICS and includes information about shipping documentation, delivery reports, pick detail reports and so on. EICS uses a report screen to preview the report by sending the request parameters as report name and required parameter for the corresponding listed reports.

This section covers the following:

- Report URL Locations
- Previewing a Report
- EICS Operational Reports

Report URL Locations

The URL Location for each report type:

Table 3-1 Report URL Location

Type	URL Location
Customer Order Report	/BIP_SIOCS_REPORTS_FOLDER /CustomerOrder Report/ CustomerOrderReport.xdo
Customer Order Bin Label Report	/BIP_SIOCS_REPORTS_FOLDER /CustomerOrderB inLabelReport/ CustomerOrderBinLabelReport.xdo
Customer Order Delivery Report	/BIP_SIOCS_REPORTS_FOLDER /CustomerOrder DeliveryReport/ CustomerOrderDeliveryReport.xdo
Customer Order Delivery BOL Report	/BIP_SIOCS_REPORTS_FOLDER / CustomerOrderDeliveryBOLReport/ CustomerOrderDeliveryBOLReport.xdo
Customer Order Pick Report	/BIP_SIOCS_REPORTS_FOLDER /CustomerOrderP ickReport/ CustomerOrderPickReport.xdo
Customer Order Pick Discrepancy Report	/BIP_SIOCS_REPORTS_FOLDER /CustomerOrderP ickDiscrepancyReport/CustomerOrderPickDiscrepancyReport.xdo
Customer Order Reverse Pick Report	/BIP_SIOCS_REPORTS_FOLDER / CustomerOrderReversePickReport/ CustomerOrderReversePickReport.xdo
Direct Delivery Report	/BIP_SIOCS_REPORTS_FOLDER /DirectDeliveryR eport/ DirectDeliveryReport.xdo
Direct Delivery AGSN Report	/BIP_SIOCS_REPORTS_FOLDER /VendorDeliveryAGSNReport/ VendorDeliveryAGSNReport.xdo
Direct Delivery Discrepant Item Report	/BIP_SIOCS_REPORTS_FOLDER /DirectDeliveryDi screpantItemsReport/DirectDeliveryDiscrepantItemsRe port.xdo
Direct Delivery Label Report	/BIP_SIOCS_REPORTS_FOLDER /VendorDeliveryLabel/ VendorDeliveryLabel.xdo

Table 3-1 (Cont.) Report URL Location

Type	URL Location
Inventory Adjustment Report	/BIP_SIOCS_REPORTS_FOLDER /InventoryAdjust mentReport/ InventoryAdjustmentReport.xdo
InventoryAdjustmentAGSNRep ort	/BIP_SIOCS_REPORTS_FOLDER /InventoryAdjustmentAGSNReport/ InventoryAdjustmentAGSNReport.xdo
Item Basket Report	/BIP_SIOCS_REPORTS_FOLDER /ItemBasketReport/ ItemBasketReport.xdo
Item Detail Report	/BIP_SIOCS_REPORTS_FOLDER /ItemDetailRepor t/ ItemDetailReport.xdo
Purchase Order Report	/BIP_SIOCS_REPORTS_FOLDER /PurchaseOrderR eport/ PurchaseOrderReport.xdo
RFID History Report	/BIP_SIOCS_REPORTS_FOLDER /RFIDHistoryReport/ RFIDHistoryReport.xdo
RTV Report	/BIP_SIOCS_REPORTS_FOLDER /RTVReport/RTV Report.xdo
RTV Shipment Report	/BIP_SIOCS_REPORTS_FOLDER /VendorShipmentReport/ VendorShipmentReport.xdo
RTV Shipment BOL Report	/BIP_SIOCS_REPORTS_FOLDER /VendorShipmentBOLReport/ VendorShipmentBOLReport.xdo
RTV Shipment Container Report	/BIP_SIOCS_REPORTS_FOLDER /VendorShipmentCartonReport/ VendorShipmentCartonReport.xdo
RTV Shipping Label Report	/BIP_SIOCS_REPORTS_FOLDER /VendorShippingLabel/ VendorShippingLabel.xdo
Scan List Report	/BIP_SIOCS_REPORTS_FOLDER /ReplenishmentGapReport/ ReplenishmentGapReport.xdo
Shelf Adjustment Report	/BIP_SIOCS_REPORTS_FOLDER /ShelfAdjustment Report/ ShelfAdjustmentReport.xdo
Shelf Replenishment Report	/BIP_SIOCS_REPORTS_FOLDER /ShelfReplenishm entReport/ ShelfReplenishmentReport.xdo
Stock Count All Location Report	/BIP_SIOCS_REPORTS_FOLDER /StockCountAllLo cReport/ StockCountAllLocReport.xdo
Stock Count Report	/BIP_SIOCS_REPORTS_FOLDER /StockCountReport/ StockCountReport.xdo
Stock Count Export Report	/BIP_SIOCS_REPORTS_FOLDER /StockCountExpo rtReport/ StockCountExportReport.xdo
Stock Count Rejected Item Report	/BIP_SIOCS_REPORTS_FOLDER /StockCountRejec tedItemReport/ StockCountRejectedItemReport.xdo
Store Order Report	/BIP_SIOCS_REPORTS_FOLDER /StoreOrderReport/ StoreOrderReport.xdo
Transfer Report	/BIP_SIOCS_REPORTS_FOLDER /TransferReport/ TransferReport.xdo
Transfer Receiving Report	/BIP_SIOCS_REPORTS_FOLDER /TransferDelivery Report/ TransferDeliveryReport.xdo
Transfer Receiving AGSN Report	/BIP_SIOCS_REPORTS_FOLDER /TransferDeliveryAGSNReport/ TransferDeliveryAGSNReport.xdo
Transfer Receiving Exception Report	/BIP_SIOCS_REPORTS_FOLDER /TransferDelivery ExceptionReport/ TransferDeliveryExceptionReport.xdo

Table 3-1 (Cont.) Report URL Location

Type	URL Location
Transfer Receiving Label Report	/BIP_SIOCS_REPORTS_FOLDER /TransferDelivery Label/ TransferDeliveryLabel.xdo
Transfer Shipment Report	/BIP_SIOCS_REPORTS_FOLDER /TransferShipmen tReport/ TransferShipmentReport.xdo
Transfer Shipment BOL Report	/BIP_SIOCS_REPORTS_FOLDER /TransferShipmen tBolReport/ TransferShipmentBolReport.xdo
Transfer Shipment Container Report	/BIP_SIOCS_REPORTS_FOLDER /TransferShipmen tCartonReport/ TransferShipmentCartonReport.xdo
Transfer Shipping Label Report	/BIP_SIOCS_REPORTS_FOLDER /TransferShippin gLabel/ TransferShippingLabel.xdo

Note:

<BIP_SIOCS_REPORTS_FOLDER> is the folder where EICS reports have been uploaded on the BI Publisher server. If EICS reports are uploaded to the SIOCS folder, the folder is /SIOCS.

Security Considerations

Customer Administration User must create an IDCS user with the following BI groups assigned to access the report endpoints. TENANT_ID is the tenant ID of the DIS tenant on-boarded as part of the customer environment provisioning. The user credentials must then be configured on the *Credential Administration* screen. Refer to Chapter 6 - Technical Maintenance Screens / Credential Administration section for more details.

IDCS groups required

- <TENANT_ID>-BIConsumer
- <TENANT_ID>-BIContentAuthor

Example:

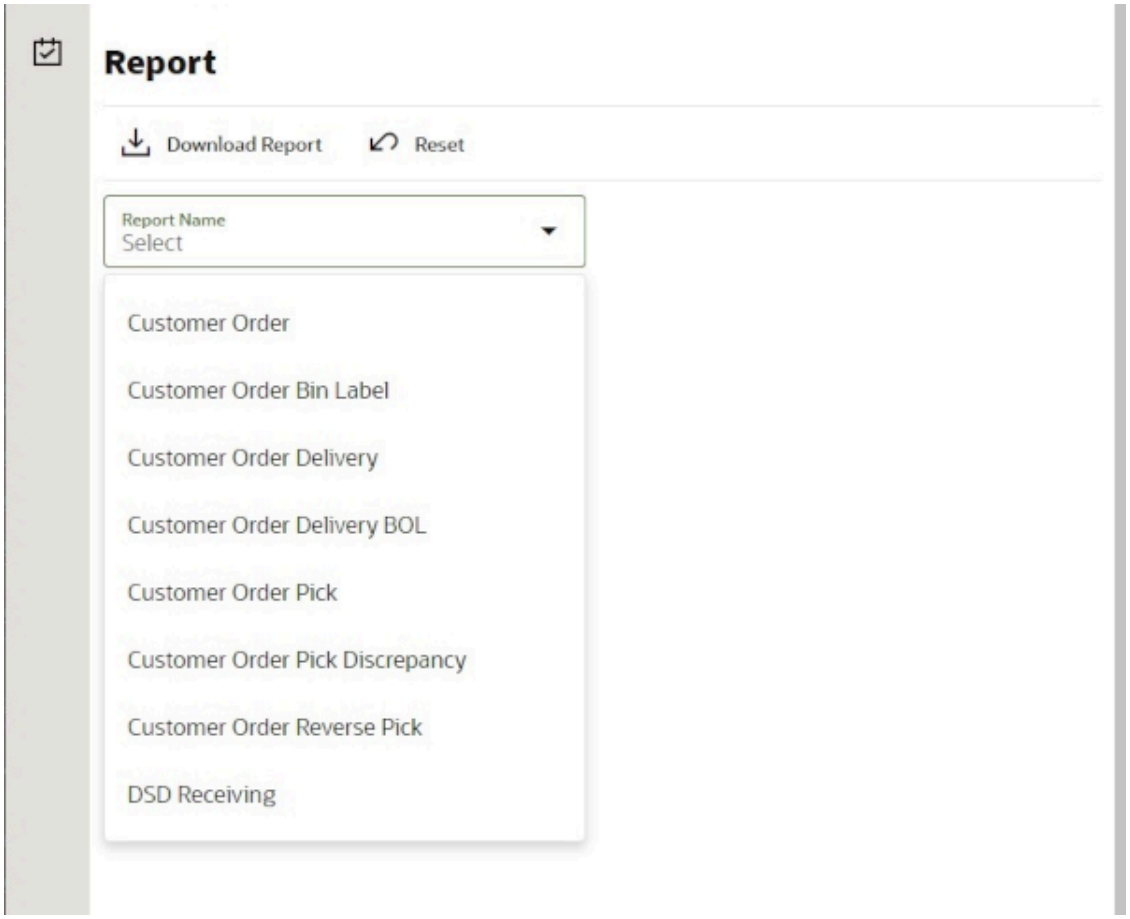
DIS URL: https://gbua.eu-xxxxxx-x.oci.oraclecloud.com/abcdefgh/xmlpserver

IDCS Groups: abcdefgh-BIConsumer and abcdefgh-BIContentAuthor

Previewing a Report

Users can download the report from EICS JET UI Home/Operations/Report.

Figure 3-1 Reports Screen



EICS Operational Reports

The following list shows the EICS Operational Reports.

Table 3-2 Operational Reports

Report Name	Report Parameters	Primary Views or Tables
Customer Order Bin Label Report	PICK_ID, COPIES	RPRT_FUL_ORD_BIN_V
Customer Order BOL Report	DELIVERY_ID, LOCALE_ID,STORE_TIMEZONE ,COPIES	RPRT_FUL_ORD_DLV_BOL_V
Customer Order Delivery Report	DELIVERY_ID, LOCALE_ID,STORE_TIMEZONE ,COPIES	RPRT_FUL_ORD_DLV_V
Customer Order Pick Discrepancy Report	PICK_ID, LOCALE_ID,STORE_TIMEZONE ,COPIES	RPRT_FUL_ORD_PICK_DISC_ V
Customer Order Pick Report	PICK_ID, LOCALE_ID,STORE_TIMEZONE ,COPIES	RPRT_FUL_ORD_PICK_V

Table 3-2 (Cont.) Operational Reports

Report Name	Report Parameters	Primary Views or Tables
Customer Order Report	ORDER_ID, LOCALE_ID,STORE_TIMEZONE ,COPIES	RPRT_FUL_ORD_V
Customer Order Reverse Pick Report	REVERSE_PICK_ID, LOCALE_ID,STORE_TIMEZONE ,COPIES	RPRT_FUL_ORD_RV_PICK_V
Direct Delivery AGSN Report	CARTON_ID,COPIES	DSD_LINE_ITEM_UIN, ITEM_UIN
Direct Delivery Discrepant Items Report	RECEIPT_ID, LOCALE_ID,STORE_TIMEZONE ,COPIES	RPRT_DSD_DISCREPANT_ITM_V, RPRT_DSD_V
Direct Delivery Label Report	CARTON_ID,LOCALE_ID	STORE,DSD,DSD_CARTON,DSD_LINE_ITEM,SUPPLIER,ADDRESS,ITEM
Direct Delivery Report	RECEIPT_ID, LOCALE_ID,STORE_TIMEZONE ,COPIES	RPRT_DSD_V, NOTES
Inventory Adjustment AGSN Report	INV_ADJUST_ID, COPIES	ITEM_UIN, INV_ADJUST_LINE_ITEM_UIN
Inventory Adjustment Report	INV_ADJUST_ID, LOCALE_ID,STORE_TIMEZONE ,COPIES	RPRT_INV_ADJUST_V, CONFIG_SYSTEM
Item Basket Report	ITEM_BASKET_ID,LOCALE_ID, STORE_TIMEZONE,COPIES	RPRT_ITEM_BASKET_V, NOTES
Item Detail Report	ITEMID,STOREID,LOCALE_ID,STORE_TIMEZONE,COPIES	STORE_SEQUENCE_ITEM,STORE_SEQUENCE_AREA,REPORT_TEMPLATE,TSF_ALLOCATION,ITEM,WAREHOUSE,RPRT_ITEM_DE TAIL_V
Purchase Order Report	PURCHASE_ORDER_ID,LOCALE_ID,STORE_TIMEZONE,COPIES	RPRT_PURCHASE_ORD_V ,RPRT_DSD_V
RFID History Report	ITEM_ID, FROM_DATE, TO_DATE, LOCALE_ID, COPIES	RPRT_RFID_HISTORY_V
RTV Report	RETURN_ID,LOCALE_ID,STORE_TIMEZONE,COPIES	RPRT_RTV_V
RTV Shipment BOL Report	SHIP_NUMBER,LOCALE_ID,STORE_TIMEZONE,COPIES	RPRT_RTV_SHIP_BOL_V, NOTES
RTV Shipment Container Report	CARTON_ID,LOCALE_ID,STORE_TIMEZONE,COPIES	RPRT_RTV_SHIP_V
RTV Shipment Report	SHIP_NUMBER,LOCALE_ID,STORE_TIMEZONE,COPIES	RPRT_RTV_SHIP_V
RTV Shipping Label Report	CARTON_ID,LOCALE_ID, COPIES	RPRT_RTV_SHIP_BOL_V,RTV_SHIP,CODE_DETAIL

Table 3-2 (Cont.) Operational Reports

Report Name	Report Parameters	Primary Views or Tables
Scan List Report	REPLENISH_GAP_ID,LOCALE_ID,STORE_TIMEZONE, COPIES	RPRT_REPLENISH_GAP_V,NOTES
Shelf Adjustment Report	SHELF_ADJUST_ID,LOCALE_ID,STORE_TIMEZONE,COPIES	RPRT_SHELF_ADJUST_V,NOTES
Shelf Replenishment Report	SHELF_REPLENISH_ID,LOCALE_ID,STORE_TIMEZONE,COPIES	RPRT_SHELF_REPLENISH_V,NOTES
Stock Count All Location Report	STORE_ID,STOCK_COUNT_ID, COPIES	RPRT_STOCK_COUNT_V
Stock Count Detail Report	STOCK_COUNT_ID,STOCK_COUNT_CHILD_ID, STORE_TIMEZONE,PHASE,COPIES,LOCALE_ID	RPRT_STOCK_COUNT_V,NOTES
Stock Count Export Report	STOCK_COUNT_ID,COPIES	STOCK_COUNT_LINE_ITEM,STOCK_COUNT,STOCK_COUNT_LINE_ITEM_ UIN
Stock Count Rejected Item Report	LOCALE_ID,COPIES,STOCK_COUNT_ID	RPRT_STOCK_COUNT_NOF_ V
Store Order Report	STORE_ORDER_ID, STORE_TIMEZONE, LOCALE_ID	RPRT_STORE_ORDER_V, STORE_ORDER, STORE_ORDER_CFA, STORE_ORDER_CDA, CUSTOM_ATT_ADMIN
Transfer Receiving AGSN Report	CARTON_ID,COPIES	TSF_DELV_LINE_ITEM_UIN,ITEM_UIN
Transfer Receiving Exception Report	DELIVERY_ID,LOCALE_ID,STORE_TIMEZONE,COPIES	RPRT_TSF_DELV_V
Transfer Receiving Label Report	CARTON_ID,LOCALE_ID	RPRT_TSF_DELV,TSF_DELV_CARTON,TSF_DELV_LINE_ITEM,STORE,ADDRESS,WAREHOUSE,PARTNER,TSF,CODE_DETAIL
Transfer Receiving Report	DELIVERY_ID,LOCALE_ID,STORE_TIMEZONE,COPIES	TSF_DELV,TSF_DELV_CARTON,TSF_DELV_LINE_ITEM,ITEM,STORE_ITEM_STOCK,STORE,WAREHOUSE,PARTNER,CONFIG_SYSTEM,NOTES
Transfer Report	TRANSFER_ID,LOCALE_ID,STORE_TIMEZONE,COPIES	RPRT_TRANSFER_V
Transfer Shipment BOL Report	SHIPMENT_ID,LOCALE_ID,STORE_TIMEZONE,COPIES	TSF_SHIP,TSF_SHIP_CARTON,TSF_SHIP_LINE_ITEM,ITEM,CONFIG_SYSTEM,SHIPMENT_BOL,SHIPMENT_CARTON_DIM,SHIPMENT_CARRIER_SERVICE, SHIPMENT_CARRIER, STORE,ADDRESS,NOTES

Table 3-2 (Cont.) Operational Reports

Report Name	Report Parameters	Primary Views or Tables
Transfer Shipment Container Report	CARTON_ID,LOCALE_ID,STORE_TIMEZONE,COPIES	TSF_SHIP,TSF_SHIP_CARTON, TSF_SHIP_LINE_ITEM,ITEM,STORE,WAREHOUSE,PARTNER,CONFIG_SYSTEM,SHIPMENT_REASON
Transfer Shipment Report	SHIPMENT_ID,LOCALE_ID,STORE_TIMEZONE,COPIES	RPRT_TSF_SHIP_V, NOTES
Transfer Shipping Label	CARTON_ID,LOCALE_ID	TSF,TSF_SHIP,TSF_SHIP_CARTON, TSF_SHIP_LINE_ITEM,ITEM,SHIPMENT_BOL,STORE,ADDRESS,CODE_DETAIL,PARTNER,WAREHOUSE

Using BI Publisher for Custom Reports

The ability to utilize Oracle Business Intelligence Publisher (BI Publisher) for custom reports is available as part of your EICS service subscription and is the only option available for creating custom reports against the live production database. Other reporting tools can be used, but must be based on other data sources, such as the replicated data in the Retail Data Store (RDS) or the Data Access Schema (DAS).

Accessing BI Publisher

In a SaaS implementation, you will access BI Publisher using a URL like this, where the hostname is replaced with that which is relevant for your implementation:

```
https://<hostname>/<tenantname>/xmlpserver
```

In order to create reports, you will need to ensure you have the below privileges assigned to your user ID through IDCS:

- <tenantId>-BIConsumer
- <tenantId>BIContentAuthor
- <tenantId>DVConsumer
- <tenantId>DVContentAuthor

Creating a BI Publisher Report

BI Publisher supports creating a number of different types of reports, including reports with charts, table-based report, and so on. For details on how to create reports in BI Publisher, see the Oracle Fusion Middleware Report Designer's Guide for Oracle Business Intelligence Publisher, especially Chapter 2 on creating and editing reports. As you build your reports consider the input parameters that are available based on where the report will be displayed. Details on the available parameters for the reports are available in section EICS Operational Reports.

> **Note:**
>
> Custom reports in a user's My Folder will not be backed up by Oracle but could be manually backed up by the user. Otherwise, all custom reports should be saved in the Shared Folders/Custom folder to ensure that they are included in the backup/ restoration processes.

Displaying a BI Publisher Report

Once you have created your report, you'll need to identify the URL for the report. The basic URL structure will be:

```
http://<hostname>/<tenantname>/xmlpserver/<ReportDirectory>/<ReportName>.xdo
```

- *hostname* and *tenantname* - will be the hostname and tenant ID for your Merchandising BI Publisher implementation
- *xmlpserver* - this is a static string
- *ReportDirectory* - folder path to the report
- *ReportName.xdo* - the filename you gave the report; if the name has spaces, then use a + between words

BI Publisher Reports Delivery Through Object Storage

Delivering scheduled reports through Object Storage

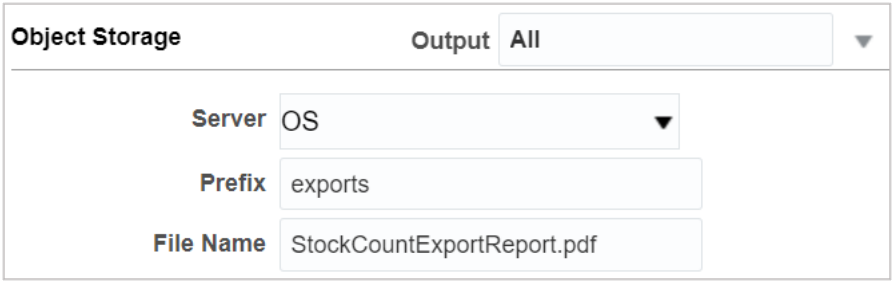
For details on how to set up reports delivery through object storage, refer to Set Output Options in Oracle Cloud Visualizing Data and Building Reports in Oracle Analytics Cloud.

While adding the destination for the report's delivery as Object Storage, you will need the following set of inputs that are required to push the file to object storage:

1. **Server** – The server is preconfigured as OS for any tenant. OS must always be selected.
2. **Prefix** – The prefix under the object storage bucket where the file will be uploaded
3. **File Name** – The file name with which the scheduled report output will be delivered to the object storage.

For example:

Figure 3-2 Object Storage



Downloading the BI Publisher reports from Object Storage

Once the reports are sent to object storage, use the createPar service to download the files. This service is available in Retail Home; it generates a PAR (Pre-authenticated Request) to download the file.

For more details on this, refer to the Retail Home documentation.

4
Internationalization

Internationalization is the process of creating software that can be translated easily. SIOCS has been internationalized to support multiple languages.

This section covers the following:

- Supported Locales
- SOCS Client Translations
- EICS Client Translations
- EICS Server Translations
- Translation Topics
- Translation Keys
- Translation Setup Screen
- Translation File Upload
- Report Translations

Supported Locales

SIOCS supports translation into following locales:

1. Arabic
2. Chinese (Simplified)
3. Chinese (Traditional)
4. Croatian
5. Dutch
6. English
7. French
8. German
9. Greek
10. Hungarian
11. Italian
12. Japanese
13. Korean
14. Polish
15. Portuguese (Brazilian)
16. Russian
17. Spanish

18. Swedish
19. Turkish

Apart from these, extension hooks are added for following new locales on EICS:

1. Albanian
2. Armenian
3. Azerbaijani
4. Belarusian
5. Bengali
6. Bosnian
7. Bulgarian
8. Burmese
9. Czech
10. Danish
11. Estonian
12. Filipino
13. Finnish
14. Georgian
15. Hebrew
16. Hindi
17. Indonesian
18. Kazakh
19. Khmer
20. Lao
21. Latvian
22. Lithuanian
23. Malay
24. Norwegian
25. Romanian
26. Serbian
27. Slovak
28. Slovene
29. Thai
30. Ukrainian
31. Urdu
32. Uzbek
33. Vietnamese

Translation records for these locales are defaulted to English. Translation value can be updated for these locales by accessing administration screen.

SOCS Client Translations

Translation of SOCS Graphical User Interface (GUI) and client-based display messages fall under this category.

SOCS client follows an XML format to organize translation records within a translation bundle. Each supported locale will have its on XLF file. For example, translations for French locale could be found under SimMobileViewControllerBundle_fr.xlf file. These XLF files are packaged with the rest of the application when the mobile application is built for deployment and any changes to them will require a new deployment of the mobile application.

EICS Client Translations

Translation of EICS Graphical User Interface (GUI) and client-based display messages fall under this category.

EICS Admin UI translations relies on following two bundle categories:

- Framework bundles: owned by JET/JRAF/LUX.
- EICS bundles: owned by EICS.

All these bundles are merged at runtime to provide an overall translation bundle which is used to provide translated UI content.

EICS owned translation are maintained in the EICS database. EICS client translation bundle is generated at runtime depending on the user locale and is applied on top of framework bundles by a custom plugin to provide an overall translation bundle for the client.

EICS Server Translations

Translation of server data, report data, notifications, server error messages, and other server-based message, fall under this category. These translation records are maintained in EICS database and are translated via a cached server translation provider.

Translation Topics

EICS translation records are grouped under translations topics for ease of management through the administration screens. Each translation key belonging to one of the translation topics below:

Table 4-1 Translation Topics

Translation Topic	Comments
Barcode	Captures translation keys for barcode processors.
Batch	Captures translation keys for batches.
Carrier	Captures translation keys for shipment carrier and carrier services.
Code Info	Captures translation keys for code type and code details.
Configuration	Captures translation keys for system, store and store default configuration parameters.
Custom Attributes	Captures translation keys related to custom attributes.

Table 4-1 (Cont.) Translation Topics

Translation Topic	Comments
Data	Captures translation keys related to system data like status, types and so on.
Date Import	Captures translation keys related to data import.
Delivery Timeslot	Captures translation keys related to delivery timestamps.
Inventory Adjustment Reason	Captures translation keys for inventory adjustment reason codes.
Isn Type	Captures translation keys related to ISN types.
Message	Captures translation keys related to server messages which are mostly error messages.
Non Sellable Type	Captures translation keys for non-sellable types.
Notification	Captures notification related translation keys
Reports	Captures translation keys related to reports.
Retail Home	Captures translation keys related to retail home tile reports.
Security	Captures translation keys for security permissions, groups and roles.
Shipment Reason	Captures translation keys for shipment reason codes.
UI	Captures translation keys related to user interface.

Translation Keys

EICS translation keys follow a dotted naming convention (for example, functional.area.key). The key name also identifies the functional area it belongs to which makes it easy to locate on the Translation Setup EICS Admin Client screen. Each key has a corresponding translation for each language. These translations can be modified using the administration screen.

Translation Setup Screen

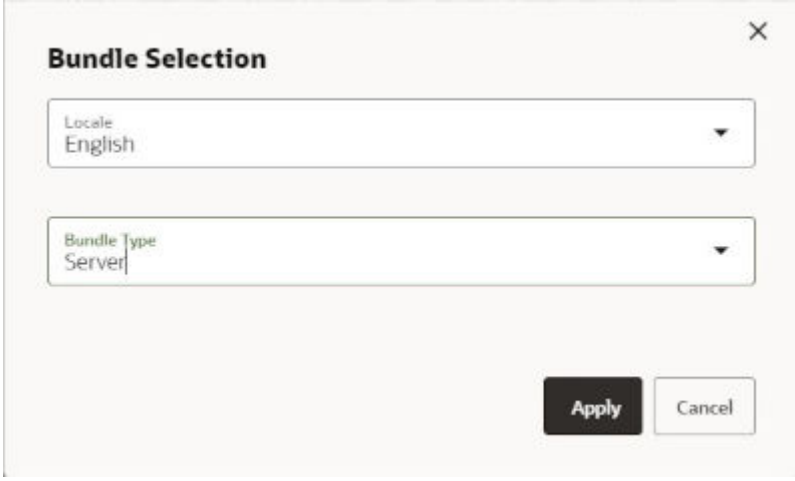
This EICS admin client screen can be used to look up and edit EICS client and server translation records for a translation locale supported by the system. This screen does not support addition and deletion of translation records.

Bundle Selection Dialog

Bundle selection dialog automatically popups up when you navigate to the Translation Setup Screen.

Select a bundle here and click **Apply** to load translation records.

Figure 4-1 Bundle Selection Dialog



- **Locale Selection**: This drop-down will list all the translation locales supported by EICS server.
- **Bundle Type Selection**: This drop-down will list the translation bundles available for customization.

The screen supports following two bundle types:

- **Server**: This bundle identifies EICS Server translation records.
- **Operations UI**: This bundle identifies EICS Client translation records.

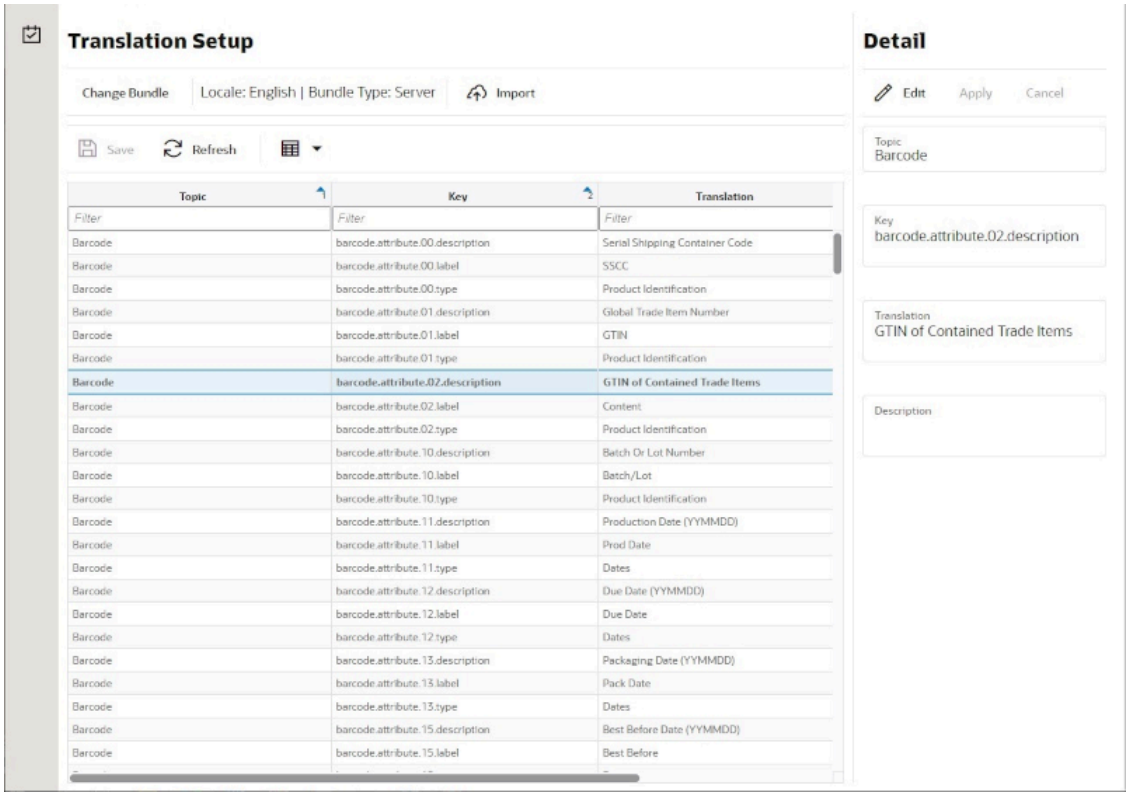
Dialog Buttons

- **Apply**: Clicking this button will load the translation records for the selected bundle criteria.
- **Cancel**: Clicking this button will close the dialog without performing any action.

Translation Setup Screen

The screen allows customization of EICS owned translation records only.

Figure 4-2 Translation Setup Screen



Navigation: Main Menu/Admin/Translations/Translation Setup

Search Bar Options

- **Change Locale**: Clicking this button will popup bundle selection dialog.
- **Filter criteria**: This area displays the current search criteria.
- **Import**: Clicking this button will display the file import dialog. Refer to Translation File Upload section for more details.

List Buttons

- **Save**: Saves any changes made to the translation records.
- **Refresh**: Refreshes the translation records by loading them again.
- **Grid View Menu**: It's a drop-down menu that provides access to options like reset view, enable/disable column filter and export grid data to a CSV file.

List Attributes

- **Topic**: Translation topic for the translation record.
- **Key**: Translation key for the translation record.
- **Translation**: Actual translated text for the translation record.
- **Description**: Any additional description for the translation record.

Detail Buttons

- **Edit**: Enable editing of translation record.
- **Apply**: Apply changes to the translation record.
- **Cancel**: Cancel any changes made to the translation record.

Detail Attributes

- **Topic**: Translation topic for the translation record. It is not editable.
- **Key**: Translation key for the translation record. It is not editable.
- **Translation**: Actual translated text for the translation record. It is editable.
- **Description**: Any additional description for the translation record. It is editable.

In addition to the Translation Setup screen, the system maintains translation records for the following JET screens:

Table 4-2 Translation Data JET Screens

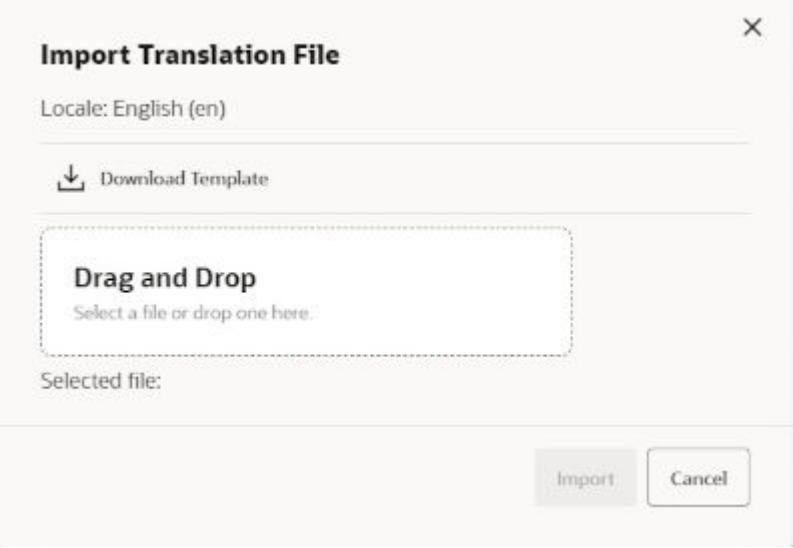
JET Screen	Column
Barcode Processor	Processor Name
Carrier	Description
Carrier Service	Description
Code Info	Description
Inventory Adjustment	Description
Shipment Reason	Description
Sub-bucket	Description
Custom Flexible Attribute	Display Label
Role Detail	Description

Translation File Upload

Translation Setup Screen allow the user to upload translations file. This allows the user to update translations in a file and import that file into the product.

Clicking on the **Import** button on **Translation Setup Screen** displays this dialog that can be used to download the template and upload updated files for the current locale.

Figure 4-3 Import Translation File



- **Locale**: It displays the selected locale.
- **Download Template**: Clicking on this button downloads a template CSV file which can be used to edit translation values across all available bundles for the selected locale. It does not matter which bundle is currently selected on the **Translation Setup Screen**. The file name would be of the format SIOCS_Translations_<locale>_Template.xlsx where locale is the selected locale.
- **Drag and Drop**: To select an updated file for import, the file can be dragged and dropped into the 'Drag and Drop' space on the dialog. Alternatively, a user can browse the file using the file selection wizard by clicking inside the 'Drag and Drop' space. File size should be > 0 MB and <= 5MB and should belong to the selected locale.
- **Selected File**: It displays the name of the selected file.
- **Import**: Clicking this button initiates the file import process. If there were any errors in the processing, an error file will be generated, and the user will be prompted if they want to save the file.
- **Cancel**: Clicking this button will close the dialog without taking any action.

Report Translations

Translation of report templates fall under this category. EICS provides XLF files for each report. At runtime BI publisher identifies the user locale and selects the appropriate XLF file to translate report template.

5
Batches

This section information describes the following topics:

- Overview
- Batch Admin Users
- Batch Configuration
- Operational Batches
- Cleanup Batches
- System Process Batches
- Batch Job Administration

Overview

The batch processes are designed to process large volume of data.

These jobs are mostly used to perform background operations on the transaction and includes such tasks as generation of a transaction, closing the transaction after a specified date is passed, auto confirmation of the transaction, and so on.

Please see the Cleanup Batches section for details on clean up batches.

Batch Admin Users

The following list shows the batch related users. For additional details, please see the *Oracle Retail Enterprise Inventory Cloud Service User Guide*.

Table 5-1 Batch Users and Roles

Job Duties	SIOCS Application Role	IDCS or OCI IAM Application Role
Manage Batch System Configuration	ADMINISTRATOR	admin_users
Access SIOCS Job Admin UI	ADMINISTRATOR	admin_users
Manage SIOCS Adhoc Job	ADMINISTRATOR	batch_users

Batch Configuration

To access the System Configuration screen, navigate Main Menu/Admin /Configuration / System Administration. To view the Batch configuration, filter by Batch topic.

Operational Batches

Operational batches are used to perform background operations on the transaction and includes such tasks as generation of a transaction, closing the transaction after a specified date is passed, auto confirmation of the transaction, and so on.

- Auto Inventory Adjustment
- Auto Replenish Capacity
- Auto Ticket Generate
- Auto Ticket Print
- Clearance File Import
- Generate Problem Line Stock Count
- Generate Unit Amount Stock Count
- Generate Unit Stock Count
- Initial Foundation Data File Import
- Initial Inventory Import
- Initial Store Data File Import
- Inventory Extract Export
- Item Basket Maintenance
- Item Price ICL Import Batch
- Price Change File Import
- Retail Sale Audit Import Batch
- Shelf Replenishment Closure
- Stock Count Authorize Recovery
- Stock Count Auto Cancel
- Stock Count Export
- Stock Count Unit and Amount Snapshot
- Store Order Auto Approve
- Store Order Auto Cancel
- Store Order Auto Generate
- Store Sequence Import
- Third Party Price Import Batch
- Third Party RFID File Import Batch
- Third Party Stock Count Import
- Warehouse Available Inventory Import

Auto Inventory Adjustment

Auto inventory adjustment is the process through which inventory is reduced/increased over time via automatic inventory adjustments.

This functionality helps retailers to do automatic adjustment either to increment or decrement the available inventory for various reasons like wastage for fresh produce which has a short shelf life.

To maintain more accurate inventory values, EICS auto adjustment functionality provides users the ability to create product group type of Auto Inventory Adjustment. Adjustment percentage or standard UOM amounts can be setup on the product group detail and assigned to individual items and/or the item hierarchy.

A user can schedule the date when the auto adjustment batch process must run and when inventory adjustments are automatically made based upon the reason code and its corresponding disposition, and adjustment quantities setup on the product group.

The batch program fetches the auto inventory adjustment product groups that are scheduled and open to be run for the current date and apply the Auto Adjustment percentage or SUOM amount to each item in the product group.

When setting up a product group, the adjustment SUOM and or adjustment % may be entered.

If a percentage and SUOM exist on the product group, the batch program will apply the least amount of the two.

The batch program considers the reason code attached to the product group and its corresponding disposition to adjust the inventory.

The system supports the following adjustment dispositions that can be attached to the product group.

Out to ATS: the system will increment the available inventory.

ATS to out: the system will decrement the available inventory.

The system adjusts the inventory and creates transaction history records using the adjustment reason attached to the product group. Actual inventory adjustment records do not get created. The adjustments will then be sent over the RIB to the merchandising system.

Batch Job Definition Name

AutoInventoryAdjustment_OpsJob

Batch Job Parameters

<input_date>input_dateinput date parameter is defaulted to current timestamp if not specified. System batch input date format is used for parsing input date if specified.

Key Tables

Table 5-2 Key Tables for Auto Inventory Adjustment

Table	Select	Insert	Update	Delete
inv_adjust_reason	Yes			
product_group_item	Yes			
product_group_hierarchy	Yes			
product_group_sched_store	Yes		Yes	
product_group_schedule	Yes			
product_group	Yes			
store_item	Yes			

Table 5-2 (Cont.) Key Tables for Auto Inventory Adjustment

Table	Select	Insert	Update	Delete
store_item_stock	Yes		Yes	
store_item_stock_history	Yes	Yes		
mps_staged_message		Yes		

Restart/Recovery

This batch can be re-run by starting a new batch job after the issues are resolved.

Auto Replenish Capacity

This job automatically updates the shopfloor to capacity for certain items based on a product group.

Batch Job Definition Name

AutoReplenishCapacity_OpsJob

Batch Job Parameters

<input_date>input_dateinput date parameter is defaulted to current timestamp if not specified. System batch input date format is used for parsing input date if specified.

<store id>

Where store id is store identifier. If store id is not specified, then data for all managed stores will be processed in parallel processing.

Key Tables

Table 5-3 Key Tables for Auto Replenish Capacity Batch

Table	Select	Insert	Update	Delete
config_system	Yes			
shelf_replenish	Yes	Yes	Yes	Yes
shelf_replenish_line_item	Yes	Yes	Yes	Yes

Restart/Recovery

This batch can be re-run by starting a new batch job after the issues are resolved.

Auto Ticket Generate

This job automatically generates the item tickets and labels depending on the store parameters for events which are subscribed. The events are price changes, clearance event, promotions, and item description changes.

The batch also generates tickets for future price change events based on **Auto Ticket Generate Future Days** store parameter value that indicates how many days of future price events are considered to generate the tickets when the price events are coming to SIOCS.

Batch Job Definition Name

TicketAutoGenerateFromEvent_OpsJob

Batch Job Parameters

<input_date>input_dateinput date parameter is defaulted to current timestamp if not specified. System batch input date format is used for parsing input date if specified.

<store id>

Where store id is store identifier. If store id is not specified, then data for all managed stores will be processed in parallel processing.

Key Tables

Table 5-4 Key Tables for Auto Ticket Generate Batch

Table	Select	Insert	Update	Delete
config_system	Yes			
ticket_event	Yes	Yes	Yes	
ticket	Yes	Yes	Yes	Yes
ticket_format	Yes			

Restart/Recovery

This batch can be re-run by starting a new batch job after the issues are resolved.

Auto Ticket Print

This batch sends the tickets generated to the 3rd party printing service. The tickets/labels generated for the items identified in the product group will be automatically sent to the 3rd party printing service.

Batch Job Definition Name

TicketAutoPrint_OpsJob

Batch Job Parameters

<input_date>input_dateinput date parameter is defaulted to current timestamp if not specified. System batch input date format is used for parsing input date if specified.

<store id>

Where store id is store identifier. If store id is not specified, then data for all managed stores will be processed in parallel processing.

Key Tables

Table 5-5 Key Tables for Auto Ticket Print Batch

Table	Select	Insert	Update	Delete
config_system	Yes			

Table 5-5 (Cont.) Key Tables for Auto Ticket Print Batch

Table	Select	Insert	Update	Delete
Store_printer	Yes			
ticket	Yes	Yes	Yes	Yes
ticket_format	Yes			

Restart/Recovery

This batch can be re-run by starting a new batch job after the issues are resolved.

Clearance File Import

This batch imports the RPCS (Retail Pricing Cloud Service) clearance records via the flat file. The batch processes the records for items ranged in stores (regardless managed or non-managed stores). If the record type is delete, the matching record in the database will be deleted. For a record type of insert/replace/update, the import is UPSERT. If the record does not exist in the database, it will be inserted; if the data already exists in the database, it will be updated.

The price records merge/upsert the data from the staging tables into the application master table ITEM_PRICE on the combination of store/item/pricetype/ ext_price_event_id.

On processing the clearance reset record (reset indicator is 1), all active clearances for that store/item which does not have an end date, that end date will be set to the clearance reset effective date.

File Handling Details

1. File provider application uploads the relevant data files to the import's location in Object Storage via FTS. See Upload Import Data Files to Object Storage for details.

   > **Note:**
   >
   > For files from Retail Pricing Cloud Service (RPCS) in legacy cloud services, RPCS price transactions will be sent via BDI File Creator Process flow from RPCS (legacy cloud services) to SIOCS Next Gen Cloud Services object storage import's location.

2. The Import Batch job will download the relevant data files from Object Storage, parse the files and insert the data into staging tables, merge/upsert the data from staging tables into SIOCS master tables, and upload any failed files/ records to the rejects folder to Object Storage.
3. On completion, the data files are moved to archive file locations and will be purged after configured days.
4. On failures, the failed records are written to reject files, and the reject files are sent to object storage reject's location. The error will be visible in by drilling down from the Job Admin screen on the failed job execution to display the batch detail. Drill down on the failed batch details to see the error message.
5. To re-run the corrected data files, repeat step 1 and 2.

File Layout

See Appendix: Batch File Layout Specifications for details.

Batch Job Definition Name

ClearanceFileImport_OpsJob

Batch Job Parameters

N/A

Key Tables

Table 5-6 Key Tables for Clearance File Import

Tables	Select	Insert	Update	Delete
Item_price	Yes	Yes	Yes	Yes
ICL_CLEARANCE	Yes	Yes	Yes	Yes

Generate Problem Line Stock Count

Before the batch process runs, the retailer establishes a group of items and item hierarchies (by associating them to the problem line group type) and selects applicable parameters (negative SOH, negative available, and so on). The problem line batch process goes through the list of items in the group, determining which fall within the parameters. The system automatically creates a stock count from those items that do fall within the parameters.

If an item is a problem line item (negative inventory for example) on a stock count, and the user does not get the chance to perform the stock count on it that day, the next day the item may no longer be a problem line (positive inventory). However, the system continues to create a stock count for that item because a problem existed at one time.

Batch Job Definition Name

GenerateProblemLineStockCount_OpsJob

Batch Job Parameters

<input_date>

Where input date parameter is defaulted to current timestamp if not specified. System batch input date format is used for parsing input date if specified.

<store id> Where store id is store identifier. If store id is not specified, then data for all managed stores will be processed in parallel processing.

Key Tables

Table 5-7 Key Tables for Problem Line Stock Count Batch

Tables	Select	Insert	Update	Delete
group_schedule_extract	Yes	Yes		
prod_group_item_bkdn			Yes	Yes

Table 5-7 (Cont.) Key Tables for Problem Line Stock Count Batch

Tables	Select	Insert	Update	Delete
stock_count	Yes	Yes	Yes	Yes
stock_count_line_item	Yes	Yes	Yes	Yes
stock_count_line_item_uin	Yes	Yes	Yes	Yes
stock_count_child	Yes	Yes	Yes	Yes
product_group_schedule	Yes		Yes	
product_group	Yes			
product_group_sched_store	Yes			
item	Yes			
store_item	Yes			
stock_count_line_item	Yes			

Generate Unit Amount Stock Count

This batch program generates Unit Amount stock counts.

On a daily basis, the batch process creates the stock counts that are scheduled for the current day or future date which matches the next scheduled date. The system looks at all the scheduled stock count records and determines whether any are scheduled for today or the user-specified future date. The process creates the stock counts for each individual store. For example, if a scheduled count includes a list of five stores, then five separate stock count records are created.

If an all-location stock count is being run, the batch processing generates individual counts for every macro sequence location.

The date parameter is optional when running the Extract Stock Counts batch. If no date is provided, today's date is used.

Batch Job Definition Name

GenerateUnitAmountStockCount_OpsJob

Batch Job Parameters

<input_date>Where input date parameter is defaulted to current timestamp if not specified. System batch input date format is used for parsing input date if specified.

<store id>

Where store id is store identifier. If store id is not specified, then data for all managed stores will be processed in parallel processing.

Key Tables

Table 5-8 Key Tables for Generate Unit Amount Stock Count Batch

Table	Select	Insert	Update	Delete
group_schedule_extract		Yes		Yes
product_group	Yes			
product_group_hierarchy	Yes			
product_group_item	Yes			
product_group_sched_store	Yes			
product_group_schedule	Yes		Yes	
product_group_item_bkdn		Yes		Yes
stock_count	Yes	Yes	Yes	
stock_count_child		Yes	Yes	
stock_count_line_item		Yes	Yes	
stock_count_line_item_uin		Yes	Yes	
item	Yes			
store_item	Yes			
store_item_stock	Yes			
item_component	Yes			

Generate Unit Stock Count

This batch program generates Unit stock counts.

On a daily basis, the batch process creates the stock counts that are scheduled for the current day or future date which matches the next scheduled date. The system looks at all the scheduled stock count records and determines whether any are scheduled for today or the user specified future date. The process creates the stock counts for each individual store. For example, if a scheduled count includes a list of five stores, then five separate stock count records are created.

If the system is configured to use unguided stock counts, the batch process does not generate multiple counts even if the item is located at multiple locations within the store.

The date parameter is optional when running the Extract Stock Counts batch. If no date is provided, today's date is used.

Batch Job Definition Name

GenerateUnitStockCount_OpsJob

Batch Job Parameters

<input_date>

Where input date parameter is defaulted to current timestamp if not specified. System batch input date format is used for parsing input date if specified.

<store id> Where store id is store identifier. If store id is not specified, then data for all managed stores will be processed in parallel processing.

Key Tables

Table 5-9 Key Tables for Generate Unit Stock Count Batch

Table	Select	Insert	Update	Delete
group_schedule_extract		Yes		Yes
product_group	Yes			
product_group_hierarchy	Yes			
product_group_item	Yes			
product_group_sched_store	Yes			
product_group_schedule	Yes		Yes	
product_group_item_bkdn		Yes		Yes
stock_count	Yes	Yes	Yes	Yes
stock_count_child		Yes	Yes	Yes
stock_count_line_item		Yes	Yes	Yes
stock_count_line_item_uin		Yes	Yes	
item	Yes			
store_item	Yes			
store_item_stock	Yes			
item_component	Yes			

Initial Foundation Data File Import

This batch imports initial foundation seed data files from external system. See Standalone Data Seeding in the Batches chapter for details.

Batch Job Definition Name

StandaloneIdlFileImport_OpsJob

Batch Job Parameters

N/A

Restart/Recovery

This batch can be re-run by starting a new batch job after the issues are resolved.

Initial Inventory Import

The Initial Inventory Import batch is used to wipe out the existing SOH data for items in a store and override it with the new SOH data from the third-party/non-Oracle/legacy systems.

This batch is meant to be used ONLY during implementation. It is designed for optimal upload to rewrite SOH.

It should not be used after the one-time inventory upload.

> **Note:**
>
> UINs must be uploaded only once. Unlike stock on hand, UINs are state driven and control the stock on hand. Only new UINs in the flat file will be considered for stock on hand update. As such, if UINs already exist, they will not reflect into the new SOH.

Batch Job Definition Name

InitialInventoryImport_OpsJob

Batch Job Parameters

N/A

File Error Handling

The batch job will be marked as failed if the file staging fails. The staging process is all or none transaction so if an error occurs during the batch process, none of the transactions in the file will be staged. The user will need to rerun the same file again after resolving any errors.

Key Tables

Table 5-10 Key Tables for Initial Inventory Import Batch

Table	Select	Insert	Update	Delete
store_item_stock	Yes		Yes	
item_uin	Yes	Yes	Yes	

Initial Store Data File Import

This batch imports initial store seed data files from external system. See Standalone Data Seeding in the Batches chapter for details.

Batch Job Definition Name

StandaloneIdlStoreFileImport_OpsJob

Batch Job Parameters

<store id> Where store id is store identifier.

Restart/Recovery

This batch can be re-run by starting a new batch job after the issues are resolved.

Inventory Extract Export

This batch extracts the inventory to a file that has been altered on or after the specified date for the specified store and uploads the file to FTS. The type of upload files is based on system parameter configuration value *Compress inventory extract files into zip file.*The system supports only exporting inventory records for managed stores that support OMNI channel orders.

File Layout

See Inventory Extract Export File Specification.

Batch Job Definition Name

InventoryExtract_OpsJob

Batch Job Parameters

<input_date>

input_dateinput date parameter is defaulted to current timestamp if not specified. System batch input date format is used for parsing input date if specified.

<store id>

Where store id is store identifier. If store id is not specified, then data for all managed stores will be processed in parallel processing.

Key Tables

Table 5-11 Key Tables for Inventory Extract Batch

Table	Select	Insert	Update	Delete
STOCK_ITEM_V	Yes			

Item Basket Maintenance

This batch updates the item basket status cancelled when the item basket has expired.

Batch Job Definition Name

ItemBasketMaintenance_OpsJob

Batch Job Parameters

<input_date>Where input_date is defaulted to current timestamp if not specified. It is used for comparing if a record date is a configured number of days prior to the input date. System batch input date format is used for parsing input date if specified.

Key Tables

Table 5-12 Key Tables for Item Basket Maintenance Batch

Table	Select	Insert	Update	Delete
config_system	Yes			
Item_basket	Yes		Yes	

Restart/Recovery

This batch can be re-run by starting a new batch job after the issues are resolved.

Item Price ICL Import Batch

This batch program searches the integration pricing event log records from the sourcing system and stage the price change and clearance change log records into SIOCS Integration Change Log Staging tables. The staged change log records will be processed by MPS Worker (DcsPrice) to import staging data into application tables.

Batch Job Definition Name

ItemPriceIclImport_OpsJob

Batch Job Parameters

N/A

Key Tables

Table 5-13 Key Table for Item Price ICL Import Batch

Table	Select	Insert	Update	Delete
ICLS_PRICE_CHANGE	Yes	Yes		Yes
ICLS_CLEARANCE	Yes	Yes		Yes
ITEM_PRICE	Yes	Yes	Yes	Yes

Restart/Recovery

This batch can be re-run by starting a new batch job after the issues are resolved.

POS Transaction Import

This batch imports POS transaction records from the flat file (SIMT-LOG file) that came from Pont of Sale System.

The batch process takes the sales/order transaction data and stages them to the database staging table (POS_TRANSACTION) from where they are picked up by the MPS worker to update the store item's inventory buckets (for example, store item's total quantity, shop floor quantity), if applicable.

The file will contain both sale and order transactions. The batch will assign separate request IDs to sales and order transactions.

For sale transactions, a single request ID cannot contain more than MAX_VALUE = 500 transaction line items with an exception that a single transaction ID cannot span across multiple request IDs.

For order transactions, a single request ID cannot contain more than MAX_VALUE = 500 transaction line items with an exception that a single customer order ID cannot span across multiple request IDs.

The file contains transactions for a single store.

The customer can set the job scheduler to be run multiple times per day by changing the schedule intervals.

File Handling Details

1. File provider application uploads the relevant data files to the import's location in Object Storage via FTS. See: Upload Import Data Files to Object Storage for details.
2. The Import Batch job will download the relevant data files from Object Storage, parse the files and insert the data into POS_TRANSACTION staging tables, to the rejects folder to Object Storage. If validation errors occurred during loading process, e.g invalid store, or duplicate the extended transaction id exists in POS_TRANSACTION table, then the entire file will be rejected.
3. On completion, the data files are moved to archive file locations and will be purged after configured days.
4. On failures, the reject files are sent to object storage reject's location. The error will be visible in by drilling down from the Job Admin screen on the failed job execution to display the batch detail. Drill down on the failed batch details to see the error message.
5. To re-run the corrected data files, repeat step 1 and 2.

File Specification

File Name format: SIMTLOG _<date in YYYYMMDDHH24MISS format>_>loc id>.dat (where loc id is the store identifier)'

File Layout: See: POS Sale Transaction Import File Specification.

Batch Job Definition Name

PosTransactionImport_OpsJob

Batch Job Parameters

<File Name>

If not specified, then the data file in incoming directory are processed.

File Error Handling

The batch job will be marked as failed if the loading file to staging table fail. The staging process is all or none transaction so if an error occurs during the batch process, none of the transactions in the file will be staged. The user will need to re-upload the data file after resolving any errors for processing.

Key Tables

Table 5-14 Key Tables for POS Transaction Import Batch

Table	Select	Insert	Update	Delete
pos_transaction	Yes	Yes		
item	Yes			
inv_adjust_reason	Yes			

Price Change File Import

This batch imports the regular price change records via flat files for hybrid pricing integration between price changes on legacy cloud services and SIOCS Next Gen Cloud Services.

The batch processes the records for items ranged in stores (regardless managed or non-managed stores). If the record type is delete, the matching record in the database will be deleted. For a record type of insert/replace/update, the import is UPSERT. If the record does

not exist in the database, it will be inserted; if the data already exists in the database, it will be updated.

File Handling Details

1. File provider application uploads the relevant data files to the import's location in Object Storage via FTS. See Upload Import Data Files to Object Storage for details.

> **Note:**
>
> For files from Retail Pricing Cloud Service (RPCS) in legacy cloud services, RPCS price transactions will be sent via BDI File Creator Process flow from RPCS (legacy cloud services) to SIOCS Next Gen Cloud Services object storage imports location.

2. The Import Batch job will download the relevant data files from Object Storage, parse the files and insert the data into staging tables, merge/upsert the data from staging tables into SIOCS master tables, and upload any failed files/ records to the reject's folder to Object Storage.
3. On completion, the data files are moved to archive file locations and will be purged after configured days.
4. On failures, the failed records are written to reject files, and the reject files are sent to object storage rejects location. The error will be visible by drilling down from the Job Admin screen on the failed job execution to display the batch detail. Drill down on the failed batch details to see the error message.
5. To re-run the corrected data files, repeat step 1 and 2.

File Layout

See Appendix: Batch File Layout Specifications for details.

Batch Job Definition Name

PriceChangeFileImport_OpsJob

Batch Job Parameters

N/A

Key Tables

Table 5-15 Key Tables for Price Change File Import

Tables	Select	Insert	Update	Delete
item_price	Yes	Yes	Yes	Yes
ICL_PRICE_CHANGE	Yes	Yes	Yes	Yes

Retail Sale Audit Import Batch

This batch program imports sales/order transaction data (ReSA File) that originated in Oracle Retail Xstore Point of Service. The external audit system will provide in its sales upload file a

percentage or quantity that indicates how much the inventory needs to be reduced by, in addition to the sold quantity.

For example, meat will become lighter as fluids evaporate. Other items, for example cheese or ham, will only be reduced when of the outside layers are cut off to sell the item.

The batch process takes the sales transaction data to update the store item's inventory buckets. From the batch program, SIOCS learns about inventory movement (that is, what is sold, what is returned, what is reserved and what is fulfilled). Once SIOCS attains the data, SIOCS assumes that sales should be taken from the store's shelf-related inventory buckets. This assumption is important to SIOCS's shelf replenishment processing. SIOCS assumes that returns should go to the backroom bucket; the system's logic is that returns must be inspected.

The batch takes the sales/order transaction data and stage them to the SIOCS database staging table from where they are picked up by the polling timer framework to update the store item's inventory buckets (for example, store item's total quantity, shop floor quantity), if applicable.

The file will contain both sales and order transactions. The batch job combines the transaction number and register number to form the transaction ID in SIOCS. Request IDs are assigned to the transactions in such a way that a single request ID will not contain more than MAX_SIZE=500 records with an exception that a single transaction ID should not span across multiple request IDs.

During processing staging records, batch also writes each failure record into a transaction log table.

Each job run will pick number of files (defined by Maximum Job Instances Per Batch) in system configuration and process them in multi-threads. The number of files to be processed is default to 20, the value can be configured via system configuration screen.

Customer can set the job scheduler to be run multiple times per day by changing the schedule intervals.

File Specification

File Name format: SIMT_<date in YYYYMMDDHH24MISS format>_<loc id>

Where <loc id> is store id.

File Layout: See Retail Sale Audit Import File Specification.

Batch Job Definition Name

RetailSalesAuditImport_OpsJob

Batch Job Parameters

<File name>

If not specified, then the data file in incoming directory are processed.

File Error Handling

The batch job will be marked as failed if the file staging fails. The staging process is all or none transaction so if an error occurs during the batch process, none of the transactions in the file will be staged. The user will need to rerun the same file again after resolving any errors.

Key Tables

Table 5-16 Key Tables for Retail Sale Audit Import Batch

Tables	Select	Insert	Update	Delete
pos_transaction		Yes		
inv_adjust_reason	Yes			

Shelf Replenishment Closure

The end of day batch process runs at the end of each day to reset the delivery bay and close any open pending shelf replenishments. The system takes the entire inventory from the delivery bay and moves it to the back room. Any pending or in progress shelf replenishment are changed to a cancelled state. Users who are performing a shelf replenishment are kicked out of the system. That is, the batch process takes over the shelf replenishment user's application activity locking. The current user's shelf replenishment process is discarded without being saved. After the batch process is run, all shelf replenishments are either completed or cancelled, and the delivery bay has zero inventory.

Batch Job Definition Name

CleanupShelfReplenishment_OpsJob

Batch Job Parameters

input_date>input_dateinput date parameter is defaulted to current timestamp if not specified. System batch input date format is used for parsing input date if specified.

Key Tables

Table 5-17 Key Tables for Cleanup Shelf Replenishment Batch

Table	Select	Insert	Update	Delete
shelf_replenish			Yes	
stock_item_stock			Yes	

Stock Count Authorize Recovery

This batch process looks for stock counts that are stuck in Authorize Processing state. This is a unique state that appears when an error occurs during the final processing of a stock count. The batch attempts to fully authorize the stock count. Errors that occur during the batch process are logged to the server error logs and will indicate the reason for any further processing failures. Successfully authorized stock counts will move to authorized completed state.

Batch Job Definition Name

StockCountAuthorizeRecovery_OpsJob

Batch Job Parameters

<input_date>Where input date parameter is defaulted to current timestamp if not specified. System batch input date format is used for parsing input date if specified.

Key Tables

Table 5-18 Key Tables for Stock Count Authorize Recovery Batch

Tables	Select	Insert	Update	Delete
stock_count	Yes		Yes	
stock_count_child	Yes		Yes	
stock_count_line_item	Yes		Yes	
stock_count_line_item_uin	Yes			
item_uin	Yes		Yes	
store_item	Yes			
store_item_stock			Yes	
product_group_schedule	Yes			
product_group_sched_store	Yes			
store	Yes			
stock_count_sale	Yes			Yes
inv_adjust_reason	Yes			

Stock Count Auto Cancel

This batch finds the stock counts which are not completed or un-executed based on the “Days To Hold Before Auto Canceling Stock Counts”, updates the stock count status as canceled (status = 20), for those stock count in progress and not completed, the open stock counts field in store item stock table will be decremented.

For unit amount stock count, the canceled stock count schedule id/store will be published via MPS Staged Messages (Message Family: StkCountSch, MessageType: StkCountSchDtlDel).

It is recommended to run this batch prior running the “Cleanup Stock Counts” batch.

Batch Job Definition Name

StockCountCancel_OpsJob

Batch Job Parameters

<input_date>

Where input_date is the date parameter to be used to find the matching records, if not specified, it is defaulted to current system timestamp. System batch input date format is used for parsing input date if specified.

Key Tables

Table 5-19 Key Tables for Stock Count Auto Cancel

Table	Select	Insert	Update	Delete
STOCK_COUNT	Yes		Yes	
STORE_ITEM_STOCK	Yes		Yes	
STOCK_COUNT_CANCEL	Yes	Yes	Yes	Yes

Restart/Recovery

This batch can be re-run by starting a new batch job after the issues are resolved.

Stock Count Export

This batch process looks for a stock count that is stuck in approval authorized state during authorizing a unit amount stock count process. This is a unique state that appears when an error occurs during the final processing of a unit amount stock count. The batch attempts to generate stock count export file and set stock count status to authorize complete state. Errors that occur during the batch process are logged to the server error logs and will indicate the reason for any further processing failures.

File Layout

See Appendix: Batch File Layout Specifications for details.

Batch Job Definition Name

StockCountExport_OpsJob

Batch Job Parameters

<stock_count_id>

Where the stock_count_id is the stock count identifier

Key Tables

Table 5-20 Key Tables for Stock Count Export Batch

Table	Select	Insert	Update	Delete
stock_count	Yes		Yes	
stock_count_child	Yes		Yes	
stock_count_line_item	Yes		Yes	
stock_count_line_item_uin	Yes			
item_uin	Yes		Yes	
store_item	Yes			
stock_count_export	Yes		Yes	Yes

Stock Count Unit and Amount Snapshot

This job takes the snapshot of current inventory for Unit and Amount Type of Stock Counts for the items across all stores or for a specific store.

Batch Job Definition Name

StockCountUnitAndAmountSnapshot_OpsJob

Batch Job Parameters

<store id> Where store id is store identifier. If store id is not specified, then data for all managed stores will be processed in parallel processing.

Key Tables

Table 5-21 Key Tables for Stock Count Unit And Amount Snapshot Batch

Table	Select	Insert	Update	Delete
stock_count	Yes		Yes	
stock_count_child	Yes		Yes	
stock_count_line_item	Yes		Yes	
Store_item	Yes			
store_item_stock	Yes			

Store Order Auto Approve

This batch processes looks for store orders which requested date is X hours of old than the number of hours after create date in SIOCS to approve store orders.

Batch Job Definition Name

StoreOrderAutoApprove_OpsJob

Batch Job Parameters

N/A

Key Tables

Table 5-22 Key Tables for Store Order Auto Approve Batch

Table	Select	Insert	Update	Delete
store_order	Yes		Yes	
store_order_line_item	Yes		Yes	

Store Order Auto Cancel

This batch processes looks for store orders which requested date is X days of old than the system defined **Days to hold before Auto Canceling Store Orders**, and the process attempts to set those store orders to cancel state when applicable.

Batch Job Definition Name

StoreOrderAutoCancel_OpsJob

Batch Job Parameters

N/A

Key Tables

Table 5-23 Key Tables for Store Order Auto Cancel Batch

Table	Select	Insert	Update	Delete
store_order	Yes		Yes	
store_order_line_item	Yes		Yes	

Store Order Auto Generate

This batch processes generate store order records for store order product group schedules.

Batch Job Definition Name

StoreOrderAutoGenerate_OpsJob

Batch Job Parameters

N/A

Key Tables

Table 5-24 Key Tables for Store Order Auto Generate Batch

Table	Select	Insert	Update	Delete
store_order	Yes	Yes	Yes	
store_order_line_item	Yes	Yes	Yes	
group_schedule_extract		Yes		Yes
product_group	Yes			
product_group_hierarchy	Yes			
product_group_item	Yes			
product_group_sched_store	Yes			
product_group_schedule	Yes		Yes	

Store Sequence Import

This batch imports store sequencing information from a flat file via the File Transfer Service (FTS). Each job run will pick number of files (defined by **Maximum Job Instances Per Batch**) in system configuration and process them in multi-threads. The number of files to be processed is default to 20, the value can be configured via system configuration screen.

The action of the import depends on the optional DELETEALL value in the header record. If DELETEALL is present, the existing records for the store are deleted from the store_sequence_area and store_sequence_item tables. The contents of the import are inserted into the database after the deletion. If DELETEALL is not present, the contents of the import file are merged into the existing data and inserted for new data.

The import validates the store and item ids during the processing. The import will fail if both the specified store and item are not in the store and item table.

Customer can set the job scheduler to be run multiple times per day by changing the schedule intervals.

File Layout

See Appendix: Batch File Layout Specifications for details.

Batch Job Definition Name

StoreSequenceImport_OpsJob

Batch Job Parameters

<File name>

If not specified, then the data file in incoming directory are processed.

File Error Handling

The file loading process is all or none transaction so if an error occurs during the batch process, none of the transactions in the file will be committed. The user will need to rerun the same file again after resolving any errors. The error will be visible in by drilling down from the Job Admin screen on the failed job execution to display the batch detail. Drill down on the failed batch details to see the error message.

Key Tables

Table 5-25 Key Tables for Store Sequence Import Batch

Tables	Select	Insert	Update	Delete
store_sequence_area	Yes	Yes	Yes	Yes
store_sequence_item	Yes	Yes	Yes	Yes

Third Party Price Import Batch

This batch imports pricing data (regular price, clearance, and promotion prices) from a third party uploaded pricing data files into SIOCS.

The price records merge/upsert the data from staging tables into the application master table ITEM_PRICE on the combination of store/item/pricetype/ ext_price_event_id.

File Handling Details

1. File provider application uploads the relevant data files to the import's location in Object Storage via FTS. See Upload Import Data Files to Object Storage for details.
2. The Import Batch job will download the relevant data files from Object Storage, parse the files and insert the data into staging tables, merge/upsert the data from staging tables into

SIOCS master tables, and upload any failed files/ records to the rejects folder to Object Storage.

3. On completion, the data files are moved to archive file locations and will be purged after configured days.
4. On failures, the failed records are written to reject files, and the reject files are sent to object storage reject's location. The error will be visible in by drilling down from the Job Admin screen on the failed job execution to display the batch detail. Drill down on the failed batch details to see the error message.
5. To re-run the corrected data files, repeat step 1 and 2.

File Layout

See Appendix: Batch File Layout Specifications for details.

Batch Job Definition Name

ExtPriceImport_OpsJob

Batch Job Parameters

N/A

Key Tables

Table 5-26 Key Tables for Third Party Price Import Batch

Table	Select	Insert	Update	Delete
item_price	Yes	Yes	Yes	Yes

Third Party RFID File Import Batch

This batch process imports bulk amount of RFID information from batch files which are uploaded by customers.

The RFID importer first sets the present attribute to 'N' for all existing RFID tags at the location thereby removing them from inventory. It then set the present attribute to 'Y' (yes) for each RFID tag in the import.

CREATE and DELETE are the only two valid actions for RFID, CREATE indicates "present in store" and DELETE indicates "absent from store", the only states an EPC has.

If an EPC in the data file has DELETE type, and exists in database, the process marks the EPC as not present.

If an EPC in the data file has CREATE type, the process inserts or updates in RFID table and mark as present.

Each file contains RFID information for a single store, store/item/action date uniquely identify a RFID record.

File Handling Details

1. Customer uploads the relevant data files to the imports folder in Object Storage via FTS. See Upload Import Data Files to Object Storage for details.

2. The Import Batch job will download the relevant data files from Object Storage, parse the files and insert the data into staging tables, merge/upsert the data from staging tables into SIOCS master tables, upload any failed files/ records to the rejects folder to Object Storage.
3. On completion, the data files are moved to archive file locations and will be purged after configured days.
4. File Error Handling. The import process writes the erroneous records into reject files and uploads to the rejects folder to Object Storage. The error will be visible in by drilling down from the Job Admin screen on the failed job execution to display the batch detail. Drill down on the failed batch details to see the error message.
5. After errors are resolved, to process the corrected data file, repeat steps 1 to 2.

File Layout

See Appendix: Batch File Layout Specifications for details.

Batch Job Definition Name

ExtRfidImport_OpsJob

Batch Job Parameters

N/A

Key Tables

Table 5-27 Key Tables for Third Party RFID File Import Batch

Table	Select	Insert	Update	Delete
RFID	Yes	Yes	Yes	
RFID_HISTORY	Yes	Yes	Yes	
RFID_ZONE	Yes			
DLS_RFID	Yes	Yes	Yes	Yes

Third Party Stock Count Import

This batch imports the stock count quantities when a stock count is setup in SIOCS, and physical counting is conducted by a third party. The batch updates the stock count counted or recounted quantities. Invalid records during the import are saved in the rejected item table.

When the stock count is set up as **Auto-authorize Unit and Amount Stock Count**, the rejected items are processed, and attempts are resolution are taken (such as ranging items and adding them to the stock count). In addition, the authorization process occurs and the stock on hand quantities for the items are updated. In addition, a **Unit and Amount Stock Counts Export** file is generated because of stock count auto authorization.

Each job run will pick number of files (defined by **Maximum Job Instances Per Batch**) in system configuration and process them in multi-threads. The number of files to be processed is default to 20. The value can be configured via system configuration screen.

Customer can set the job scheduler to be run multiple times per day by changing the schedule intervals.

File Layout

See Appendix: Batch File Layout Specifications for details.

Batch Job Definition Name

ThirdPartyStockCountImport_OpsJob

Batch Job Parameters

N/A

File Error Handling

The batch job will be marked as failed if the file staging fails. The staging process is all or none transaction so if an error occurs during the batch process, none of the transactions in the file will be staged. The user will need to rerun the same file again after resolving any errors.

Key Tables

Table 5-28 Key Tables for Third Party Stock Count Import Batch

Table	Select	Insert	Update	Delete
stock_count_import	Yes		Yes	
stock_count_rejected_item		Yes		
stock_count	Yes		Yes	
stock_count_child	Yes		Yes	
stock_count_line_item	Yes		Yes	
item_price	Yes			
item	Yes			
store_item	Yes			
item_uin	Yes			
stock_count_line_item_uin	Yes			

Warehouse Available Inventory Import

This batch imports warehouse available inventory from a CSV file. The batch updates the warehouse item available quantity. This is calculated by subtracting transfer reserved qty, customer reserved qty, non-sellable inventory and RTV from stock on hand. Available inventory is in the standard unit of measure.

File Handling Details

1. Customer uploads the relevant data files to the imports folder in Object Storage via FTS. See Upload Import Data Files to Object Storage for details.
2. The Import Batch job will download the relevant data files from Object Storage, parse the files and insert the data into staging tables, merge/upsert the data from staging tables into SIOCS master tables, upload any failed files/ records to the rejects folder to Object Storage.

3. On completion, the data files are moved to archive file locations and will be purged after 7 days.
4. File Error Handling. The import process writes the erroneous records into reject files and uploads to the rejects folder to Object Storage. The error will be visible in by drilling down from the Job Admin screen on the failed job execution to display the batch detail. Drill down on the failed batch details to see the error message.
5. After errors are resolved, to process the corrected data file, repeat steps 1 to 2.

File Layout

See Appendix: Batch File Layout Specifications for details.

Batch Job Definition Name

WarehouseAvailInvFileImport_OpsJob

Batch Job Parameters

N/A

Key Tables

Table 5-29 Key Tables for Warehouse Available Inventory Import

Tables	Select	Insert	Update	Delete
WAREHOUSE_ITEM	Yes	Yes	Yes	
DLS_WAREHOUSE_ITEM	Yes	Yes		Yes

Cleanup Batches

Removal of temporary, staged, non-essential data is critical for smooth running of business. If this data is not purged at frequent interval, then these tables can grow to such an extent that normal business operations would get impacted; backup and disaster recovery will take enormous amount of time. Since purge process locks database records, cleanup/purge must be done at short intervals.

Some of these clean-up jobs are restricted and enabled by default.

Customers can configure number for day to retain the records in database via System Admin Parameters.

Table 5-30 Cleanup Batches

Batch Name	Description	Default	Minimum Value	Maximum Value
Cleanup Activity Locks	Deletes activity lock records from ACTIVITY_LOCK table. Any lock record with a lock date/timestamp older than Days to Hold Locking Records system configuration value will be deleted	1 hour	30 minutes	24 hours

Table 5-30 (Cont.) Cleanup Batches

Batch Name	Description	Default	Minimum Value	Maximum Value
Cleanup Adhoc Stock Count	Deletes ad hoc stock counts with a status of in progress. Any ad hoc stock count with a creation date/time stamp older than the Days to Hold In Progress Ad Hoc Counts parameter value will be deleted.	24 hours	30 minutes	24 hours
Cleanup Activity History	Deletes activity history records from ACTIVITY_HISTORYtable. Record with create date older than **Days to Hold Audit Records** system configuration value will be deleted.	24 hours	30 minutes	24 hours
Cleanup Batch Job Repo	Deletes batch job repository records from JOBINSTANCEDATA table and associated tables. Records with create time older than **Days to Hold Batch Repository Record**s system configuration value will be deleted.	24 hours	30 minutes	24 hours
Cleanup Item UIN History	Deletes Item UIN history records from ITEM UIN HISTORY table. Records with create date older than Days to Hold UIN Audit Information system configuration value will be deleted.	24 hours	30 minutes	24 hours
Cleanup Recently Edited	Deletes recently edited security user history records from SECURITY_USER_HISTORY table. Record with update date older than **Days to Hold Recently Edited** system configuration value will be deleted.	24 hours	30 minutes	24 hours
Cleanup RFID History	Deletes RFID history records from RFID_HISTORY table. Records with event date older than **Days to Hold RFID History** system configuration value will be deleted	24 hours	30 minutes	24 hours
Cleanup Store Order	Deletes canceled or approved Store Order records from STORE_ORDER table and associated tables. Records with update date older than **Days to Hold Store Orders** system configuration value will be deleted	24 hours	30 minutes	24 hours
Cleanup Tickets	Deletes ticket records from TICKET table. Records with create date older than **Days to Hold Ticket** system configuration value will be deleted.	24 hours	30 minutes	24 hours
Cleanup Ticket Histories	Deletes ticket history records from TICKET table. Records with printed date older than **Days to Hold Ticket history** system configuration value will be deleted.	24 hours	30 minutes	24 hours

Table 5-30 (Cont.) Cleanup Batches

Batch Name	Description	Default	Minimum Value	Maximum Value
Cleanup Batch Activity	This job deletes the activity records that are no longer needed after the default time specified and if such records have a status different than COMPLETED.	30 minutes	30 minutes	24 hours
Cleanup Batch Data Error	Deletes the batch data errors records that are no longer needed after the default time specified.	30 minutes	30 minutes	24 hours
Cleanup Batch Directories	Deletes the processed files from batch archive folder and failed files from reject folder of the respective job's directory.	24 hours	30 minutes	24 hours
Cleanup Batch Log	Deletes old batch log records. Batch log record with an end date/ timestamp older than the Days To Hold Batch Logs system configuration value and with the Status value of 2 (COMPLETED) is deleted. For example, if the default value is 30 and the batch program is run with the default value, the batch program would delete all the records that are more than 30 days old and are in completed status. Deletes purge error logs.	24 hours	30 minutes	24 hours
Cleanup Batch Schedule	Deletes the batch schedule records that are no longer needed after the default time specified.	30 minutes	30 minutes	24 hours
Cleanup Closed Transfers	Deletes all the closed transfer which are in either cancelled or completed status, and shipments related to them. Any closed transfer with an update date older than the Days to Hold Transfer Documents parameter value will be deleted.	24 hours	30 minutes	24 hours
Cleanup Completed UINs	Deletes completed UIN Detail records. A completed UIN is any UIN with a status of Removed from Inventory, Missing, Sold, Shipped to Vendor, or Shipped to Warehouse. Any UIN detail record with a complete status and update date at least X days in the past (where X is with system parameter Days to Hold Completed UINs) will be deleted.	24 hours	30 minutes	24 hours

Table 5-30 (Cont.) Cleanup Batches

Batch Name	Description	Default	Minimum Value	Maximum Value
Cleanup Customer Orders	Deletes all the fulfillment order records which are not in New or In Progress status and for which the update date has expired the purge_date by number of days more than Days to Hold Customer Order parameter value. Additionally, only those fulfillment orders will be deleted for which customer order ID and fulfillment order ID combination does not exist for any Transfer, Return, Purchase Order, and Warehouse delivery transaction.	24 hours	30 minutes	24 hours
Cleanup DSD and Purchase Orders	Deletes the Direct Store Delivery receiving. Any DSD record which is in Closed/ Cancelled status and which has a complete date older than Days to Hold Received Shipments is an eligible record for purge. In effect, a DSD record can be purged only if its associated PO records can be purged.	24 hours	30 Minutes	24 hours
Cleanup Invalid Users	Deletes invalid application users from data store for those user names that are not found in identity store.	12 Hours	30 Minutes	24 hours
Cleanup Invalid User Roles	Removes all expired user roles and orphaned user roles (roles that were deleted by removing a store) from the SIOCS system. The batch process finds user role assignments that have an end date that is at least X days in the past (where X is specified by the system parameter Days to Hold Expired User Roles), and deletes these expired role assignments. The users (excluding super users) with role assignments that have no matching store assignments (orphaned role assignments) are also deleted.	24 hours	30 Minutes	24 hours
Cleanup Inventory Adjustments	Deletes inventory adjustments records with a create date/timestamp older than Days To Hold Completed Inventory Adjustments parameter value.	24 hours	30 Minutes	24 hours

Table 5-30 (Cont.) Cleanup Batches

Batch Name	Description	Default	Minimum Value	Maximum Value
Cleanup Item Baskets	Purges item basket records (with status of cancelled or completed) based on the retention period. The retention period is specified by system configuration parameter- Days to Hold Item Basket.	24 hours	30 Minutes	24 hours
Cleanup Item Hierarchy	Purges all Item Hierarchies that are in deleted status.	24 hours	30 Minutes	24 hours
Cleanup Item Prices	Purges records which were expired or were marked as deleted based on the retention period. The retention period is specified by system configuration parameter Days to hold expired item price. Following are the rules defining records to be purged: • Regular Price Change: Has status of completed or deleted, effective date was X number of days in the past (relative to the specified date if specified). At any given time, at least one completed latest regular price must be retained for a store item. • Promotion Change: Has status of completed or deleted, and end date is number of days in the past (relative to the specified date if specified). • Clearance Change: Has status of completed or deleted, and end date is number of days in the past (relative to the specified date if specified).	24 hours	30 minutes	24 hours

Table 5-30 (Cont.) Cleanup Batches

Batch Name	Description	Default	Minimum Value	Maximum Value
Cleanup Items	This batch program deletes items with a status of Delete (D). There are two segments which do the following different tasks: 1. Validate if the Item should be deleted. 2. Delete item from all associated tables if all following validation checks are passed. • If SOH of item, item parent and item grandparent is 0. • If any transfers exist for item, item parent and item grandparent. • If any RTV exists for item, item parent and item grandparent. • If any Inventory adjustment exists for item, item parent and item grandparent. • If any Item Basket exists for the item. • If any Product Group exists for the item. • If any Stock Count exists for the item. • If any Store Order exists for the item. • If any Item Request exists for the item. • If any Direct Store Delivery exists for the item. • If any Warehouse Delivery exists for the item. If the validations checks are met, the records related to the item which is marked for the purge action are deleted.	24 hours	30 minutes	24 hours
Cleanup Notifications	Deletes notifications. The retention period is specified by system configuration parameter Days to Hold Notifications.	24 hours	30 minutes	24 hours
Cleanup Price Change Worksheet	This batch process deletes price change worksheet records from the staging table which are in Rejected/ Completed status. Any price change record with an effective date/timestamp older than Days To Hold Price Changes parameter value will be deleted.	24 hours	30 minutes	24 hours

Table 5-30 (Cont.) Cleanup Batches

Batch Name	Description	Default	Minimum Value	Maximum Value
Cleanup Price History	This batch process deletes price histories. At least a minimum set of historical prices are maintained for an item/store.	24 hours	30 minutes	24 hours
Cleanup Product Areas	Deletes the product areas records that are no longer required if the status is CANCELED(3) and the basked id is not into the picks for store fulfillment orders table. Days to Hold Areas will determinate the number of days that product areas can be kept in the database.	24 hours	30 minutes	24 hours
Cleanup Related Items	Deletes the related items for which the end date has expired for more than Days To Hold Related Items system configuration value.	24 hours	30 minutes	24 hours
Cleanup Resolved UIN Problems	Deletes resolved UIN exception records. UIN exception records with status of resolved and resolved date is at least X days in the past (where X is system parameter Days to Hold Resolved UIN Exceptions) are deleted.	24 hours	30 minutes	24 hours
Cleanup RFID	Deletes RFIDs which is not present in location. The retention period is specified by system configuration parameter Days to Hold RFID.	24 hours	30 minutes	24 hours
Cleanup Sales Posting	This batch process deletes the Point-of-Service transaction from the Oracle Retail Xstore Point of Service transaction staging table. It reads the Days to Hold Sales Posting and Days to Hold Failed Sales configuration parameters and all the transactions which are present beyond the configuration parameter are deleted. It also purges the POS transaction logs for the request IDs that are in processed status.	24 hours	30 minutes	24 hours
Cleanup Shelf Adjustments	Purges shelf adjustment records (with status of completed) based on the retention period. The retention period is specified by system configuration parameter- Days to Hold Shelf Adjustment List.	24 Hours	30 minutes	24 Hours
Cleanup Shelf Replenishment	Deletes shelf replenishment lists which are in Completed/Cancelled state. Any shelf replenishment list record with a status date/timestamp older than Days To Hold Shelf replenishment parameter value will be deleted.	24 Hours	30 minutes	24 Hours

Table 5-30 (Cont.) Cleanup Batches

Batch Name	Description	Default	Minimum Value	Maximum Value
Cleanup Staged Messages	This batch finds integration staging records that are marked as processed or deleted, and update date is at least X days in the past (where X is the system parameter Days to Hold Completed Staging Records).	24 Hours	30 minutes	24 Hours
Cleanup Staged Initial Data Load	This batch will purge data from all “IDLS” and “ERR_IDLS” tables. This batch is designed to be run as ad-hoc only, therefore a schedule is not available for this batch.			
Cleanup Stock Counts	This batch process deletes stock counts which are in Completed/ Cancelled status. Any stock count with a schedule date/timestamp older than Days To Hold Completed Stock Counts parameter value will get deleted.	24 Hours	30 minutes	24 Hours
Cleanup Store Item Stock History	Deletes store item stock history records from STORE_ITEM_STOCK_HISTORY table. Records with printed date older than Days to Hold Transaction History system configuration value will be deleted.	24 Hours		
Cleanup Temporary UINs	This batch process deletes temporary UIN detail records. UIN detail records with no status and update date is at least X days in the past (where X is system parameter Days to Hold Temporary UINs).	24 Hours	30 minutes	24 Hours
Cleanup Vendor Returns	This batch process deletes vendor returns which are in closed or completed status. Any vendor return record with a closed date/timestamp older than Days to Hold RTV system configuration value will be deleted.	24 Hours	30 minutes	24 Hours
Cleanup Closed Warehouse Containers	This batch job deletes fully received warehouse to store deliveries (a fully received delivery is one where all cartons are in received status and not marked as damaged or missing) whose received date of delivery is older than X days (Specified in Days to hold closed warehouse containers system configuration parameter)			

Cleanup Closed Warehouse Containers

This batch job deletes fully received warehouse to store deliveries (a fully received delivery is one where all cartons are in received status and not marked as damaged or missing) whose

received date of delivery is older than X days (Specified in Days to hold closed warehouse containers system configuration parameter).

This batch job can only be run in Adhoc Mode either via the EICS Job Admin screen or by calling the Execute Batch REST API to run the batch execution at any desired time.

In order to delete the closed containers and the associated delivery for a particular data set id on the EICS Job Admin screen, user must enter the transfer delivery id as the Data Set ID.

Batch Job Definition Name

WarehouseClosedContainers_PurgeJob

Batch Job Parameters

<input_date> Optional, if not specified defaulted to current timestamp. System batch input date format is used for parsing input date if specified.

<Data Set Id> Optional, if specified, the batch looks for the data set id (transfer delivery id) for processing.

Key Tables

Table 5-31 Key Tables for Delete Closed Warehouse Containers Batch

Table	Select	Insert	Update	Delete
CONFIG_SYSTEM	x			
TSF_DELV	x			x
TSF_DELV_CARTON	x			x
TSF_DELV_LINE_ITEM	x			x
TSF_DELV_LINE_ITEM_UIN	x			x
TSF_DELV_CFA	x			x
TSF_DELV_CDA	x			x
TSF_DELV_CARTON_CFA	x			x
TSF_DELV_CARTON_CDA	x			x
TSF_DELV_LINE_ITEM_ATT	x			x

Restart/Recovery

This batch can be re-run by restart batch job after any issues are resolved.

To Invoke the batch job using the Execute Batch Rest Service, refer to the REST Service Batch section of this guide.

Table 5-32 Execute Batch REST Request Example

API URL	Operation	Description	Example Request Payload
https://<siocs-lb> /siocs-int-services/api/batches	POST	Submit batch job for immediate execution.	{ "batchName": "WarehouseClosedContainers_PurgeJob” }

Table 5-32 (Cont.) Execute Batch REST Request Example

https://<siocs-lb> /siocs-int-services/api/batches	POST	Submit batch job for immediate execution for the particular data set id (TSF_DELIVERY_ID)	{ "batchName": "WarehouseClosedContainers_PurgeJob", "parameterId": 9999 }

System Process Batches

Processing of critical alerts, data migration for storage, and closure of old records is critical for the continued efficient and normal processing of the system.

These clean-up jobs are restricted and enabled by default.

Customers can configure number for day to retain information prior to closure in database via System Admin Parameters.

Table 5-33 System Process Batches

Batch Name	Description	Default	Minimum Value	Maximum Value
DSD Receiving Closure	This batch program looks for all the open vendor deliveries whose expected date added to store parameter "Auto Close Days after Expected Date" is before today and automatically confirms all the vendor deliveries.	24 hours	30 minutes	24 hours
Fulfillment Order Pick Reminders	This batch process generates notifications for fulfillment order picks for which status is new or in progress and create date has expired by X number of minutes (specified in system configuration “Minutes To Hold Open Customer Order Pick Before Sending Notification”).	24 hours	30 minutes	24 hours
Fulfillment Order Reminders	This batch process generates notifications for fulfillment orders for which create date has expired by X number of minutes (specified in system configuration “Minutes To Hold New Customer Order Before Sending Notification”).	24 hours	30 minutes	24 hours
Item Price To History	This batch writes the active item price records into item price history table. After the active item prices are recorded in the item price history table, the batch updates the ITEM_PRICE table statuses as completed for these records.	24 hours	30 minutes	24 hours

Table 5-33 (Cont.) System Process Batches

Batch Name	Description	Default	Minimum Value	Maximum Value
Product Group Schedule Closure	This batch program searches for all open product group schedules that have ended date before today (or user specified date) and change the product group schedule status to closed.	24 hours	30 minutes	24 hours
Return Not After Date Alert	This batch process warns users a number of days in advance that the RTV/RTW is about to reach the Not After date and must be dispatched. The value for the number of days of advance warning is configurable using the system's administration screens.	24 hours	30 minutes	24 hours
Transfer Close	This batch program looks for all the open transfers which have passed their not after date and are in valid state for closure.	24 hours	30 minutes	24 hours
Transfer Delivery Auto Receive	This batch auto receives the transfer deliveries to stores if delivery option is defined as date driven in store configuration. If the Auto Receive store parameter is set to Date Driven, then the batch auto-receives all deliveries that are in New and In Progress status and whose Ship Date added to the Auto Receive Number of Days is less than the current date.	24 hours	30 minutes	24 hours
Transfer Delivery Close	This batch program looks for all the open transfer deliveries and auto con-firms all the transfer deliveries based on the store parameter "Auto Close Receipt". When the parameter value is "0", close the deliveries at the end of day today and when value is "x" close the deliveries at the end of "x" days starting from today.	24 hours	30 minutes	24 hours
Transfer Not After Date Alert	This batch process generates email alerts for any pending transfer requests with not after date coming up within number of days specified in the system parameter "Days to Send Email Alert Before Not After Date for Transfer Requests".	12 hours	30 minutes	24 hours
Transfer Overdue	This batch process generates notification for dispatched transfers which have not been received after X number of days (specified in system configuration “Days Shipped Delivery Overdue Notification”).	24 hours	30 minutes	24 hours
Vendor Return Closure	This batch program looks for all the open vendor returns which are in valid state (Closed /Rejected) for closure.	24 hours	30 minutes	24 hours

Batch Job Administration

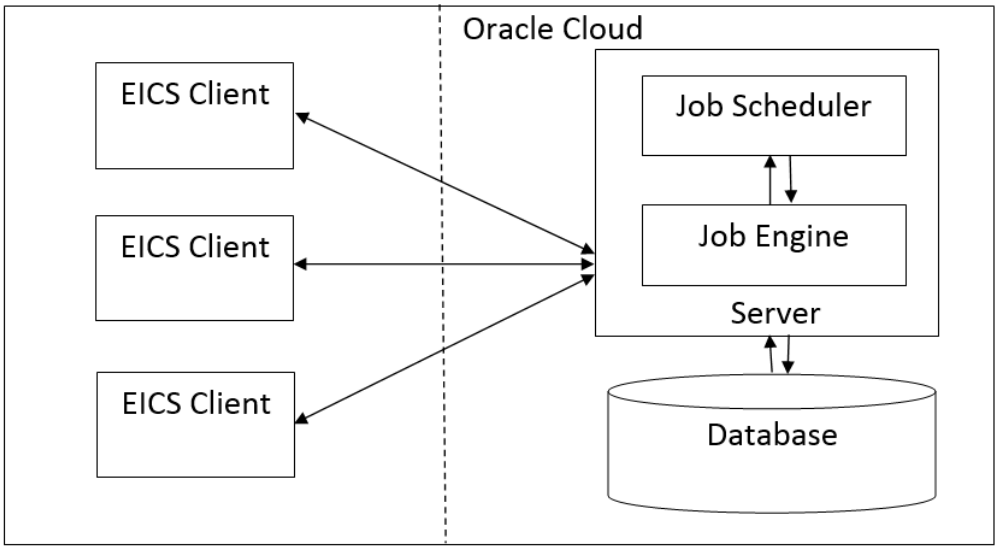
This provides information about the processing and operating details of batch job administration and operations, and covers the following topics:

- Job Administration
- Job Scheduler

The batch jobs are installed and configured by the EICS application installer. The batch processes are designed to process large volume of data. The batch jobs can be scheduled as per the retailer's choice to be executed on specific intervals on the SIOCS GUI.

The following diagram illustrates the high-level architecture of the current batch processing implemented for the EICS Application Server.

Figure 5-1 EICS Application Server Batch Processing Architecture



- **EICS Client - Server Communication**

  EICS client provides an option to the retailer to run the batch jobs on demand. This call to the server is made via HTTP REST service call. The batch job selected, and the parameter selected by the user are set on the request and sent to the server on this call. The server handles this request and invokes a start job on the batch engine for the respective job.

- **Job Engine**

  The job engine manages the state of a running job and guarantees the execution of each step defined for the job. The call made from the client or job scheduler is passed onto the job engine to start a new instance of the batch job. Users also have an option to stop the running job or to restart any particular job which failed during the processing.

- **Job Scheduler**

  This feature of EICS allows a retailer to schedule the batch jobs to run at a specific time interval. Each batch job will initially execute at a pre-configured schedule interval. Retailers

will have an option to enable or disable specific batch job schedules as well. For more information, please refer to Job Scheduler Section below.

> **Note:**
>
> A few of the jobs which are system required will not be available for the retailer to change the schedule interval or disable them. These jobs will always be enabled and can be managed only by cloud admin.

Job Administration

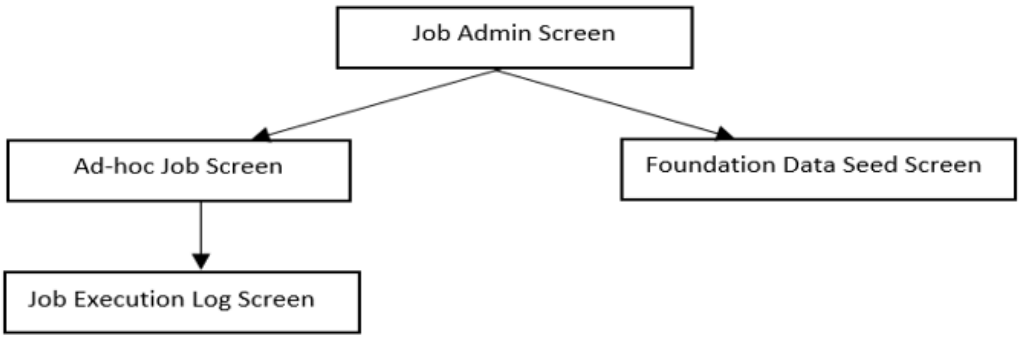
SIOCS Job Admin is a web application that provides the runtime and GUI for managing batch jobs.

SIOCS provides an independent user interface for executing and scheduling of the batch jobs. These user interface screen will facilitate users to perform following operations:

- Execute Ad hoc Jobs
- Manual Data Seeding Importer Jobs
- View the Job Execution Log
- View the list of Jobs executed
- Schedule Jobs for execution on specific intervals

The following diagrams show the Batch Admin and relevant GUI components:

Figure 5-2 Batch Admin and GUI Components



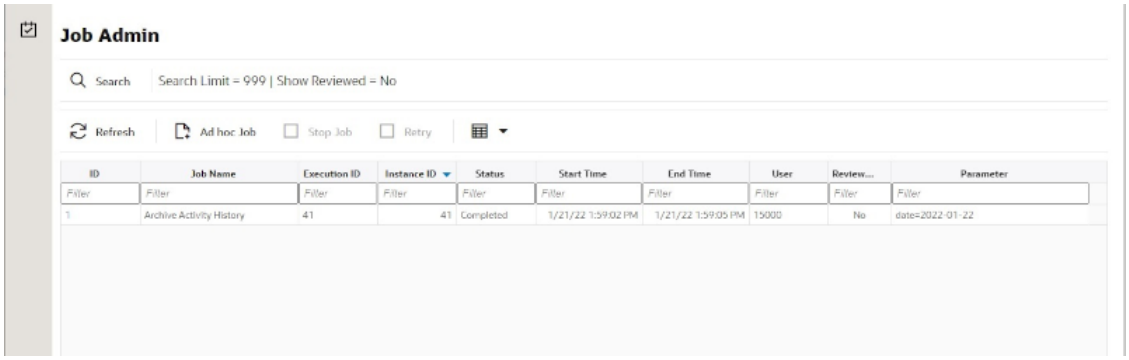
Job Admin Screen

This screen views the list of the job that have been instantiated on the server. User can filter the list of the job loaded on screen on Job Status, Reviewed and Search Limit. The default filter will load the list of the failed job when the screen is initially loaded.

User requires appropriate permission to access this screen on SIOCS. Navigation path for the Job Admin screen is:

Admin/Technical Maintenance/Job Admin

Figure 5-3 Job Admin Screen



Screen Options

- **Search**

  Allows user to filter the list of the batch jobs instantiated by user and the scheduler.

- **Refresh**

  This option refreshes the list loaded on the screen; call is made to server to load the list of batch jobs with current state.

- **Ad hoc Job**

  This option refreshes the list loaded on the screen; call is made to server to load the list of batch jobs with current state.

- **Data Seed**

  This option navigates user to Data Seed Job Launcher screen. User can start a new data seed job from this screen by setting the required parameters and data seeding options on the job. User will require appropriate access permission to view this option on the screen.

- **Stop Job**

  This option allows user to stop a running job. This option is available only when selected job is in running state that is, STARTED and STARTING.

Table Information

- **Execution ID**

  This column displays the job execution identifier generated by the job engine.

- **Job Name**

  This column displays the name of the batch job.

- **Instance ID**

  This column displays the job instance identifier generated by the job engine.

- **Status**

  This column displays the current status of the batch job.

- **Start Time**

  This column displays the time when job was started.

- **End Time**

This column displays the time when job was completed. If the job is currently in running state, the end time will not be available.

- **User**

  This column displays the user name of the user who started the job. The job started by the scheduler will have server user name in this column.

- **Reviewed**

  This column denotes if user has reviewed this job.

- **Parameter**

  This column displays the parameter if entered while starting the batch job.

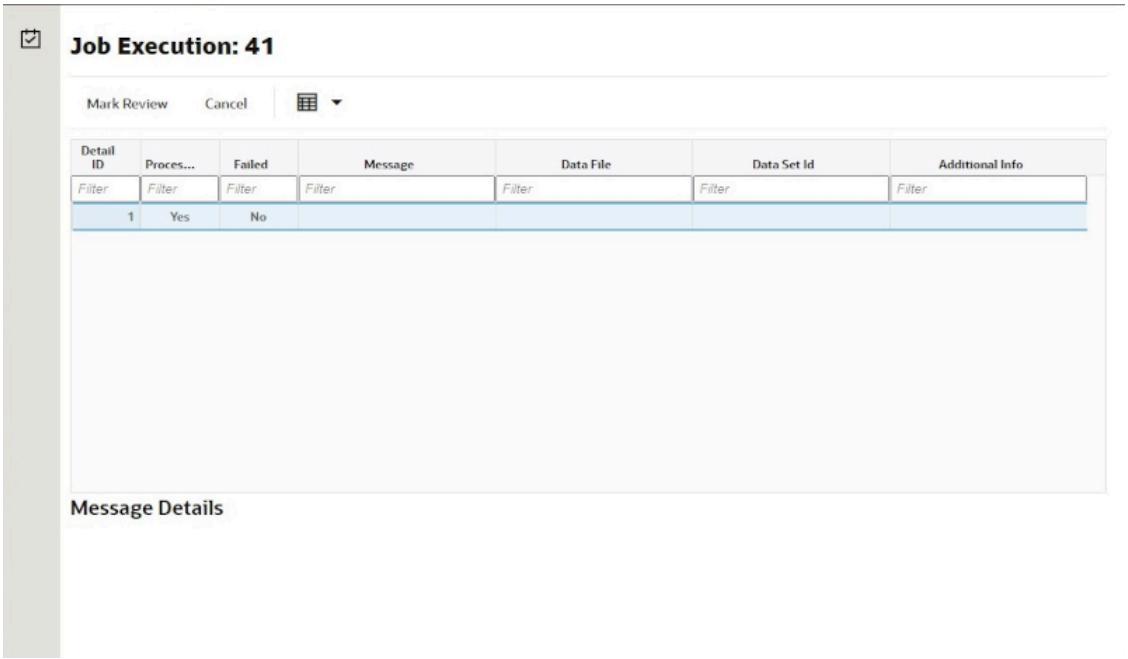
Job Execution Log Screen

This screen displays the execution details for the batch job. Users can navigate to this screen by clicking **Job Execution ID** on the Job Admin screen. This screen is designed to view the error logs for the batch jobs and to mark the job as reviewed.

Review flag denotes the selected job has been reviewed by the user. By using this, the user can keep a track of which of the failed jobs have already been reviewed and take necessary actions as required.

The execution detail table load the entries of the execution record if exist for the select batch job.

Figure 5-4 Job Execution Screen



Screen Options

- **Mark Review**

  This option allows user to mark the batch job as review. Once the job is marked as review user is navigated back to the Job Admin screen.

- **Cancel**

  This option allows user to navigate back to the Job Admin screen.

Table Information

- **Detail ID**

  This column displays the execution identifier generated by the job engine.

- **Processed**

  This column denotes if the execution record for the job was successfully processed.

- **Failed**

  This column denotes if the execution records was failed during the processing.

- **Message**

  This column displays the message from the server to viewing the further details on the job status. This column will be empty for the execution records which are been successfully processed.

- **Data File**

  This column displays the file name for which the execution record was created. This column holds data only if the job details with file import processing for example, Retail Sales Audit Import Job.

- **Additional Info**

  This column holds the addition info if any for the execution record.

Message Details

This section in non-editable and displays the detailed explanation for the job failure. This section will display respective data on selection job execution record. The log traces of the exact point of failure shall be printed on this section of the screen.

Job Launch Screen

This screen allow user to start a new instance for a job. Select the batch job and enter the parameter, if required, to start job.

The job launcher screen has the following categories:

- Ad hoc Job
- Start an Ad hoc Job
- View Details for Job
- Stop a Running Job

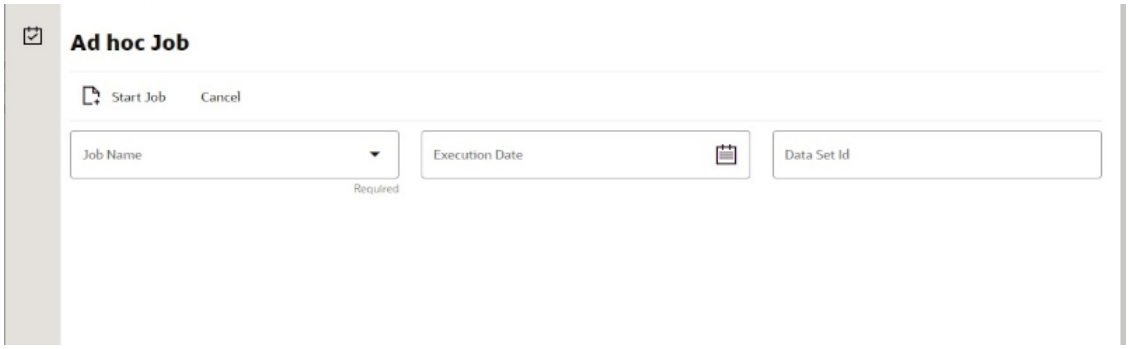
Ad hoc Job

This screen allows user to run an ad hoc job. All the job which are either operational or purge job can be started from this screen. All the batch job will not support the execution date and store identifier parameters. The job for which store level processing can be performed will consider these parameters. User will not receive an error or confirmation if the parameters are not valid for the selected job.

User will require appropriate data permissions to select and run a particular job. Navigation to this screen is as follows:

Admin/Technical Maintenance/Job Admin /Ad hoc Job

Figure 5-5 Ad hoc Job Screen



Screen Options

- **Start Job**

  This option allows user to start a new instance of the selected job. When the user selects this option, a call is made to server to start a selected batch job. Server internally makes a call to Batch Operator specifying the Job Name and the parameter if any.

- **Cancel**

  This option navigates user back to Job Admin Screen.

Menu Options

- **Job Name**

  Allows user to select the job to be started. This is a required field on this screen. The user will be able to view only those jobs in the drop-down list which are permitted to the user via data permissions.

- **Store ID**

  Allow user to enter the store identifier if the job is expected to be executed for specific store. Not all job accepts store identifier as parameter. For the jobs which doesn't take store identifier as parameter for execution shall ignore this parameter.

- **Execution Date**

  Allows user to enter the execution date for the job, if not entered the job will consider the current date for the execution. Not all job accepts execution date as parameter. For the jobs which doesn't take execution date as parameter for execution shall ignore this parameter.

- **Data Set Id**

  The Data Set Id is optional parameter for batch job to process the specified data set id.

Start an Ad hoc Job

1. Login to SIOCS and navigate to Job Admin screen.
2. Click on Ad hoc Job Menu to navigate to Ad hoc Job screen
3. Select the required Job from the drop-down menu.

4. Set the required job parameter for the job, that is, Store ID and/or Execution Date.

> **Note:**
>
> Job Parameter doesn't apply to all available batch jobs, for the job which do not take any parameter input will simply ignore the value in case entered.

5. Click on Start Job to start the new instance of job.
6. The new execution record will be populated on the Job Admin Screen List.

View Details for Job

1. Login to SIOCS and navigate to Job Admin screen.
2. Click the Job Execution Id to navigate to details.
3. To navigate back to Job Admin screen, click Cancel.
4. To mark job execution reviewed, click Mark Reviewed.

Stop a Running Job

1. Login to SIOCS and navigate to Job Admin screen.
2. Select the job which is currently in running state that is, STARTED or STARTING.
3. Click on Stop Job to stop the execution of the running job.
4. Prompted to confirm if the job needs to be stopped. Click Yes to continue.
5. Request is sent to job engine to stop the instance of the job. In case the job is already completed or stopped appropriate message shall be displayed to user.
6. Refresh the list on the screen to view the changes.

Job Scheduler

EICS Job Scheduler allows user to schedule the job available to run under Ad hoc Job screen.

The key features of Job Scheduler are as follows:

- **Interval Based Schedule**

  User can schedule the job to run on interval basis. User will be provided below specified interval to be set for each job for execution.

Table 5-34 Interval Based Schedule

Interval	Detail	Execution Hour
30th Minute	Job execution will be every 30th minute of the hour starting from 01:00.	01:00, 01:30, 02:00, 02:30, 03:00, 03:00 … 23:00, 23:30, 00:00, 00:30.
1 Hour	Job execution will be every 1 hour starting from 01:00.	01:00, 02:00, 03:00, 04:00, 05:00 … 22:00, 23:00, 00:00.

Table 5-34 (Cont.) Interval Based Schedule

Interval	Detail	Execution Hour
2 Hours	Job execution will be every 2nd hour of the day starting from 01:00.	01:00, 03:00, 05:00, 07:00, 09:00 … 21:00, 23:00.
3 Hours	Job execution will be every 3rd hour of the day starting from 01:00.	01:00, 04:00, 07:00, 10:00, 13:00, 16:00, 19:00, 22:00.
4 Hours	Job execution will be every 4th hour of the day starting from 01:00.	01:00, 05:00, 09:00, 13:00, 17:00, 21:00.
6 Hours	Job execution will be every 6th hour of the day starting from 01:00.	01:00, 07:00, 13:00, 19:00.
8 Hours	Job execution will be every 8th hour of the day starting from 01:00.	01:00, 09:00, 17:00.
12 Hours	Job execution will be every 12th hour of the day starting from 01:00.	01:00, 13:00.
24 Hours	Job execution will be every 24th hour of the day starting from 01:00.	01:00.
1 Week	Job execution will be every Monday at 1AM.	
2 Weeks	Job execution will be every other Monday at 1AM starting on the 2nd Monday of the year.	

- **Schedule Management**

  User will have an option enabling or disabling the scheduler feature for a job at any given point of time. Once disabled the auto execution on the future scheduled interval will be stopped.

- **Scheduler Configuration**

  User will have option of configuring the scheduler related configuration.

Table 5-35 Batch Job Schedules

Batch Job	Interval
Auto Replenish Capacity	24 hours
Auto Ticket Generate	24 hours
Auto Ticket Print	24 hours
Generate Problem Line Stock Count	24 hours
Generate Unit Amount Stock Count	24 hours
Generate Unit Stock Count	24 hours
Item Basket Maintenance	24 hours
Item Price ICL Import Job	30 minutes
Retail Sales Audit Import	24 hours

Table 5-35 (Cont.) Batch Job Schedules

Batch Job	Interval
Shelf Replenishment Closure	24 hours
Stock Count Unit And Amount Snapshot	24 hours
Store Order Auto Approve	12 hours
Store Order Auto Cancel	24 hours
Store Order Auto Generate	24 hours
Third Party Pricing Import	30 minutes
Third Party RFID Import	30 minutes

Job Scheduler Screen

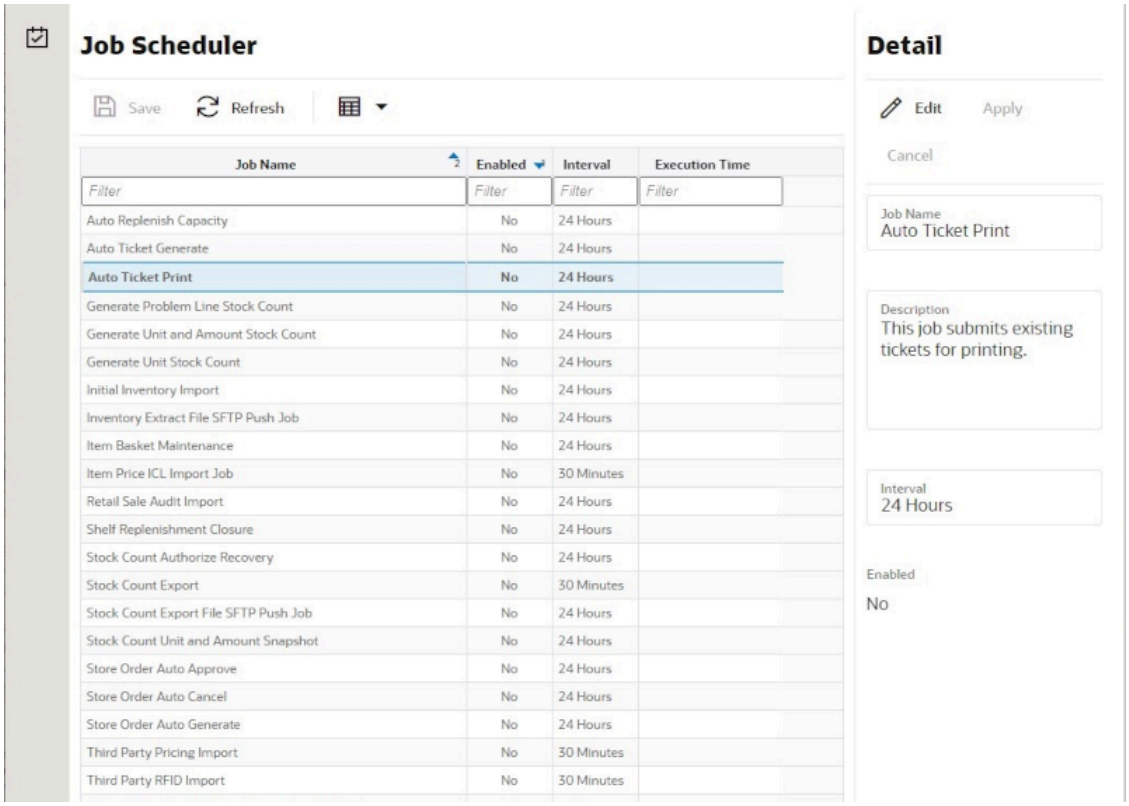
This screen allows user to manage the schedules for the batch jobs. The list of job available to be scheduled will be sort by the enabled flag followed by Job Name alphabetically that is, the jobs for which the scheduling is enable will be on the top of the list sorted by the Job Names in alphabetical order.

User can edit the batch job and perform following operations:

1. Enable / Disable a schedule.
2. Setting the execution interval for a batch job.

User will require appropriate authorization to access this screen. Retailer will not be able to view or disable the batch jobs which are system required. This batch jobs will be accessible to Cloud Admin user will have a predefined schedule. Navigation to this screen is as follows: Admin/Technical Maintenance/Job Scheduler

Figure 5-6 Job Scheduler Screen



Screen Options

- **Save**

  This option allows user to save the changes made to the job schedules. User can make changes to multiple schedules in edit mode and apply the changes, however the changes will take affect only when the save operation is performed.

- **Refresh**

  This option refreshes the list of values on the screen.

- **Edit**

  This option will be enabled only when a schedule is selected to be edited and user have privileges to edit the job schedule. This option takes user to the edit mode in which user can change the execution interval for the job and enable or disable the schedule.

- **Apply**

  This option will only apply the changes which are been made during edit mode. This option does not save the changes, it only applies the changes for the job and exit the edit mode. Click the Save option to save the changes.

- **Cancel**

  This option is enabled only when user is in edit mode. This option allows user to discard any changes made to the schedule on edit mode or to exit the edit mode.

Menu Options

- **Interval**

Users have the option to select the appropriate interval for the batch job execution. Depending on the selected interval, server calculates the execution time for the batch job and runs it according. The base time for calculating the execution time on basis of the interval is 01:00 AM. For the store based job this is as per the store time zone and for system jobs this is as per the server time zone.

Option for selecting the execution interval are as follows:

1. 30 Minutes - Runs every 30 minutes.
2. 1 Hour - Runs every hour.
3. 2 Hours - Runs every second hour of the day.
4. 3 Hours - Runs every third hour of the day.
5. 4 Hours - Runs every fourth hour of the day.
6. 6 Hours - Runs every sixth hour of the day.
7. 8 Hours - Runs every eighth hour of the day.
8. 12 Hours - Expected to be executed twice a day.
9. 24 Hours - Expected to be executed only once a day.

- **Enabled**

Table Information

- **Job Name**

  This column displays the name of the job.

- **Enabled**

  This column denotes whether a job is current scheduled.

- **Interval**

  This column denotes the interval of the job execution.

- **Execution Time**

  This column denotes the next execution time for the batch job. This column will not contain data for the jobs which are been disabled.

Configuring a Job Schedule

1. Login in SIOCS and navigate to the Job Scheduler screen.
2. Select the desired job from the list.

> **Note:**
>
> If the Edit option is not available in the menu section, the user is not authorized to edit the batch job. Kindly assign appropriate user group or contact system admin for desired changes.

3. Edit the Job details on right panel.
4. Set required interval for execution and **Enabled** to Yes.
5. Click **Apply** to exit the edit mode.

6. For configuring multiple jobs on a go, repeat Step 2 for each job.
7. Click **Save** to save the current changes made on the screen.

Disabling a Job Schedule

1. Login in SIOCS and navigate to Job Scheduler screen.
2. Select the job from the list which needs to be disabled.
3. Edit the Job details on right panel.
4. Set **Enabled** as No.
5. Click **Apply** to exit the edit mode.
6. For disabling multiple jobs on a go, repeat Step 2 for each job.
7. Click **Save** to save the current changes made on the screen.

6

Technical Maintenance Screens

This section covers the following topics:

- Credential Administration
- External Service Administration
- File Transfer Service
- Job Admin
- Job Scheduler
- MPS Staged Message
- MPS Work Type
- DCS Work Type
- Operational Issues
- POS Transaction Resolution
- Sequence Administration
- Integration Dashboard

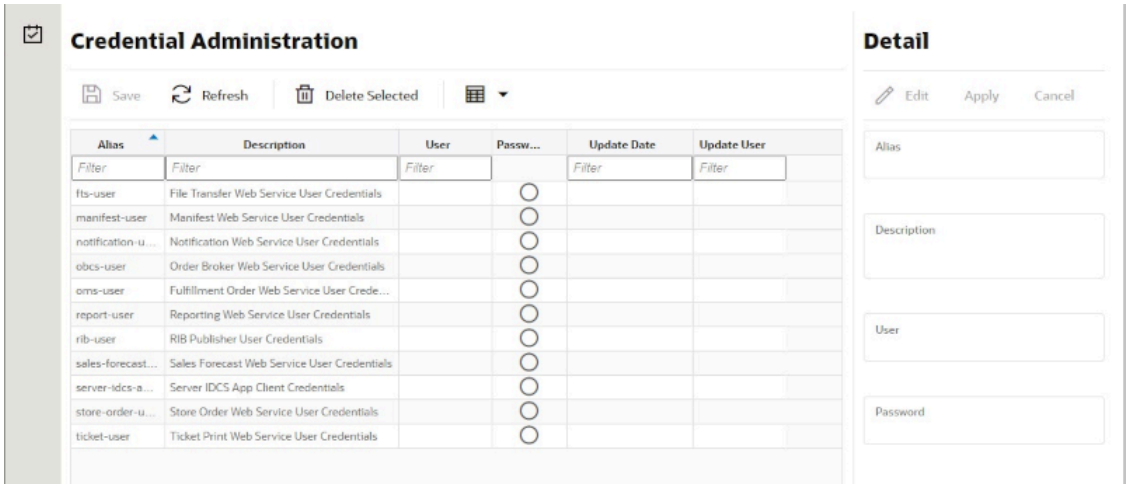
Credential Administration

The Credential Administration screen is used to setup integration credentials to connecting with external/third party systems.

To access this screen, user need to be assigned followings:

- **IDCS or OCI IAM app roles**: <SIOCS Primary APP>.admin_users
- **Security Permission**: Access Credential Administration

Figure 6-1 Credential Administration



List Buttons

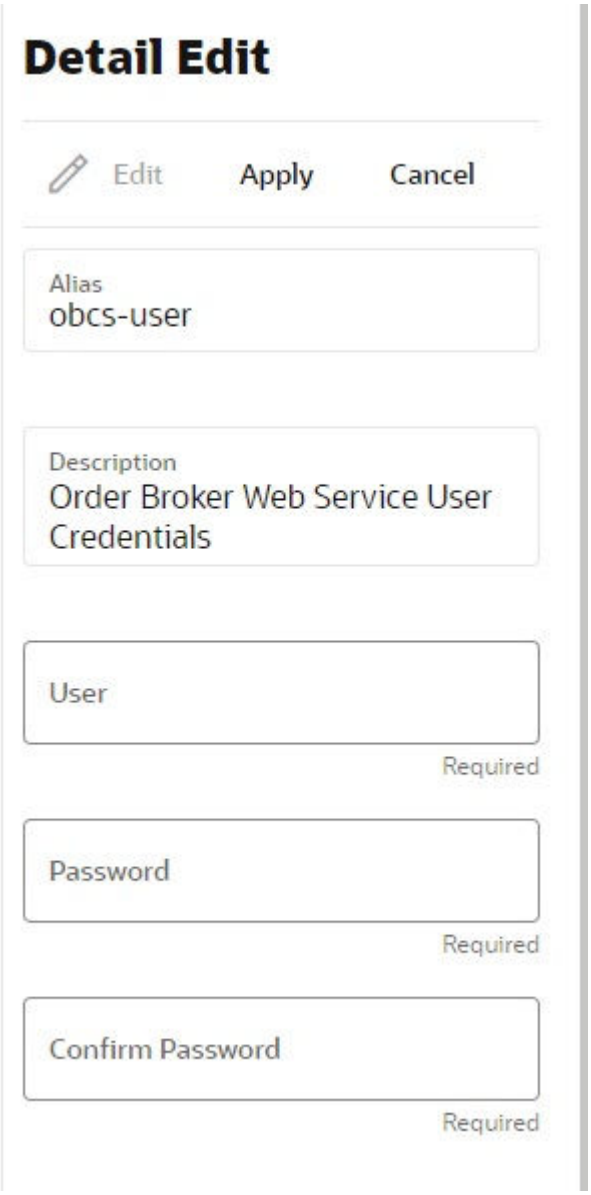
- **Save**: Persists currently altered information.
- **Refresh**: Refreshes the screen with currently persisted information.
- **Delete Selected**: Deletes the user and password value for the records selected from database.
- **Filter**: Enabled are disabled the filtering row in the table.

List Attributes

- **Alias:** The alias for the user.
- **Description**: The description of the user.
- **User**: The user name.
- **Password**: Displays whether a password has been set currently for the user.
- **Update Date**: The last update date.
- **Update User**: The user who updated the record last.

Detail Panel

Figure 6-2 Detail Panel In Edit Mode



Detail Buttons

- **Edit**: Places the detail information displayed into edit mode.
- **Apply**: Applies the current data to the credential admin record and updates the list information.
- **Cancel**: Places the panel back into display mode without applying the information.

Detail Attributes

- **Alias**: Shows the alias for the user. Pre-defined read only field.
- **Description**: displays the user description. Pre-defined read only field.
- **User**: Field to enter user name.
- **Password**: Field to enter password for the user.
- **Confirm Password**: Field to enter password one more time in order to make sure the user has entered the right password.

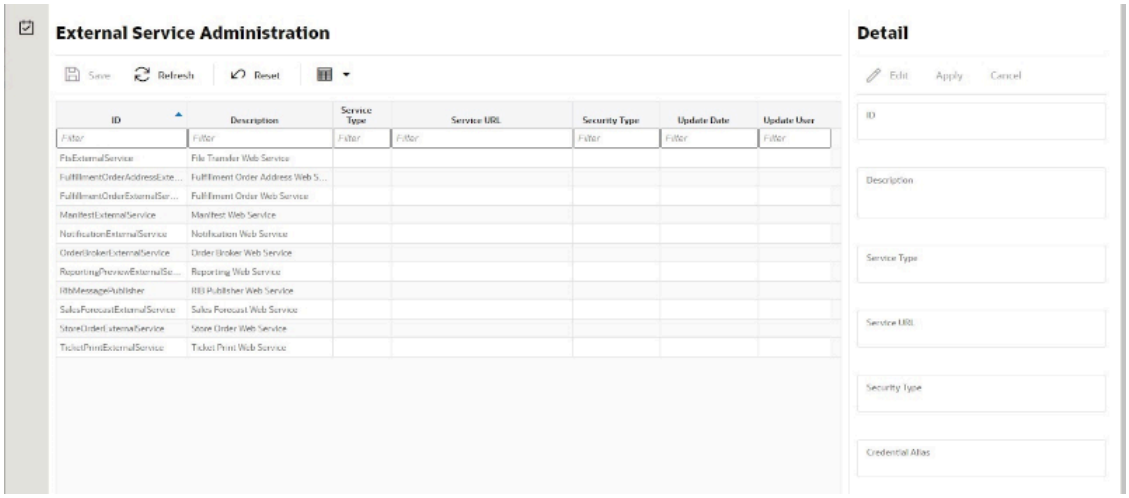
External Service Administration

The External Administration screen is used to setup integration information to connect with external/third party systems, such as specific the URL for external web service connection.

To access this screen, user need to be assigned followings:

Security Permission: Access Credential Administration

Figure 6-3 External Services Administration



List Buttons

- **Save**: Persists currently altered information.
- **Refresh**: Refreshes the screen with currently persisted information.
- **Reset**:
- **Filter**: Enabled are disabled the filtering row in the table.

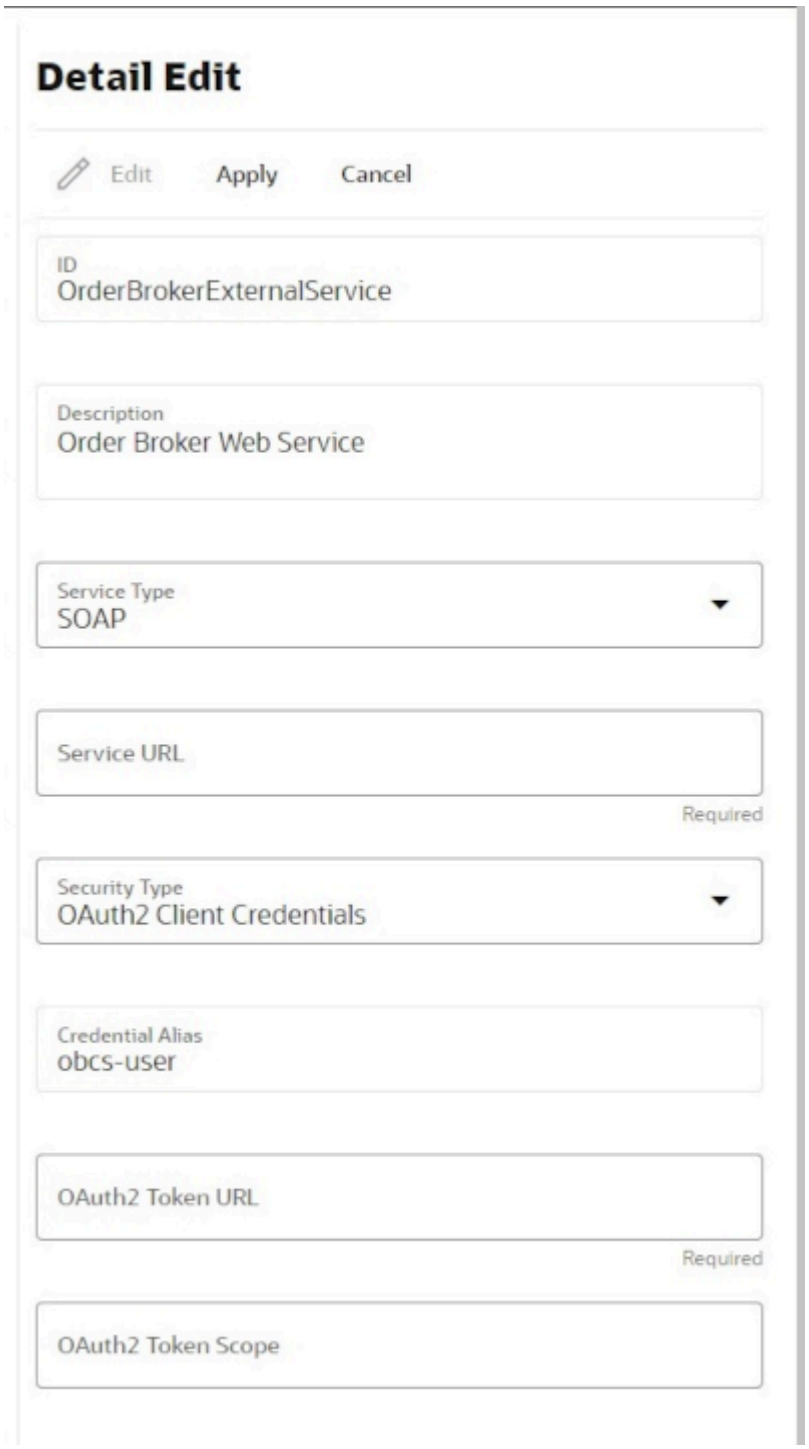
List Attributes

- **ID:** An identifier for the external service.
- **Description**: A description of the external service.
- **Service Type**: The type of web service (for example, SOAP, ReST).

- **Service URL**: The URL of the service location to access.
- **Security Type**: The type of security the service uses.
- **Update Date**: The last update date.
- **Update User**: The user who updated the record last.

Detail Panel

Figure 6-4 Detail Panel In Edit Mode



Detail Buttons

- **Edit**: Places the detail information displayed into edit mode.
- **Apply**: Applies the entered data to the record and updates the list information.
- **Cancel**: Places the panel back into display mode without applying the information.

Detail Attributes

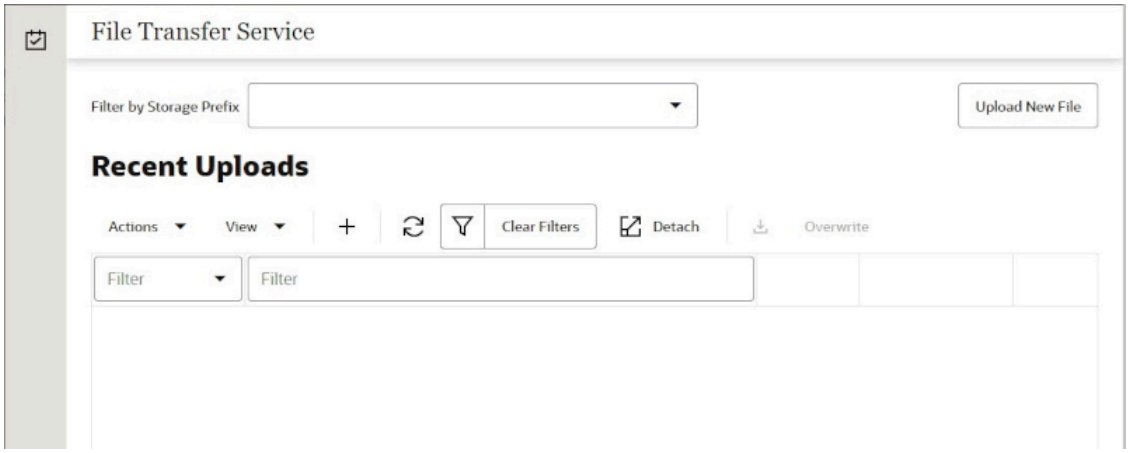
- ID: An identifier for the external service.
- **Description**: A description for the external service.
- **Service Type**: The type of service.
- **Service URL**: The URL used to connect to the service.
- **Security Type**: The type of security associated with the service. Selection will lead to more fields to populate.

File Transfer Service

The File Transfer Service screen allows the user to upload data files into a location within EICS to be processed. This supports access to Next Gen Cloud Services object storage through the file transfer service. The UI allows users to browse previous uploads, upload new files, overwrite previously uploaded files, and download files.

Security Permission: Access File Transfer Service

Figure 6-5 File Transfer Service



Screen Elements

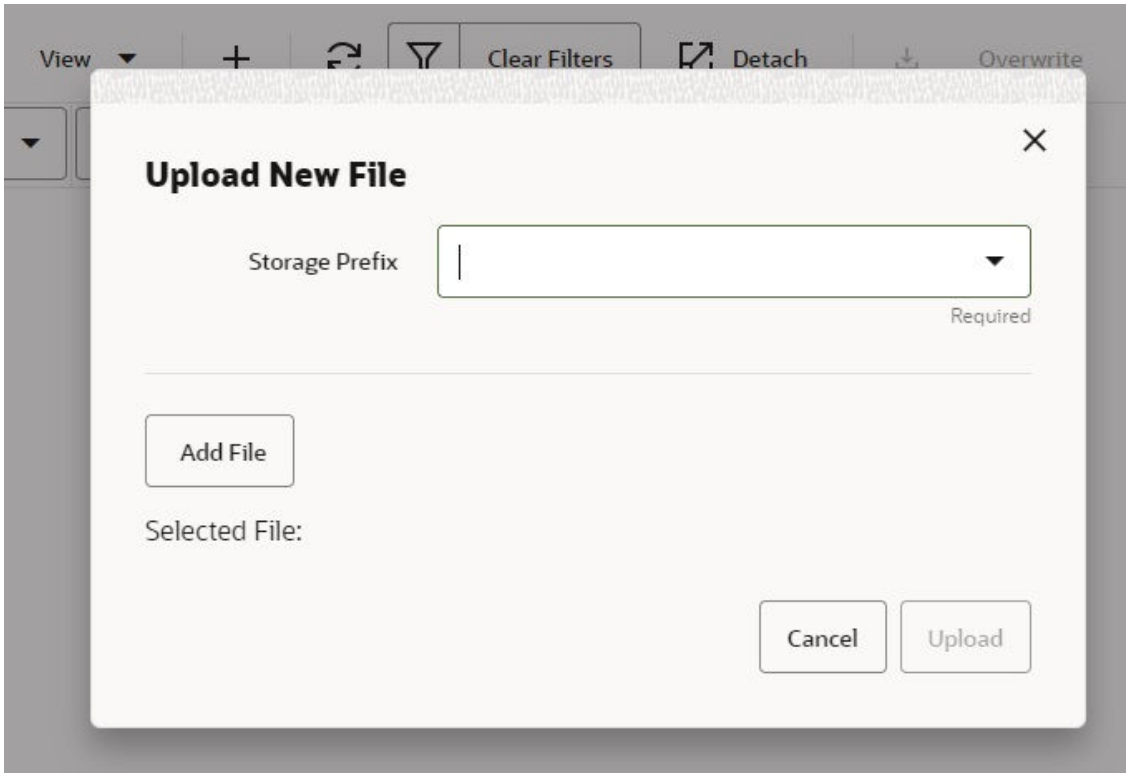
- **Filter By Storage Prefix**:
- **Upload New File**:
- **Refresh**:
- **Detach**:

List Attributes

- **Unknown**:

Detail Panel

Figure 6-6 Detail Panel in Edit Mode



Detail Components

- **Storage Prefix**:
- **Add File**:
- **Upload**:
- **Cancel**:

Job Admin

Documentation for the Batch Job Administration screen can be found in the Batches chapter within this guide.

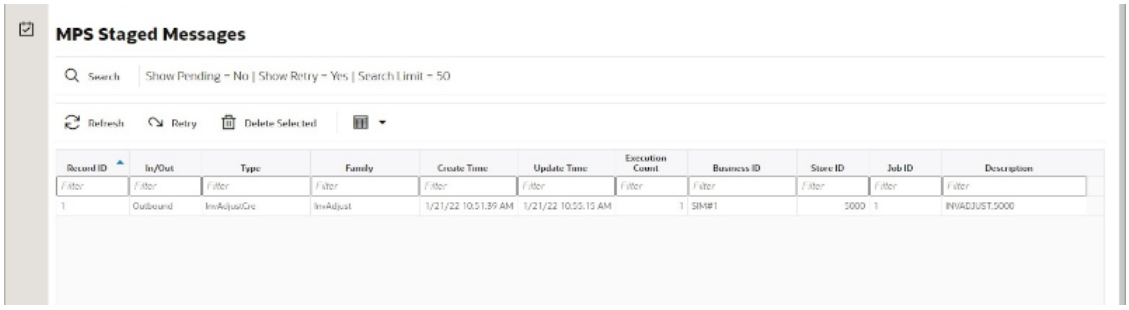
Job Scheduler

Documentation for the Batch Job Scheduler screen can be found in the Batches chapter within this guide.

MPS Staged Message

The MPS Staged Message administration form is used to monitor the messages in the Message Processing System queue. Each message contains the detailed information being transmitted between EICS and an external system.

Figure 6-7 MPS Staged Message



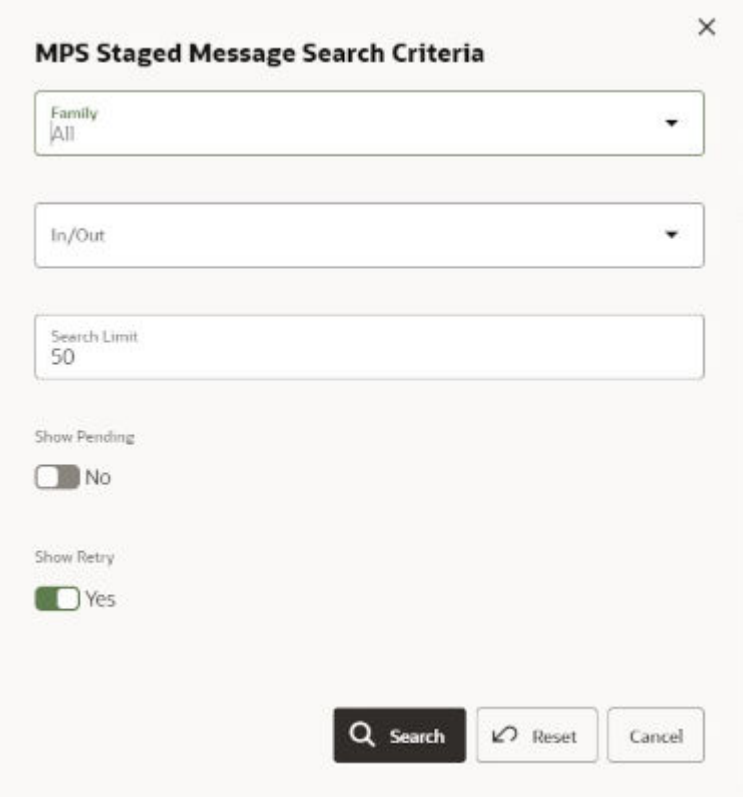
Filter Bar

- **Search**: This is a button that displays a filtering dialog.
- **Description**: A non-labeled text field follows the Filter button that displays what criteria are currently selected to filter the staged message table.

Filter Dialog Attributes

- **Family**: Limits the search results to only those messages in the specified family.
- **In/Out**: Indicates if the message is inbound to EICS or outbound to an external system.
- **Search Limit**: Limits the number of messages found to the specified limit or less.
- **Show Pending**: If true, only messages that are pending will be displayed.
- **Show Retry**: If true, only messages that are in retry will be displayed.

Figure 6-8 MPS Staged Message Filter



Filter Dialog Buttons

- **Search**: Executes a search for staged messages based on the selected criteria and refreshes the list table with that information.
- **Reset**: Results the filter dialog back to its default settings.
- **Cancel**: Closes the dialog without taking any action.

List Buttons

- **Refresh**: Refreshes the staged message list with current information.
- **Retry**: If a staged message is selected, selecting this button will retry the processing of the staged message.
- **Delete Selected**: If a stage message is selected, selecting this button will attempt to delete the message.
- **Filter**: Activates the filter dialog allowing the user to filter information.

List Attributes

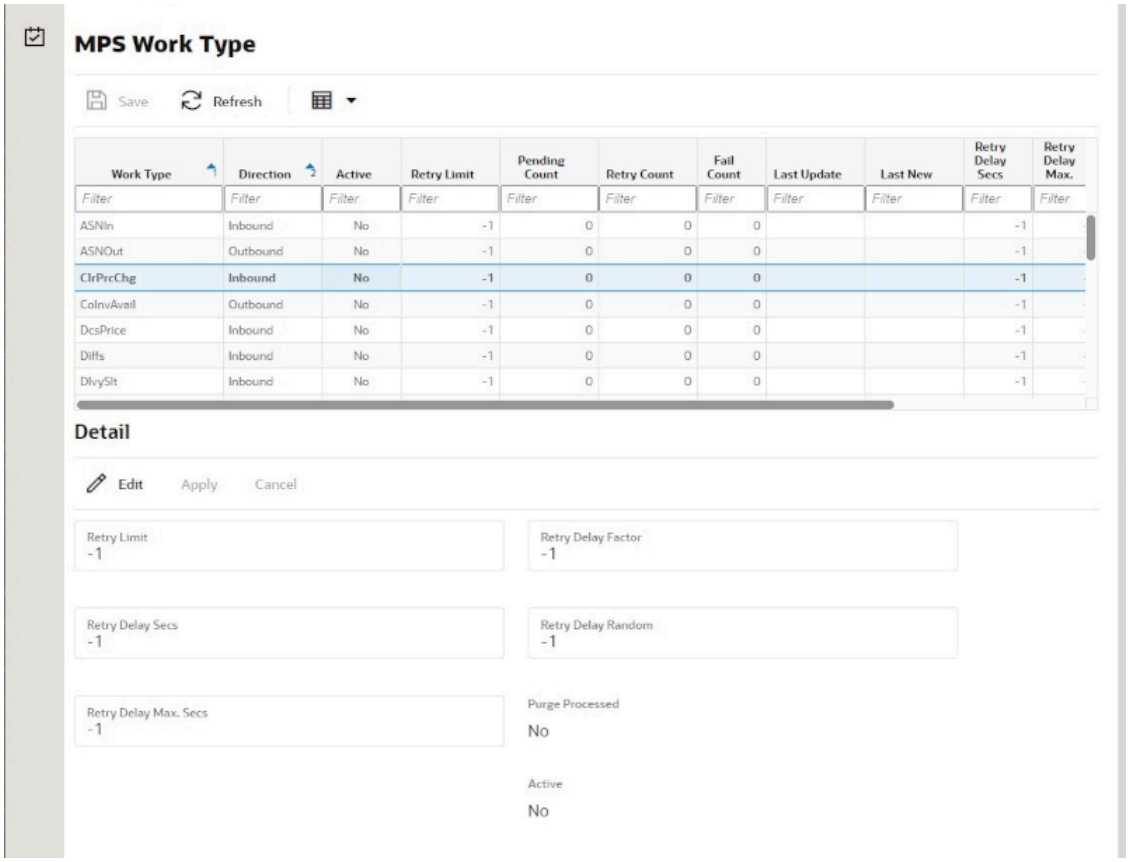
- **Record Id**: A unique identifier assigned to the stage message.
- **In/Out**: Indicates if the message is inbound to EICS or outbound to an external system.
- **Type**: Type of message within a family. For example, most families of messages have a create, modify, and delete type of message. See Appendix G: MPS Message Types for further information about DCS message types.

- **Family**: Describes the family that the message belongs to. This will align with a MPS Work Type scheduled to process this family of messages.
- **Create Time**: The timestamp of the first time the message was created.
- **Update Time**: The timestamp of the last time the message was updated.
- **Execution Count**: The number of times the system has attempted to process the message.
- **Business ID**: A business identifier associated to the message. More than one message may carry a business identifier, and so this can be used to associate messages that may be related to the same activity.
- **Store ID**: The identifier of the store associated to the message.
- **Job ID**: The job identifier is a server-generated sequence number used for grouping related messages. If a message is singular with no other related messages, then its record identifier and job identifier will be identical. For messages, that must be executed sequentially together as a group, they will be assigned the same job identifier but different record identifiers.
- **Description**: A brief formatted description of the staged message that gives some indication of the contents within.

MPS Work Type

The MPS Work Type administration form is used to configure the MPS work types. MPS stands for Message Processing System and each work type represents an external inbound or outbound message family or grouping to be delivered to another system. These work types do not represent the messages themselves (see MPS Staged Message) but the working queue that handles the processing of these external messages.

Figure 6-9 MPS Work Type List



List Buttons

- **Save**: Persists currently altered information.
- **Refresh**: Refreshes the screen with currently persisted information.
- **Filter**: Enabled are disabled the filtering row in the table.

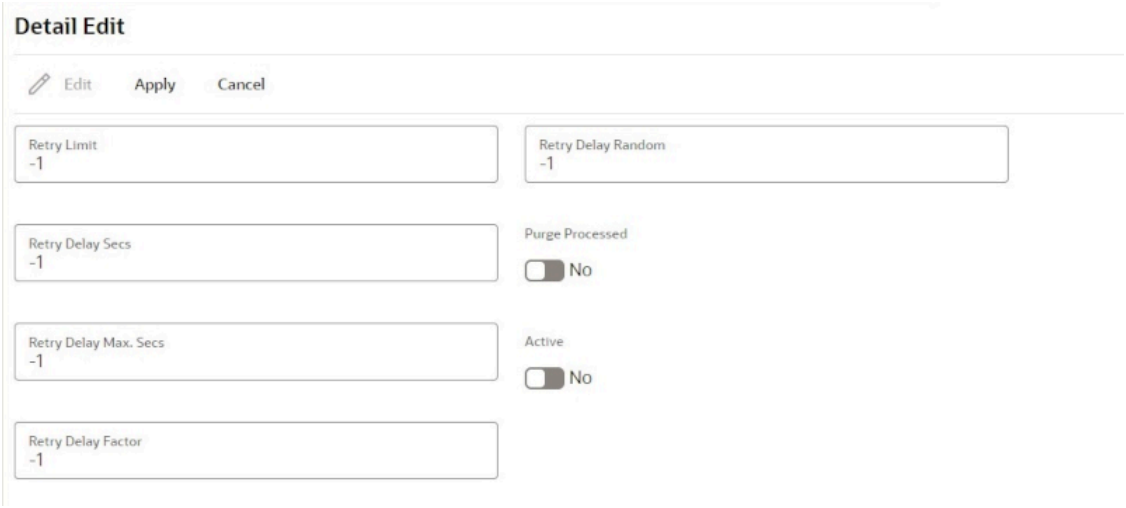
List Attributes

- **Work Type**: The work type is the name of the message queue being worked on.
- **Direction**: Indicates if the message queues is inbound to EICS or outbound to an external system.
- **Active**: Yes indicates the work type is currently active and attempting to process messages. No indicates it has been disabled.
- **Retry Limit**: The number of times to attempt to process a single message before marking it as failed.
- **Pending Count**: Number of messages pending processing.
- **Retry Count**: Number of times the system as attempted to process the message. Zero times means processing has not been attempted yet.
- **Fail Count**: Number of messages that have failed to be processed.
- **Last Update**: The timestamp of the last time a staged message record was updated in the database for this specific work type.

- **Last New**: The timestamp of the last time a staged message record was created in the database for this specific work type.
- **Retry Delay Secs**: The delay in seconds between retries.
- **Retry Delay Max Secs**: The maximum delay in seconds between retries.
- **Retry Delay Factor**: This factor is used to increase retry delay. The access to this flag is restricted to Oracle.
- **Retry Delay Random**: The factor used to limit the range of retry delay randomization. The access to this flag is restricted to Oracle.
- **Purge Processed**: Indicates if automatic purging of messages that are processed successfully is enabled. This flag is enabled by default and the edit access is restricted to Oracle.
- **Update Date**: The date/time when the MPS work type was updated.
- **Update User**; The user that updated the work type.

Detail Panel

Figure 6-10 Detail Panel In Edit Mode



Detail Buttons

- **Edit**: Places the detail information displayed into edit mode.
- **Apply**: Applies the current data to the work type and updates the list information.
- **Cancel**: Places the panel back into display mode without applying the information.

Detail Attributes

- **Retry Limit**: The number of times to attempt to process a single message before marking it as failed. Zero indicates it should only be attempted once and will not be retried. A positive value indicates how many attempts to process the message should be made in addition to the original attempt.
- **Retry Delay Seconds**: The number of seconds between attempts to process a message. Zero indicates no delay whereas a value of 30 indicates 30 seconds between attempting to

process the message. This value does not represent an absolute value as it used with other parameters below to produce a calculated message time delay.

- **Retry Delay Max Seconds**: This is the maximum number of seconds for a delay between processing messages. This caps the maximum value for the calculated delay.
- **Retry Delay Factor**: This attribute produces an increased delay between each retry of a failed message in the queue. It is a decimal value starting at 1.0 and increasing. If you enter 1.0, it means there will be no increase in the retry delay seconds during repeated attempts to process a message. A value of 1.5 indicates that the retry delay seconds will be 150% of the retry delay seconds on retry.

  A linear delay of 1.0 can be used but may result in messages reaching their retry limit prior to any issues being resolved. A value of 1.5 or larger will produce an increased delay that may allow time for other dependent messages that could be holding up execution to arrive. The access to this flag is restricted to Oracle.
- **Retry Delay Random**: Defines the amount of the delay to increase or decrease by a random amount. It accepts values from 0.0 to 1.0. A value of 0.0 disables random variation of the delay. A value of 0.5 indicates the calculated delay may be randomly altered by up to 50% of its value whereas 1.0 indicates it could be randomly altered up to 100% of its value. It is recommended to use at least some small random variation to improve message throughout as this will reduce resource convention and help avoid timing issues. The access to this flag is restricted to Oracle.
- **Notes**: The parameters are applied to retries in the following manner. The retry delay seconds is first increased by the retry delay factor, then the retry delay max seconds is applied, and finally the random delay is added or subtracted to determine the final number of seconds to wait between message processing attempts.
- **Purge Processed**: Switch to enable/disable automatic purge of messages that are processed successfully. This flag is enabled by default and the edit access is restricted to Oracle.
- **Active**: Switch to enable/disable a work type.

MPS Work Type (DCS)

Those message types prefixed with DCS (Data Collection System) is a series of work types that processes the staged messages that came from the DCS work types as they gathered data from MFCS and imported it into the stage message table. These work types take the DCS staged message and process it. These should be activated if MFCS is integration via a direct PDB integration (not using the RIB).

Table 6-1 MPS Work Type Messages

Work Type	Description
DcsAllocation	Processes allocation messages
DcsAsn	Processes incoming shipment messages
DcsDiff	Processes differentiator related messages (diffs, diff types)
DcsFiscalDocument	Processes fiscal document message
DcsHierarchy	Processes merchandise hierarchy messages (department, class, subclass)
DcsItem	Process item related messages (item, item images, etc)
DcsItemLocation	Processes item location messages (store items, warehouse items, item replenishment)

Table 6-1 (Cont.) MPS Work Type Messages

DcsOrder	Processes order messages (purchase orders)
DcsPartner	Processes finisher message
DcsPrice	Processes price messages
DcsRtv	Processes return-to-vendor request messages
DcsStore	Processes store messages
DcsSupplier	Processes supplier messages
DcsSupplierItem	Processes supplier item messages (supplier item, supplier item country, etc)
DcsTransfer	Processes transfer request messages
DcsUda	Processes user defined attribute messages
DcsWarehouse	Processes warehouse message

MPS Work Type (DPS)

Those message types prefixed with DPS (Data Publishing System) is a series of work types that processes the staged messages that are outgoing to external third party systems. These work types take the DPS staged message and process it sending it to configured REST service endpoints that must be directly implemented by a third party system.

Table 6-2 MPS Work Type Messages

Work Type	Description
DpsCountSchedule	Publishes stock count schedule messages
DpsDsdReceipt	Publishes direct-store-delivery receipt message
DpsFiscalDocument	Publishes fiscal document request messages
DpsInvAdjustment	Publishes inventory adjustment message
DpsNotification	Publishes system notification message
DpsShipment	Publishes shipment messages
DpsStockStatus	Publishes modifications to stock status messages (often such things a reserved status)
DpsStoreOrder	Publishes store order requests and approval messages
DpsTicketPrint	Publishes requests for ticking printing messages
DpsTransferReceipt	Publishes transfer receipt messages
DpsVendorReturn	Publishes return-to-vendor messages

MPS Work Type (RMS)

Those message types prefixed with RMS (Retail Merchandising System) is a series of work types that processes the staged messages that are outgoing specifically to MFCS through direct PDB shared tablespace. These work types take the RMS staged message and process it sending it to intermediate shared table to be picked up and processed by MFCS.

Table 6-3 MPS Work Type Messages

Work Type	Description
RmsCountSchedule	Publishes stock count schedule messages
RmsDsdReceipt	Publishes direct-store-delivery receipt message
RmsFiscalDocument	Publishes fiscal document request messages
RmsInvAdjustment	Publishes inventory adjustment messages
RmsPoReceipt	Publishes purchase order receipt messages
RmsShipment	Publishes shipment messages
RmsStockStatus	Publishes modifications to stock status messages (often such things a reserved status)
RmsStoreOrder	Publishes store order requests and approval messages
RmsTransferReceipt	Publishes transfer receipt messages
RmsVendorReturn	Publishes return-to-vendor messages

MPS Work Type (RIB)

Those message types that do have a prefix define a series of work types that processes the incoming messages from the RIB and the outgoing messages to the RIB.

Table 6-4 MPS Work Type Messages

ASNIn	Processes incoming shipment messages
ASNOut	Processes outgoing shipment messages
ClrPrcChg	Processes incoming clearance price change messages
ColInvAvail	Processes outgoing customer order store based inventory availability messages
Diffs	Processes incoming differentiator messages
DlvySlt	Processes incoming delivery slot messages
DSDReceipt	Processes incoming direst-store-delivery receipt messages
FulfilOrd	Processes incoming fulfillment order messages
FulfilOrdCfm	rocesses outgoing fulfillment order confirm messages
FulfilOrdCfmCnc	Processes outgoing fulfillment order confirm messages
Global (Inbound)	Overall processing type that activates all incoming work types
Global (Outbound)	Overall processing type that activates all outgoing work types.
InvAdjust (Inbound)	Processes incoming warehouse inventory position change messages.
InvAdjust (Outbound)	Processes outgoing store inventory position change message.
InvReq	Processes outgoing inventory request messages
ItemLoc	Processes incoming inventory location (store items, warehouse items) messages
Items	Processes incoming item message
ManifestCloseShipment	Processes incoming requests to close a manifest
MerchHier	Processes incoming merchandise item hierarchy messages

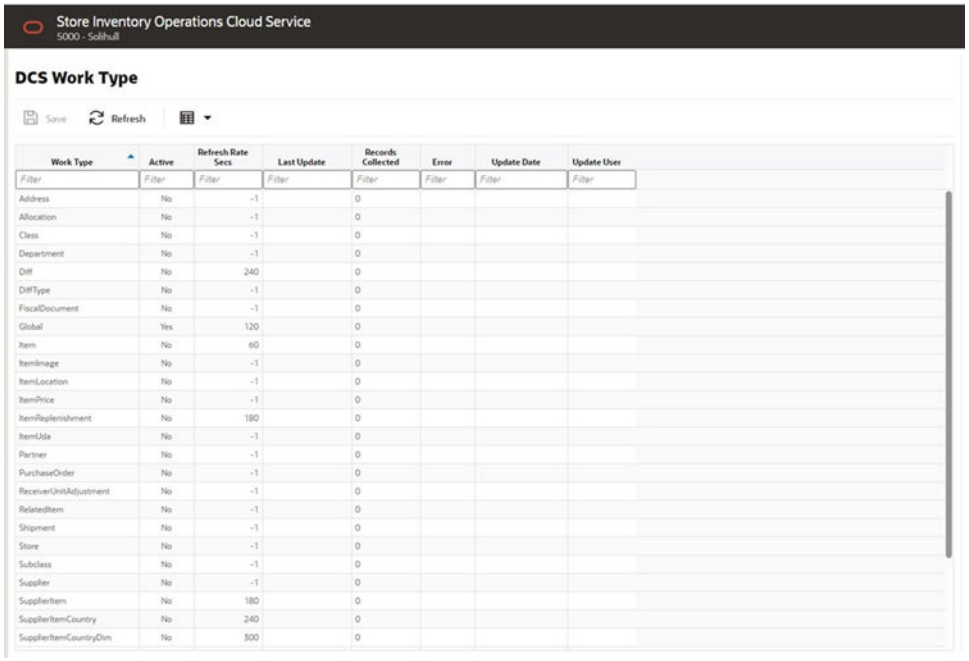
Table 6-4 (Cont.) MPS Work Type Messages

Notification	Processes outgoing system notification message
Order	Processes incoming purchase order messages
Partner	Processes incoming finisher messages
POSTransaction	Processes incoming POS transaction messages
PrmPrcChange	Processes incoming promotion price change messages
RcvUnitAdj	Processes incoming receiver unit adjustment messages
Receiving (Inbound)	Processes inbound receipt messages
Receiving (Outbound)	Processes outgoing receipt messages
RegPrcChg	Processes incoming regular price change messages
RTV (Inbound)	Processes inbound return-to-vendor request messages
RTV (Outbound)	Processes outbound return-to-vendor shipment messages
SeedData	Processes incoming basic foundation (differentiator types) messages
ShipInfo	Processes outgoing pre-shipment messages
SoStatus (Incoming)	Processes incoming stock order status change messages
SoStatus (Outcoming)	Processes outgoing stock order status change messages
StkCountSch	Processes outgoing stock count schedule messages
StockOrder	Processes incoming stock order messages
StoreOrder	Processes outgoing store order messages
Stores	Processes incoming store messages
TicketPrint	Processes outgoing ticket print request messages
UDA	Processes incoming user-defined-attributes messages
Vendor	Processes incoming supplier messages
WH	Processes incoming warehouse messages

DCS Work Type

The DCS Work Type administration form is used to configure the DCS work types. DCS stands for Data Collection System and each work type represents an external inbound message family or grouping to be delivered from MFCS. These work types do not represent the messages themselves (see MPS Staged Message) but the working queue that handles the processing of these external messages. DCS Work types are a type of polling system that reaches out to MFCS tables for recently modified data and transfers that information into MPS staged messages that will processed through normal MPS processing at that point.

Figure 6-11 DCS Work Type Form



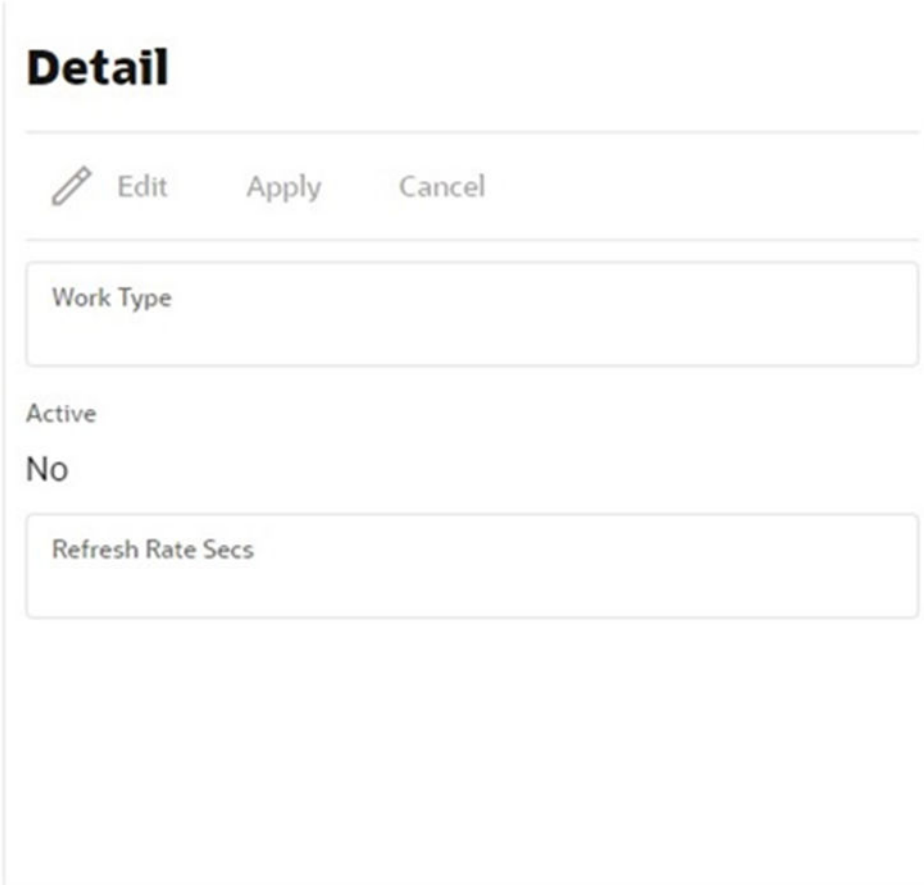
List Buttons

- **Save**: Persists currently altered information.
- **Refresh:** Refreshes the screen with currently persisted information.
- **Filter:** Enabled are disabled the filtering row in the table

List Attributes

- **Work Type:** The work type is the name of the message queue being worked on.
- **Active:** Yes indicates the work type is currently active and attempting to process messages. No indicates it has been disabled.
- **Refresh Rate Secs:** It indicates the interval after which the work type must start polling the information again.
- **Last Update:** The last timestamp when the work type woke up and gathered records(s).
- **Records Collection**: The number of records collected when the work type woke up last.
- **Error:** An error if the work type has failed and stopped functioning.
- **Update Date:** The date when the work type settings were last updated.
- **Update User:** The last user who updated the work type settings.

Figure 6-12 Detail



Detail Buttons

- **Edit:** Places the detail information displayed into edit mode.
- **Apply:** Applies the current data to the work type and updates the list information.
- **Cancel:** Places the panel back into display mode without applying the information.

Detail Attributes

- **Active:** Denotes whether the Work Type is active or not. When active, data will be gathered from MFCS for that work type.
- **Refresh Rate Seconds**: This indicates how frequently the work type will activate and attempt to gather information from MFCS.

Table 6-5 Work Type

Work Type	Description
Address	Gathers location addresses
Allocation	Gathers transfer allocations
Class	Gathers merchandise hierarchy class
Department	Gathers merchandise hierarchy departments
Diff	Gathers item differentiators

Table 6-5 (Cont.) Work Type

DiffType	Gathers item differentiator types
FiscalDocument	Gathers fiscal document response information
Global	This is the one used to override the settings of all work types
Item	Gathers item information
ItemImage	Gathers item image information
ItemLocation	Gathers item location information (item at store, warehouse, etc)
ItemPrice	Gathers item price information
ItemReplenishment	Gathers item location replenishment information
ItemUDA	Gathers user defined attributes for an item
Partner	Gathers finisher information
Purchase Order	Gathers purchase order information, including those for store order review
Receiver Unit Adjustment	Gathers receipt adjustments
RelatedItem	Gathers related item information
Shipment	Gathers shipment (ASN) information
Store	Gathers store information
Subclass	Gathers merchandise hierarchy subclasses
Supplier	Gathers supplier information
SupplierItem	Gathers supplier's item information
SupplierItemCountry	Gathers supplier's item information for a specific country
SupplierItemCountryDim	Gathers supplier's item's dimensions information at a specific country.
SupplierItemCountryManufacturer	Gathers the country of manufacture information for a supplier's item
SupplierItemUOM	Gathers unit of measure information for a supplier's item
Transfer	Gathers transfer request information
UDA	Gathers user defined attribute setup information
VendorReturn	Gathers information about vendor return requests
Warehouse	Gathers information about warehouses

Operational Issues Screens

This chapter describes administration screens which users with System-Operator role, can view for operational issues.The operational issues are divided into four categories.

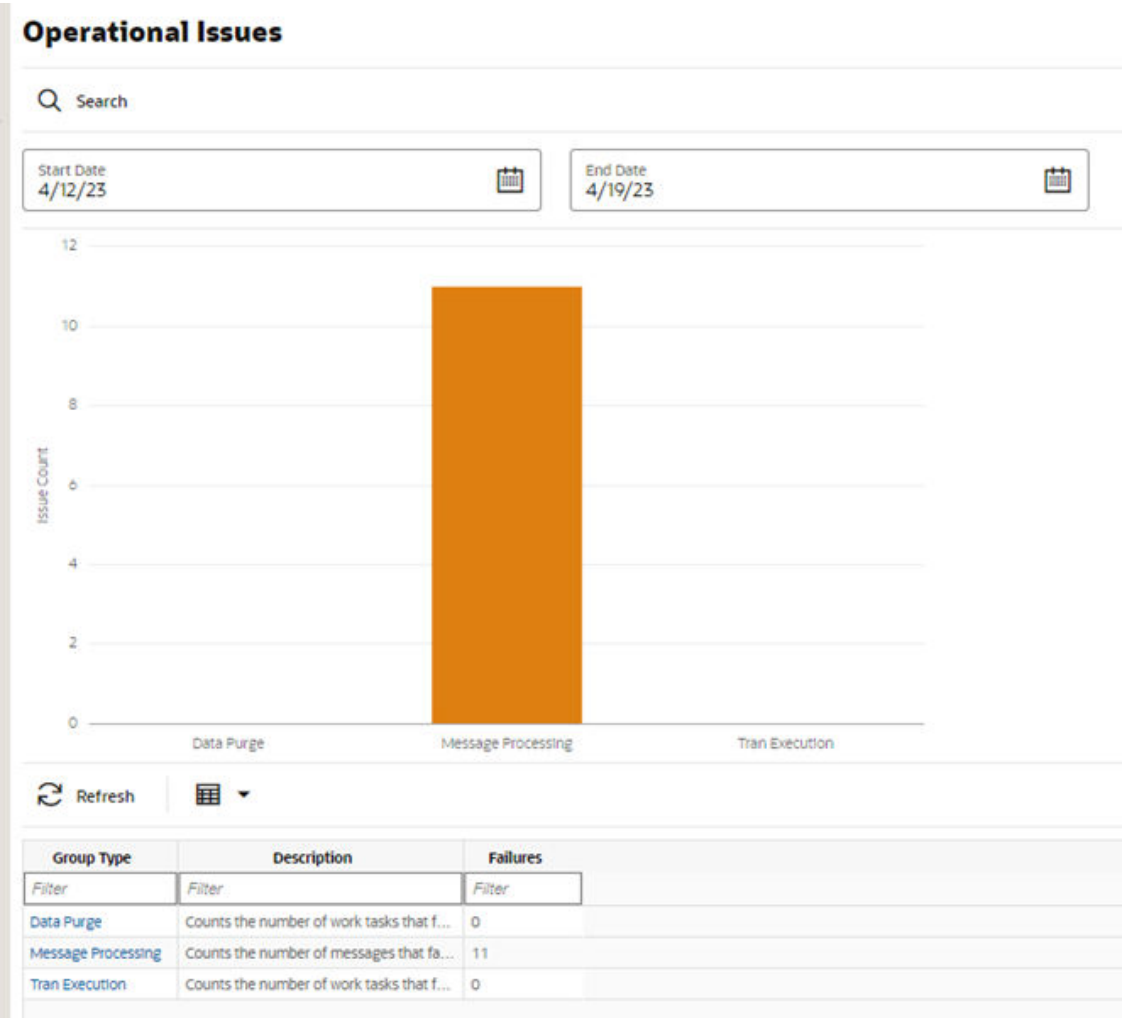
> **Note:**
>
> **Data Search Range** has been defaulted to last 14 days. Users can change the date range.

- **Data Purge** — Scheduled background work tasks that archive and remove data from the database. The issue counts the number of work tasks that failed. Primary Tables: BATCH_EXECUTION, BATCH_ACTIVITY
- **Message Processing** — Scheduled background work tasks that process asynchronous messages in a queue. The issue counts the number of messages that failed. Primary Tables: MPS_STAGED_MESSAGE
- **Transactional Execution** — Scheduled background work tasks that execution business processes on transactional data. The issue counts the number of work tasks that failed. Primary Tables: BATCH_EXECUTION, BATCH_ACTIVITY

Operational Issues List Screen

This screen displays the summary of operational issues for each category.

Figure 6-13 Operational Issues List Screen



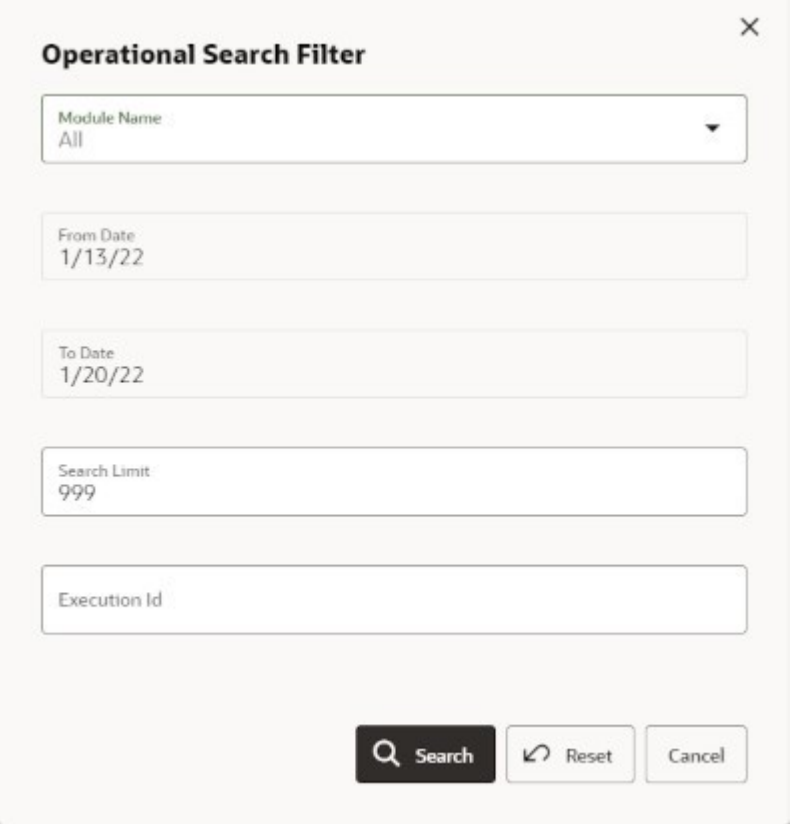
> **Note:**
>
> On the bar graph, the **Issue Count** will be displayed when user moves the cursor to the bar graph.

Operational Issues Review

This screen displays the operational issues for selected categories and date ranges. It provides following common actions in the operational issue review screens:

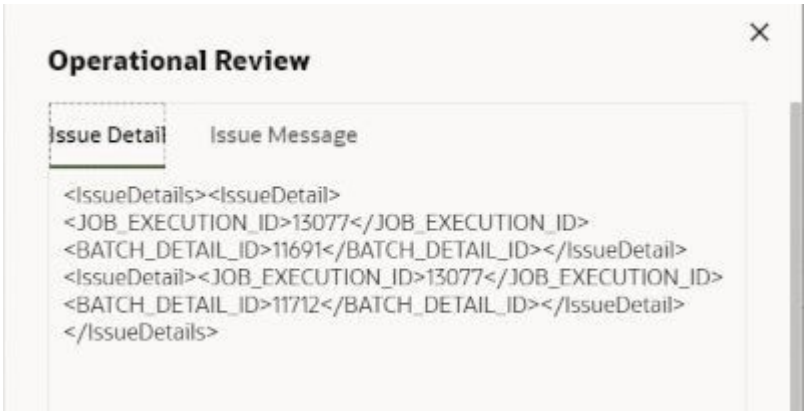
- **Sreach Filter** — Each Operational Review Screen has Search filters. The search filter fields vary based on the operational groups. The search filter fields are: Search Limit, date range.

Figure 6-14 Search Filter



- **Issue Detail** — Click the Issue Link, and the issue detail dialog will be displayed. For example, the following screenshot displays the Bulk Data Import Issue Detail:

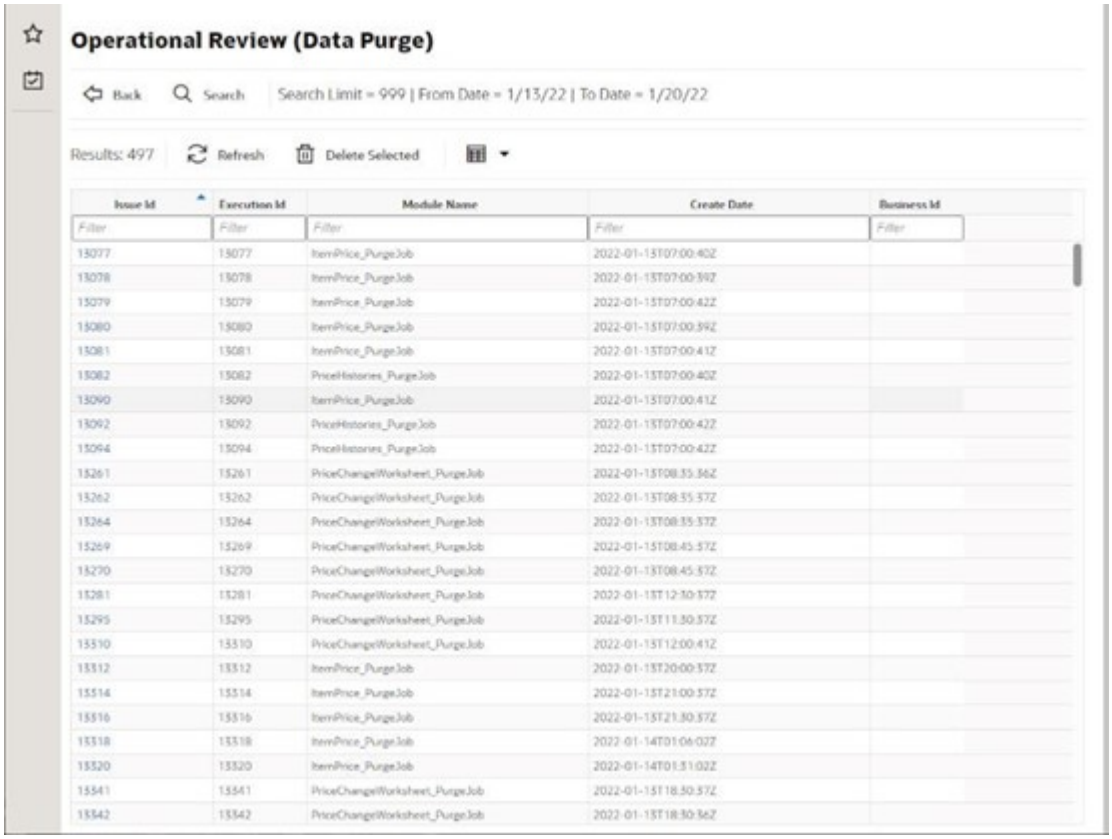
Figure 6-15 Issue Detail



Operational Review (Data Purge)

This screen displays a list of failed scheduled background work tasks that archive and remove data from the database.

Figure 6-16 Data Purge Screen



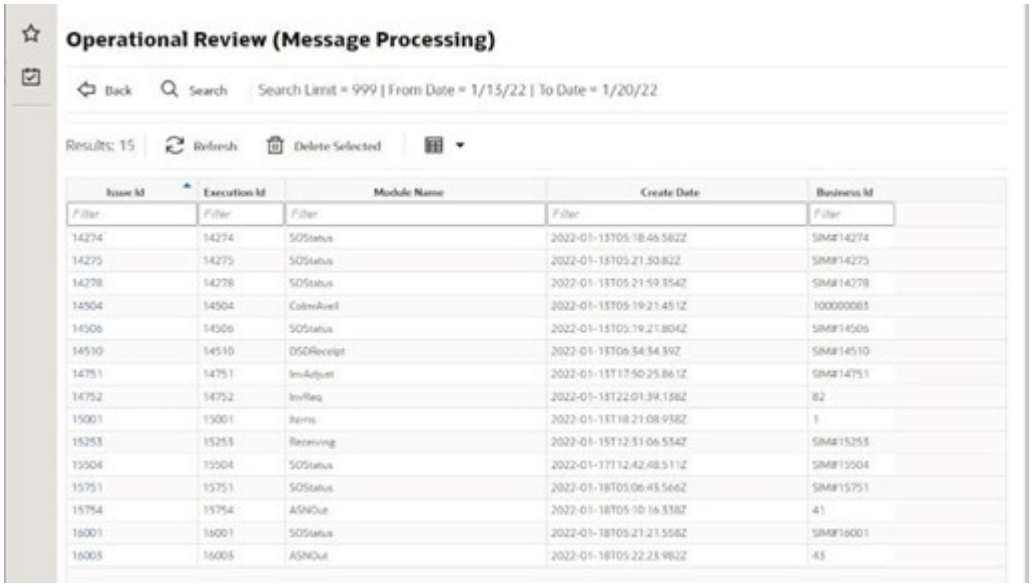
Delete Selected Button

On the Operational Review (Data Purge) screen, the “Delete Selected” button will delete the selected batch execution records if user has security permission “Batch Execution Delete”

Operational Review (Message Processing)

This screen displays a list of failed background work tasks that process asynchronous messages in a queue.

Figure 6-17 Message Processing Screen



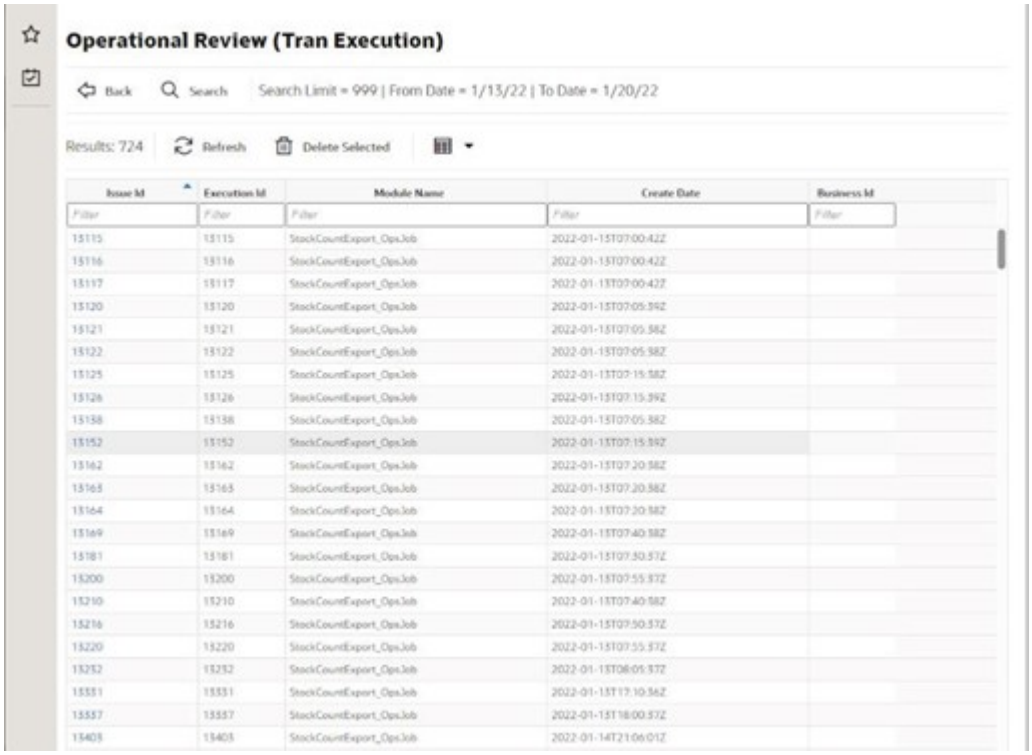
Delete Selected Button

Onthe Operational Review (Message Processing) screen, the “Delete Selected” button will mark the selected record MPS staged message record as deleted if user has security permission “Delete MPS Staged Message”

Operational Review (Transactional Execution)

This screen displays a list scheduled background work tasks that execute business processes on transaction batches if user has security permission “Batch Execution Delete”

Figure 6-18 Transactional Execution Screen



Delete Selected Button

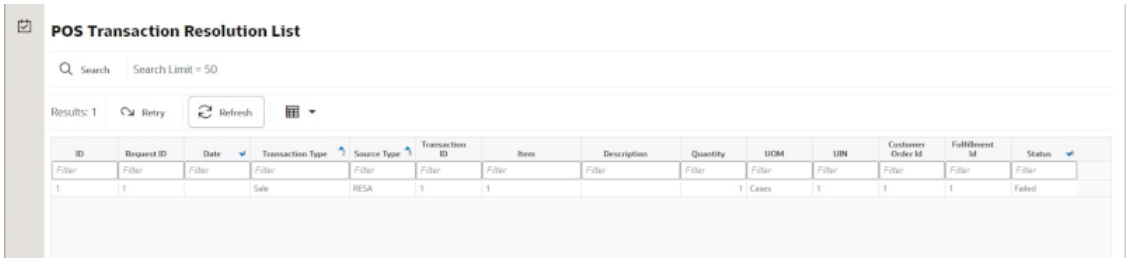
On Operational Review (Transaction Execution) screen, the “Delete Selected” button will delete the selected batch execution records.

POS Transaction Resolution

The POS Transaction Resolution screen allows viewing and maintenance of point-of-sale transactions that failed processing and all retry attempts have failed:

Security Permission: Troubled POS Transaction Resolution

Figure 6-19 POS Transaction Resolution List



List Buttons

- **Search**: Navigates to the resolution search criteria screen.

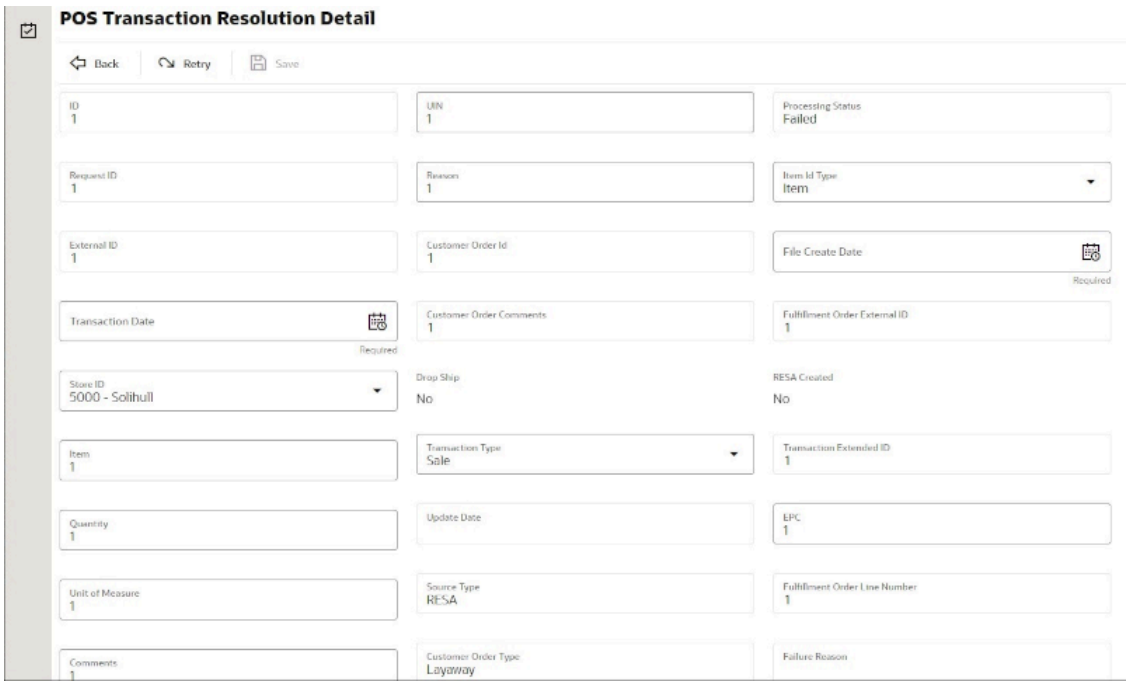
- **Retry**: Will reset and retry processing for the selected POS transaction.
- **Refresh**: Will refresh the screen with current up to date information.

List Attributes

- **ID**: An internally generated unique transaction identifier. Clicking on the ID link will navigate to the POS Transaction Resolution Detail screen.
- **Request ID**: A request identifier indicating the processing group it was executed with.
- **Date**: The date of the transaction.
- **Transaction Type**: The type of transaction that occurred.
- **Source Type**: The source of the transaction.
- **Transaction Id**: The full sale transaction that this item sale is a part of.
- **Item**: The item.
- **Description**: The description of the item.
- **Quantity**: The quantity of item that was processed.
- **UIN**: A universal identifiable number (such as Serial number). If this is present, the quantity of the transaction is 1.
- **Co ID**: Customer Order identifier.
- **Fulfillment ID**: Fulfillment Order identifier.
- **Process Status**: The current state of processing for this item transaction.

Detail Screen

Figure 6-20 POS Transaction Resolution Detail



Detail Buttons

- **Edit**: Places the detail information displayed into edit mode.
- **Apply**: Applies the entered data to the record and updates the list information.
- **Cancel**: Places the panel back into display mode without applying the information.

Detail Attributes

- **ID**: An internally generated unique transaction identifier.
- **Request ID**: A request identifier indicating the processing group it was executed with.
- **External ID**: The external sale transaction that this item sale is a part of.
- **Transaction Date**: The date of the transaction.
- **Store ID**: The identifier of the store the transaction took place item.
- **Item**: The item number.
- **Quantity**: The quantity of the transaction.
- **Unit Of Measure**: The unit of measure of the quantity.
- **Comments**: Comments associated to the point-of-sale transaction.
- **UIN**: A unique number, such as a serial number, associated with the transaction.
- **Reason**: A reason associated with the transaction.
- **Customer Order Id**: A customer order identifier if a customer order is associated to the transaction.
- **Customer Order Comments**: Comments associated with a customer order.
- **Drop Ship**: Yes indicates drop ship.
- **Transaction Type**: The type of transaction: sale, return, void sale, void return, customer order, customer order cancelation, customer order fulfillment.
- **Update Date**: The timestamp of the last update of this transaction record.
- **Source Type**: The source type of the transaction: RESA or POS.
- **Customer Order Type**: The type of customer order: Layaway, Pickup, Customer Order, Pending Purchase, Special Order, Web Order, or On Hold.
- **Processing Status**: The status of the POS transaction: New, Processed, Failed, Retry, or Reverted.
- **Item Id Type**: The type of item identifier: ITEM or UPC.
- **File Create Date**: The date the file the data came from was created.
- **Fulfillment Order External Id**: The external order system identifier for the fulfillment order.
- **RESA Created**: Yes if the record was created in RESA.
- **Transaction Extended ID**: A full generated unique transaction identifier.
- **EPC**: An EPC if one exists.
- **Fulfillment Order Line Number**: The
- **Failure Reason**: The reason the POS transaction is in a failed state.

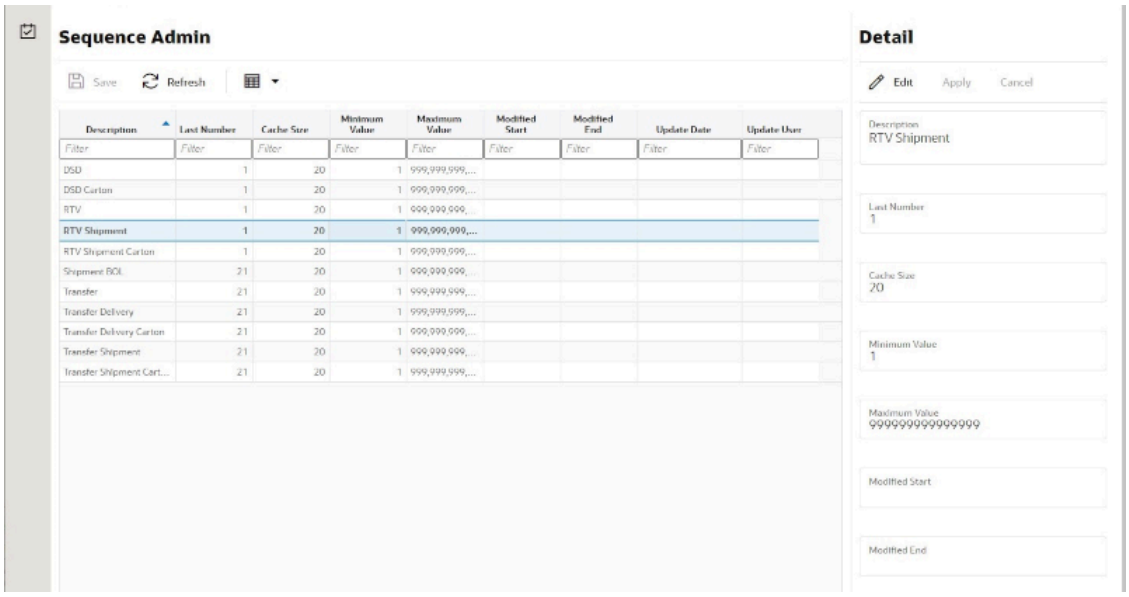
Sequence Administration

The Sequence Administration screen is used to setup database sequence information for a specific set of tables. This can be used to prevent overlapping sequence generation between multiple database tables.

To access this screen, user need to be assigned followings:

Security Permission: Access Sequence Administration

Figure 6-21 Sequence Administration



List Buttons

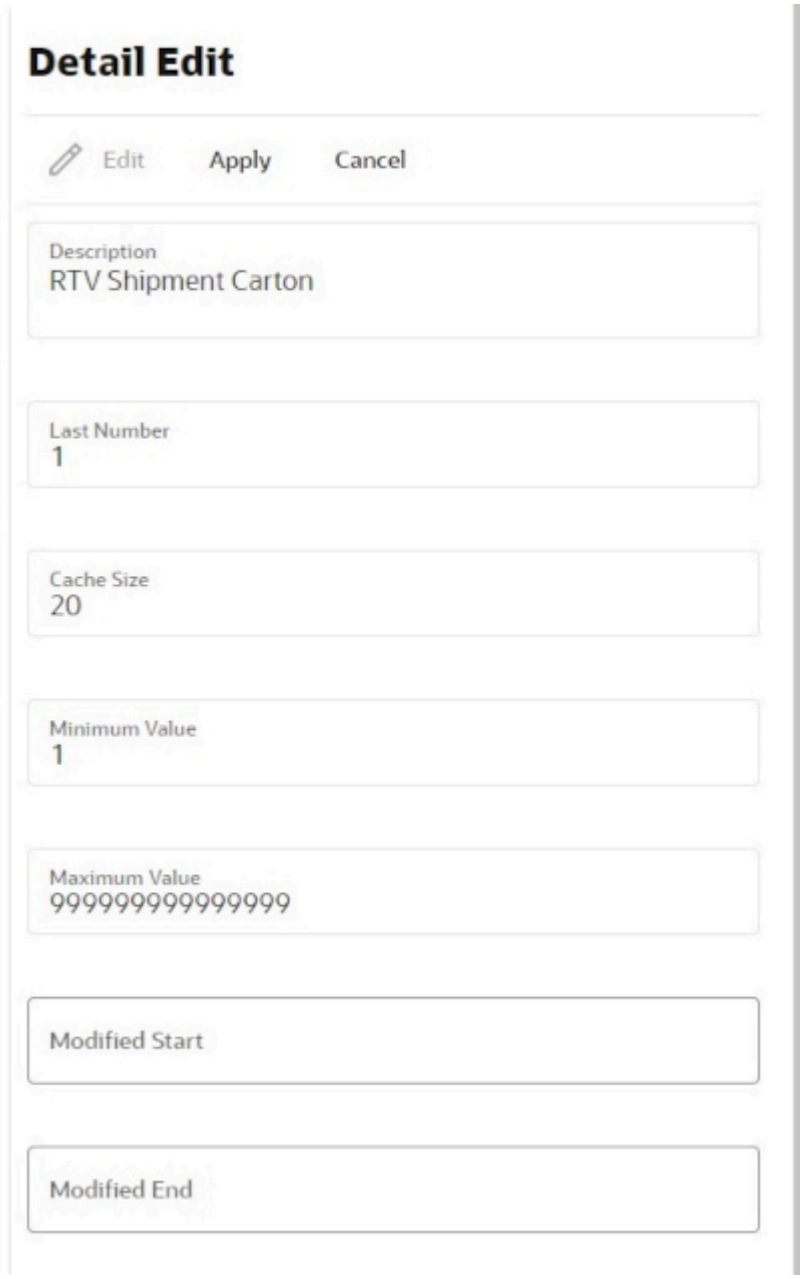
- **Save**: Persists currently altered information and refreshes the screen.
- **Refresh**: Refreshes the screen with currently persisted information.
- **Filter**: Enabled are disabled the filtering row in the table.

List Attributes

- **Description**: A description of the sequence.
- **Last Number**: The last currently used sequence number.
- **Cache Size**: The number of sequence number to keep in the database cache.
- **Minimum Value**: The minimum value the sequence can become.
- **Maximum Value**: The maximum value the sequence can become.
- **Modified Start**: Retailer specified starting number for the sequence.
- **Modified End**: Retailer specified ending number for the sequence.
- **Update Date**: The timestamp of the last update of the record.
- **Update User**: The user associated with the last update of the record.

Detail Panel

Figure 6-22 Detail Panel in Edit Mode



Detail Buttons

- **Edit**: Places the detail information displayed into edit mode.
- **Apply**: Applies the entered data to the record and updates the list information.
- **Cancel**: Places the panel back into display mode without applying the information.

Detail Attributes

- **Description**: The description of the sequence administration record. This can only be changed through translation administration (view only).
- **Last Number**: The last number written to the database disk (view only).
- **Cache Size**: The number of sequences kept in the database cache (view only).
- **Minimum Value**: The minimum value the sequence can become (view only).
- **Modified Start**: The retailer modified starting sequence number. It must be greater than minimum value and less than modified end.
- **Modified End**: The retailer modified ending sequence number. It must be less than the maximum value and greater than modified start.

Integration Dashboard

The integration dashboard screen displays information about the integration messages publication and subscription failures and the current processing statistics. This screen can be accessed from the technical maintenance menu. Administration users can use this dashboard to quickly verify the message failures based on the message families and export any data if applicable for fixing.

On the header portion, the system displays the pie chart with the failure numbers based on the message family and the total processing, failed and aged values. Aged here indicates the total integration messages that are aged>=24 hours and still not processed.

Users need proper permission to access this dashboard. A user with access permission is allowed to do all the operations on this screen.

The system supports the below message groups.

1. **DCS = Data Collection System**

   DCS Inbound counts the number of DCS family MPS messages in MPS staged message table waiting to be processed or failed.

   DCS Outbound does not exist.

2. **RIB = Retail Integration Bus**

   RIB inbound counts the number of inbound RIB family MPS messages in MPS staged message table waiting to be processed or failed.

   RIB outbound counts the number of outbound RIB family MPS messages in MPS staged message table waiting to be processed or failed.

3. **Bulk = Mass Data Imports**

   Overlaps with other areas.

   Bulk imports count mass temporary storage tables or things that process asynchronously in large quantities.

   Includes bulk REST imports, bulk transaction file imports and pos transaction.

   Bulk allows exporting of errors.

Figure 6-23 Integration Dashboard

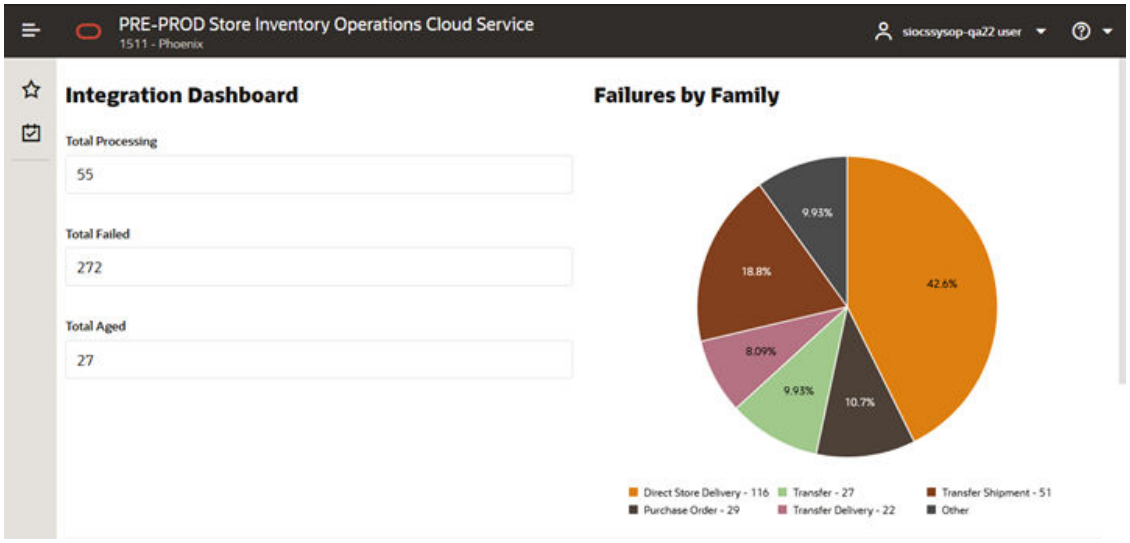
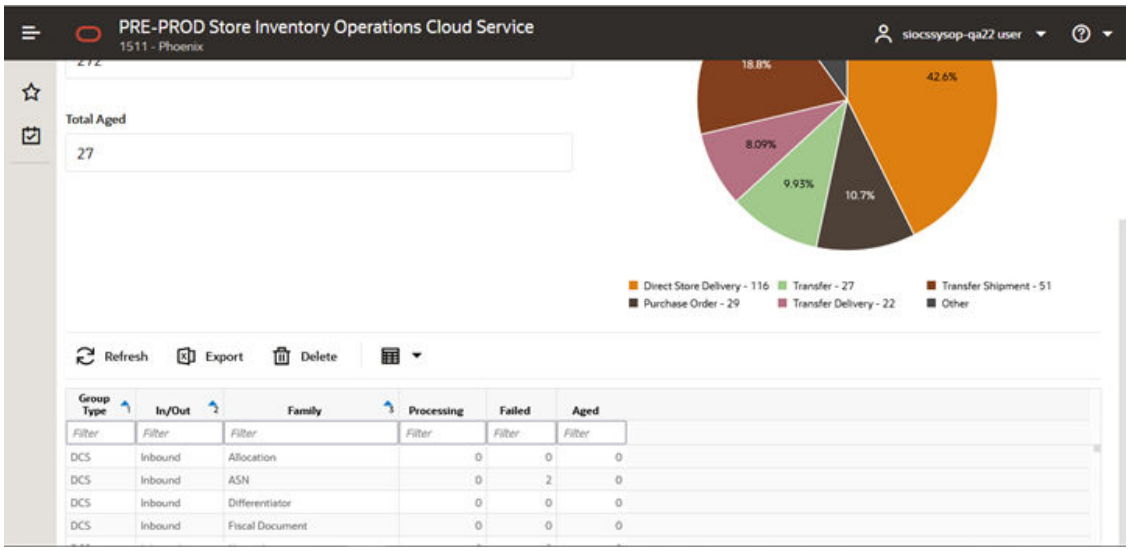


Figure 6-24 Integration Dashboard Showing Statistics



List Attributes

- **Group Type**: Values include DCS, RIB and Bulk.
- **In/Out**: This is to indicate whether it is inbound or outbound.
- **Family**: This displays the message family.
- **Processing**: Total messages under processing for the group type + in/out and family.
- **Failed**: Total messages failed for the inbound or outbound for the message family..
- **Aged**: Total messages that are not processed >=24 hours for the shown message family.

Detail Buttons

- **Export**: Pressing this button after selecting a ‘Bulk’ type group will export all failed messages for that bulk type group to one or more files that can be retrieved via the file transfer system. Export is only allowed for bulk group types.
- **Export Process**: When exported, all the records for a particular bulk type that are currently marked in failed status will be exported. Once the files are created, the standard process for file transfer can be used. If you load several different initial import files before exporting, the errors from all previous file uploads will be exported at one time.
- **Delete**: This button is used when the user wants to permanently delete all error integration messages that are marked as previously exported for a particular bulk group type. Delete is only allowed for bulk group types.
- **Delete Process**: Once data is loaded from an external source into bulk processing intermediate tables, the data may end up failed due to business process or technical errors. Once previously failed data is exported to the file, the user can delete the records in the table.

  All exported records for the selected family will be deleted, so if several files are loaded and several error files are exported, then the delete feature is used, it will delete all exported records. None exported records will not be deleted.

  Duplicate records are not allowed in the intermediate temporary tables, so attempting to reload previous failed data which is now fixed will automatically fail unless the previous copy of the record is removed.

7

Integration

This section describes the integration through RIB, batches, and web services.

- Retail Integration Cloud Service (RICS) - based Integration
- SOAP Web Services
- REST Web Services
- Sales Integration
- Integration with Customer Order System
- Integration with Manifesting Systems
- Integration for Notifications
- Integration for Sales Forecast
- Integration for Sales Forecast
- Integration for Ticket Printing
- Retail Home Integration
- REST Web Service OAuth2 Requests

Retail Integration Cloud Service (RICS) - based Integration

- Security Considerations
- Customer Orders
- Picking
- Deliveries
- Reverse Picking
- Multi Leg
- RIB Payloads
- Purchase Orders and Vendor Deliveries
- Inventory Adjustments
- Items
- Stock Counts
- Transfers
- Transfer Creation
- Transfer Messages
- Transfer Shipment Creation
- Transfer Receiving
- Transfer Doc

- Transfer Shipment
- Transfer Receiving
- Vendor Return

Security Considerations

Customer Administration User must create an IDCS user with the required RIB admin group to access the publisher endpoints.

- ribAdminGroup – For Production environment
- ribAdminGroup_preprod – For Dev/Stage/UAT/test environments

The same user credentials must then be configured on the Credential Administration screen. Refer to Chapter 6 - Technical Maintenance Screens / Credential Administration section for more details.

Customer Orders

- Customer Order Create is used for Customer Orders that are a type of Web Order integrated through a message (FulfilOrdDesc). These integrations are used for the customer order from the Order Management System (OMS).
- The Customer Order Create failure message (FulfilOrdCfmDesc) is a message that will be sent out to external system when we get a Customer Order that comes into the system through the RIB and fails due to validation issues such as an invalid item. The purpose of the create failure is so other systems will know it has failed when it came in and that it is not being processed.
- The Stock Order Status message (SOStatusDesc) will be sent out with an SI upon reserving inventory for the customer order.

Picking

- A Stock Order Status message (SOStatusDesc) is sent out with a type of SI upon reserving inventory. This happens when more is picked than what was on the order due to tolerances. This could also occur when a substitute item is added during the picking process.
- The Stock Order Status message (SOStatusDesc) with a type of SD will be published to un-reserve the original items inventory when a substitute item has been added during picking.
- A Stock Order status message (SOStatusDesc) is sent out with a type of PP when picking is completed.
- Item Substitutes are sent to EICS from the merchandising system through the item message (ItemDesc).

Deliveries

- An ASN Out message (ASNOutDesc) is sent out upon dispatching of the Delivery. This will be done for pick-ups and for shipments.
- The Stock Order Status message (SOStatusDesc) with a type of PP will be published for the pick quantity in the scenario that more was delivered than what was picked.

- The Stock Order Status message (SOStatusDesc) with a type of SI will be published for the reserved quantity. This will occur when more was delivered than what was reserved. This can happen when picking was not required, the reservation occurs upon receipt of a delivery, and the full amount had not been received, therefore not reserved.

Reverse Picking

- Customer Order Cancellations (FulfilOrdRef) will come into EICS from external system such as an OMS through the RIB. This service will perform all the validations to determine if it should create a reverse pick and whether or not that reverse pick should be auto completed.
- Customer Order Cancellation Confirmation (FulfilOrdRef) is a message to send to OMS upon completing of the system-generated reverse pick.
- Stock Order Status message (SOStatusDesc) with a type of SD will be published for the reserved quantity to un-reserve the inventory for the reverse pick for system-generated picks.
- Stock Order Status message (SOStatusDesc) with a type of PU will be published for the reverse picked quantity to un-pick the inventory for system-generated picks.

Multi Leg

The following integrations are in addition to the standard integrations that already exist such as receipt message, and so on:

- The Stock Order Status message (SOStatusDesc) with a type of SI will be published for the reserved quantity.
- The Stock Order Status message (SOStatusDesc) with type of PP will be published for the picked quantity.

RIB Payloads

RIB payloads are used to communicate information to external systems through RIB Integration.

RIB Payload	Description
FulfilOrdDesc	RIB payload that contains information about a new web order type of fulfillment order to be created in.
FulfilOrdCfmDesc	RIB payload sent from EICS that contains fulfillment order information when that order creating in EICS failed
FulfilOrdRef	RIB payload that contains information about a fulfillment order cancelation. It is sent to EICS to convey a cancelation request and sent from EICS to convey actual cancellations.
SOStatusDesc	Sent from EICS to convey changes in item status for a specific fulfillment order. Such changes of status include (un)reservation and (un)picking.
ASNOutDesc	Sent from EICS to convey a delivery for specified fulfillment order.

Purchase Orders and Vendor Deliveries

MERCHANDISING publishes the Purchase Orders created for the direct store deliveries using RIB messages. EICS subscribes to these messages and stores them in the EICS database to enable receipt against Purchase Orders.

MERCHANDISING publishes the unit cost of the item at the item/supplier/country level for EICS to use in the receiving process.

EICS publishes the receipts done against the Purchase Order to the merchandising system (Receiving message).

EICS publishes the DSD receipts created in EICS without a Purchase Order to the merchandising system (DSDReceipts and DSD Deals messages).

EICS publishes the receiver unit adjustment done for the deliveries that are already confirmed (receiving message).

EICS is also capable of subscribing to the vendor EDI ASNs through RIB using the ASN In message format.

RIB payloads are used to communicate information from EICS to external systems and from external system to EICS through RIB Integration.

RIB Payload (Subscriber)	Description
PORef	RIB payload that contains reference level information of a purchase order.This payload is used for removal of purchase orders.
PODesc	RIB payload that contains detailed information of a purchase order.This payload is used for creation and modification of purchase orders.
ASNInRef	RIB payload that contains reference level information of an ASN. This payload is used for removal of an ASN.
ASNInDesc	RIB payload that contains detailed information about the ASN. This payload is used for creation of a direct delivery (document type= 'P') or a warehouse delivery (document type= 'D'). EICS consumes this payload from warehouse when source and/or destination for ASN is a warehouse system.

RIB Payload	Description
ReceiptDesc	RIB payload that contains detailed information of the direct delivery receipt. This is published when the purchase order is not null. EICS also consumes this payload for warehouse receiving.
DSDReceiptDesc	RIB payload that contains detailed information of the direct delivery receipt. This is published when the purchase order is null.
SOStatusDesc	RIB payload sent from EICS to convey changes in item status for a specific fulfillment order. EICS also consumes this payload from warehouse for stock movements originating at the warehouse.
InvAdjustDesc	RIB payload that contains information about destination of the adjustment and an InvAdjustDtl.

Inventory Adjustments

Inventory adjustments integrate to MERCHANDISING at the item level using the RIB. EICS creates the adjustments and groups them together by a header with multiple items, but for integration purposes they are published out at an item level.

Inventory adjustments are published for all manual and external system generated adjustments where the Publish indicator for the reason code is checked. Adjustments are also published for other types of transactions in EICS where the merchandise system is expecting an adjustment for stock on hand updates, for example, receiving a DSD with damaged goods. An adjustment is created behind the scenes only for publishing purposes to notify the merchandising system to move the goods into the unavailable bucket. These system type adjustments are not considered an adjustment within EICS; however, they are published as such for integration purposes.

EICS subscribes to inventory adjustment messages from warehouse systems and updates the warehouse inventory buckets in EICS.

RIB payloads are used to communicate to external systems through RIB Integration.

The following table shows the list of RIB Payloads available for inventory adjustments.

RIB Payload	Description
InvAdjustDesc	RIB payload that contains information about destination of the adjustment and an InvAdjustDtl.
InvAdjustDtl	Contains detailed information about the item adjustment.

Items

Items come to EICS from a merchandising system through the RIB (items, item loc messages). EICS also gets information about items associated to a supplier through the RIB. Extended attributes are not received or sent on RIB payloads.

RIB Payload	Description
ItemDesc	This payload contains information about an item. It contains a wide variety of information about the item including suppliers, UPCs, ticketing information, image information, UDAs, and related items
ItemLocDesc	This payload contains information about an item at a specific location.
ItemSupDesc	This payload contains information about an item for a specific supplier.
ItemSupCtyDesc	This payload contains information about an item for a specific supplier within a specific country.

Stock Counts

Stock counts generate inventory adjustment when completed.

RIB payloads are used to communicate to external systems through RIB.

RIB Payload	Description
InvAdjustDesc	RIB payload that contains information about destination of the adjustment and an InvAdjustDtl.

RIB Payload	Description
InvAdjustDtl	Contains detailed information about the item adjustment.

EICS does not integrate using a web service to any other Oracle Retail products for stock counts.

Transfers

The Transfer Shipping allows for creating shipment, dispatching shipment, canceling shipment, creating container, approving container, adjusting container, and canceling the container.

The Transfer Receiving dialog allows for confirming receipt, copying misdirected container, receiving container and detailed receiving.

This section covers creating transfer documents which are then included in a transfer shipment and dispatched to another store, warehouse, or finisher.

Transfer Creation

Transfer documents can be created in the following ways:

- Requesting store can create a transfer request.
- Sending store can initiate a transfer by creating a transfer.
- Merchandising can create a transfer request.

Each transfer document will have one or more items.

Transfer Messages

EICS will publish messages to Merchandising when the following happen:

- Transfer is rejected.
- Transfer is approved.
- Transfer quantity is updated from the shipment.

Transfer Shipment Creation

Transfer Shipment describes the containers and the items for the shipment taking place. The shipment may be for one or more transfer documents if the transfer is going to the same destination. Dispatching a shipment will update the transfer document.

The user can create a shipment without referencing existing transfers or can create a new transfer on fly (Ad hoc transfer) based on the shipment information.

Transfer Receiving

This transaction captures a delivery that took place from a warehouse, store, or finisher to the store receiving the delivery. It describes the containers and the items of the delivery that should be received by the store. Receiving a container of the delivery will update the transfer document.

Figure 7-1 Transfer Request Flow

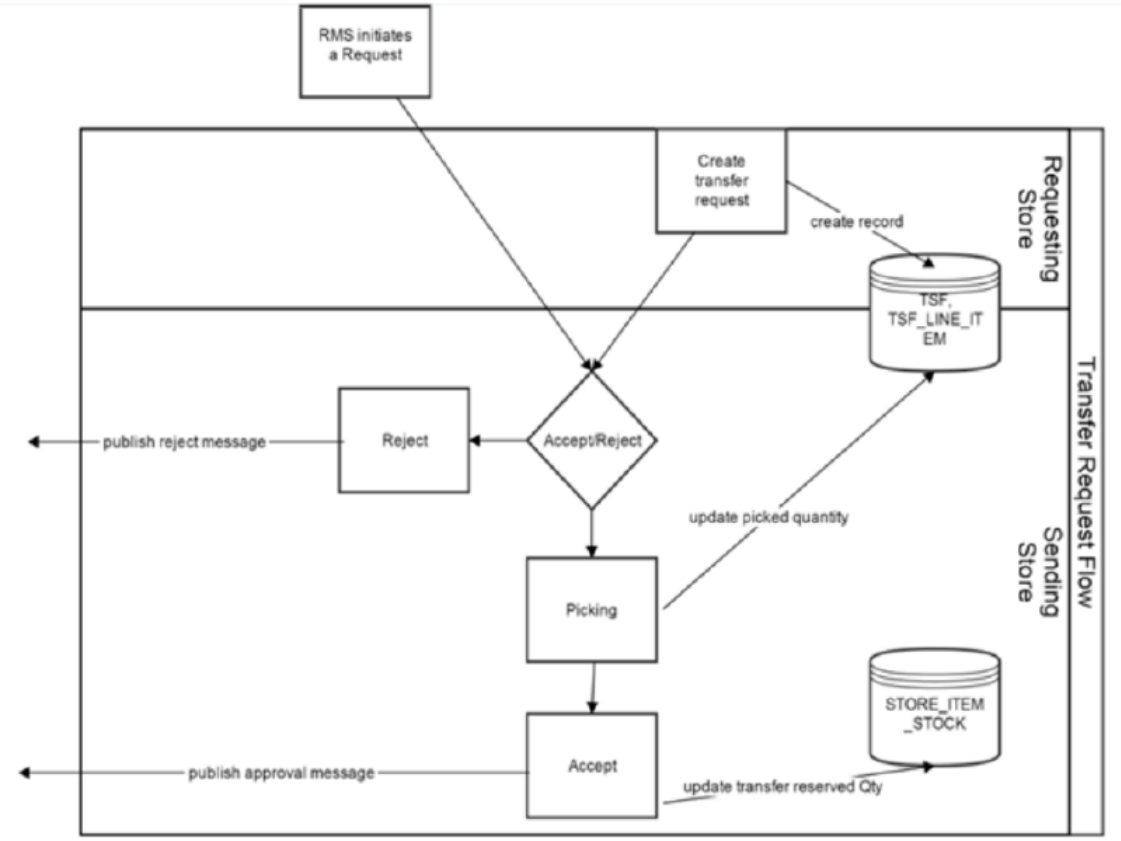


Figure 7-2 Transfer Create Flow

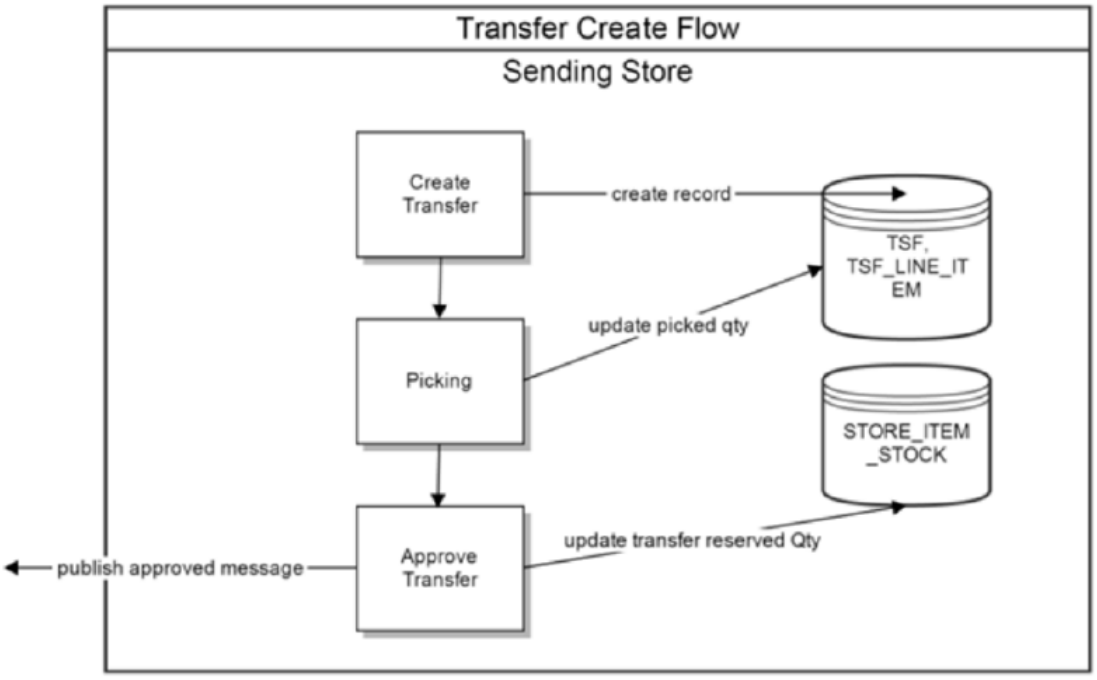


Figure 7-3 Transfer Shipment Creation Flow

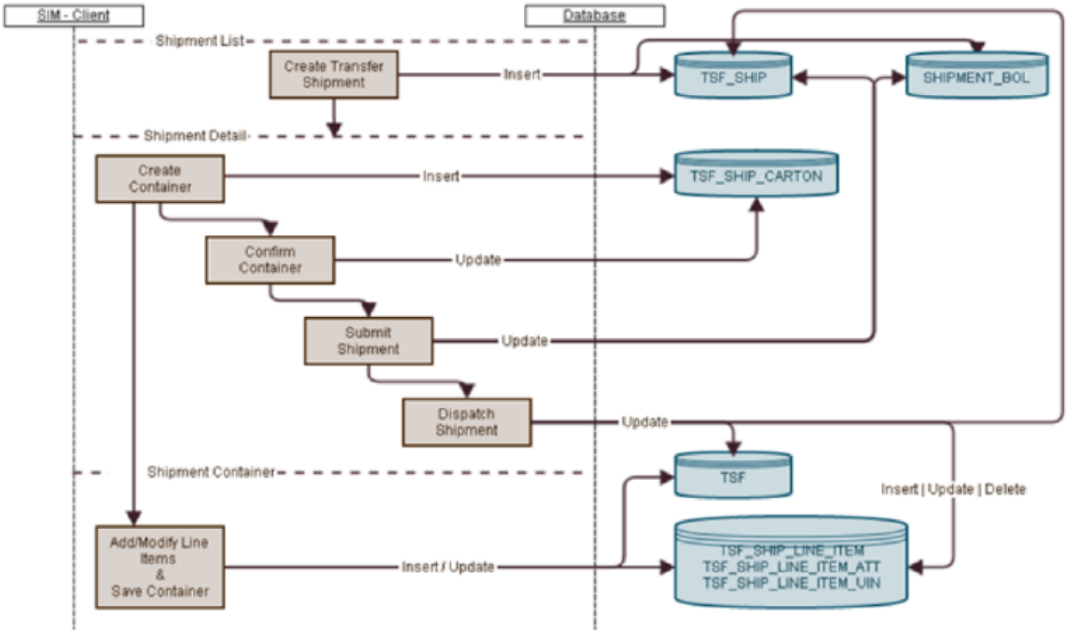
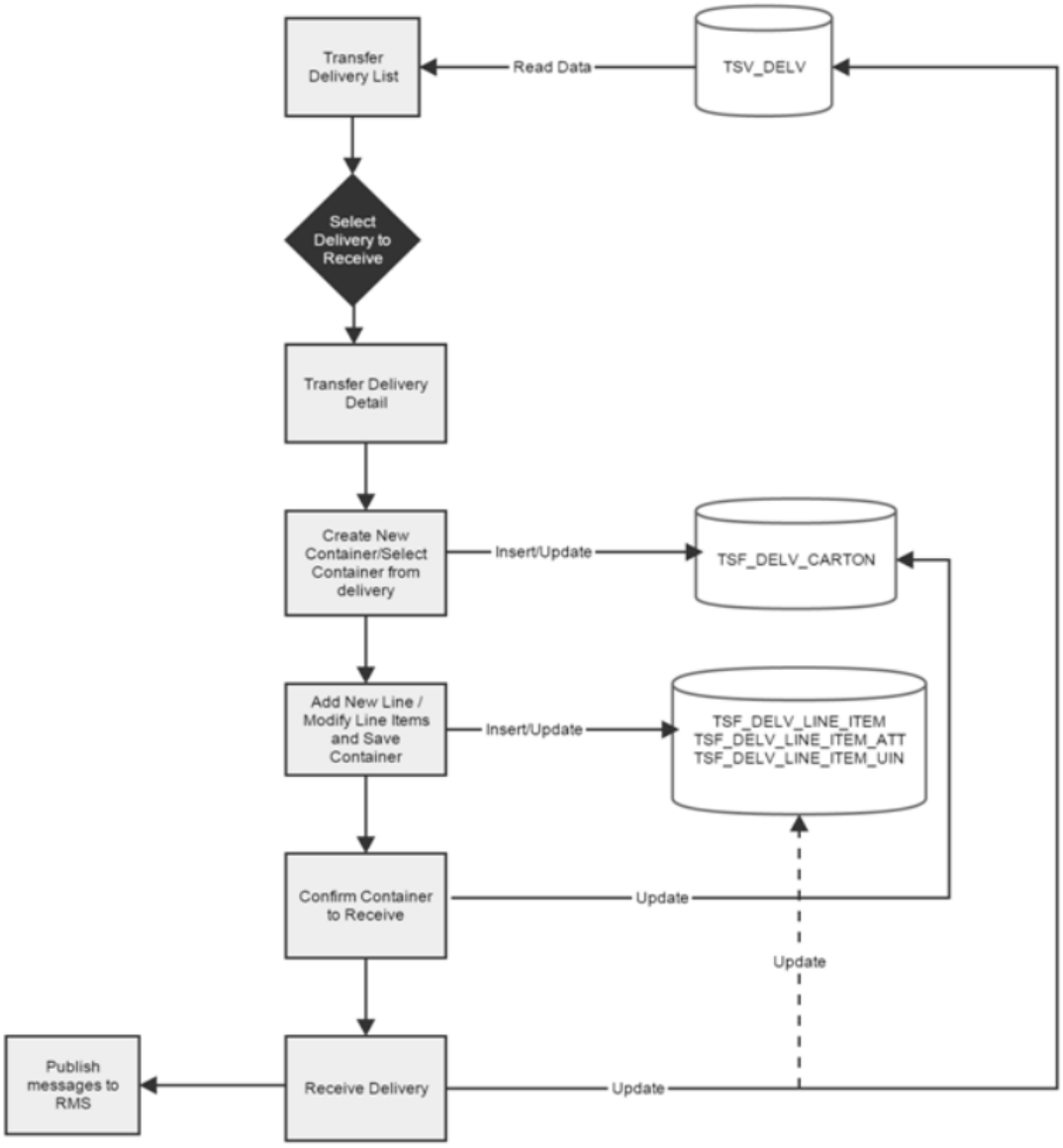


Figure 7-4 Transfer Receiving Process Flow



Transfer Doc

RIB Payload	Description
SODesc	This message is received from external systems when a stock order/ transfer has been created
SOStatusDesc	This message is received from external systems when a stock order/ transfer has been modified.
SORef	This message is received from external systems when a stock order/ transfer has been deleted.

Transfer Shipment

RIB Payload	Description
ASNOutDesc	This message is sent to external systems when the transfer shipment is dispatched.
ManifestCloseVo	This message is received from an external system to indicate physical shipment has been accepted. This will attempt to auto-close the transfer shipment if all items are shipped.
ManifestDesc	This message is sent to an external system when manifesting is activated, and a transfer shipping container is confirmed.
ShipInfoDesc	This message is sent to an external system when pre-shipment notifications are active, and a transfer shipment is either submitted or dispatched (without previously being submitted).
SOStatusDesc	This message is sent to an external system when a transfer shipment container is saved with shipping quantities. It is also sent when a transfer shipment container is canceled but had shipping quantities. Increase and decrease of quantities is indicated by the SI or SD codes.

Transfer Receiving

RIB Payload	Description
ASNInDesc	Sent from external system to indicate a delivery is tracking place. It creates a transfer delivery record within EICS when a store location is involved.
ReceiptDesc	Sent to external system when a transfer delivery is confirmed. Sent from external warehouse system when a transfer delivery is received at the warehouse.

Vendor Return

RTV Creation

RTVs can only be created by a request from MERCHANDISING:

Each vendor return will have one or more items.

RTV Shipment

Each RTV shipment will tie back to a single vendor return document.

RTV shipment can be created in two ways:

- From an externally initiated approved vendor return document.
- Creation of ad hoc vendor return shipment which will create an approved vendor return on the fly.

Each vendor return shipment will have one or more containers; each container in turn will have one or more items.

EICS may publish messages when the following happens:

- RTV shipment container is updated, and saved (Return To Vendor Publish)
- RTV shipment is cancelled or rejected (Return To Vendor Publish)
- RTV shipment is dispatched (Return to Vendor Publish and Ship Info Desc Publish, if dispatched without submitting)
- RTV shipment is submitted (Ship Info Desc Publish)
- RTV shipment container is confirmed (RTV manifesting, if configured)
- RTV shipment is submitted (Pre-shipment notification, if configured)

Figure 7-5 RTV Creation Flow

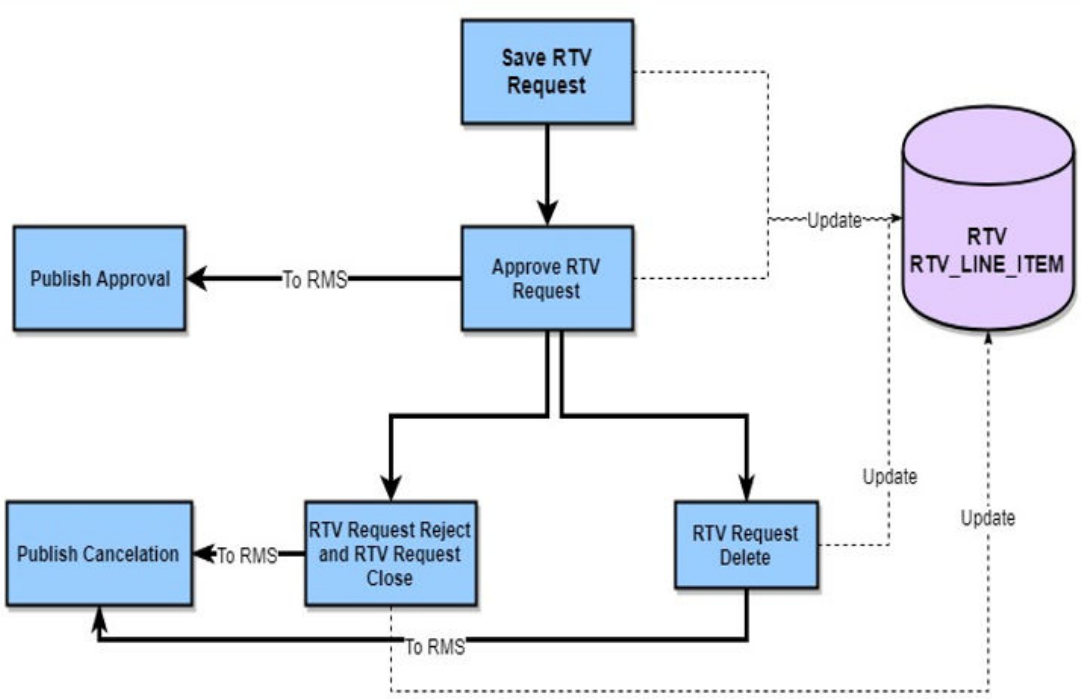


Figure 7-6 RTV Shipment Flow

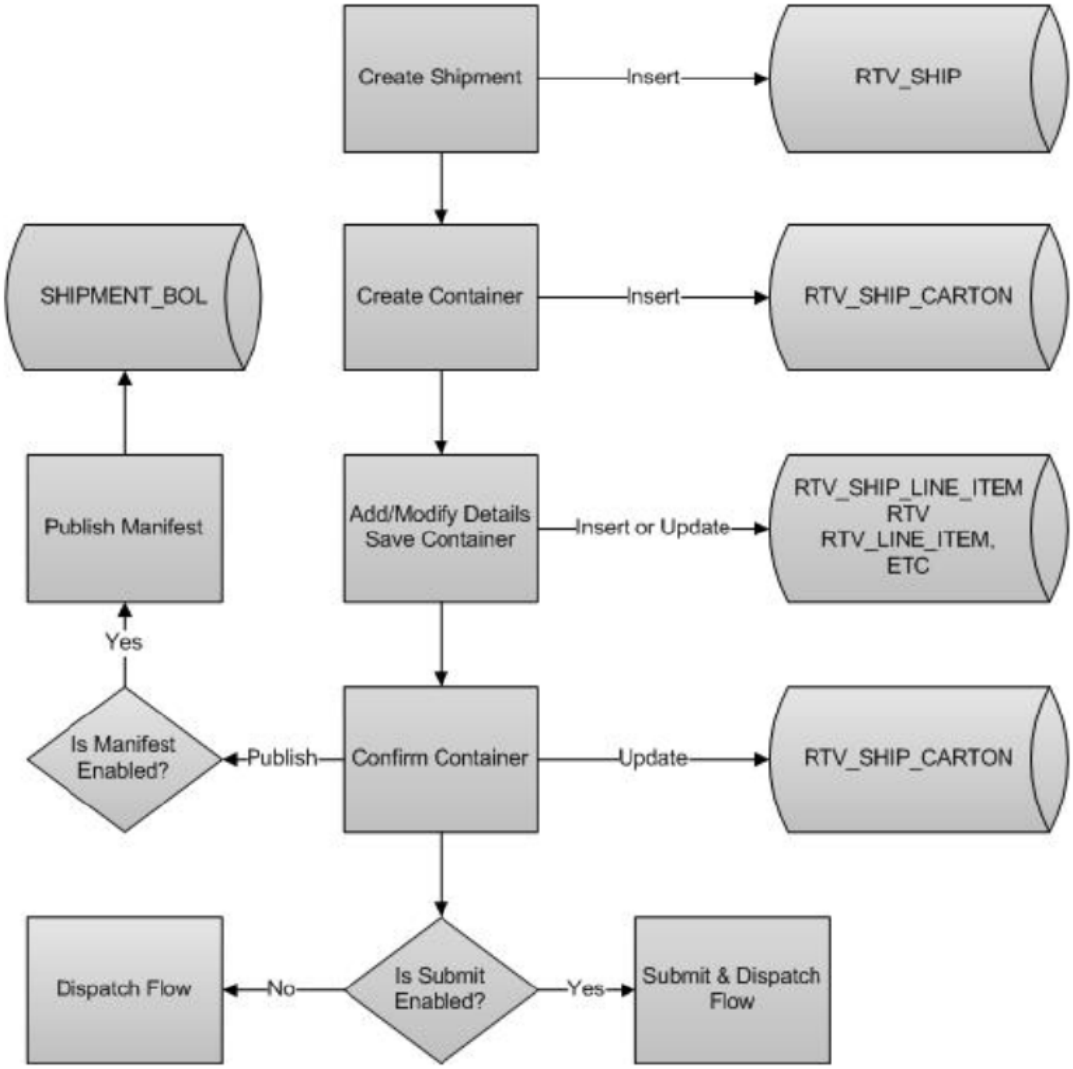


Figure 7-7 RTV Shipment Submit and Dispatch Flow

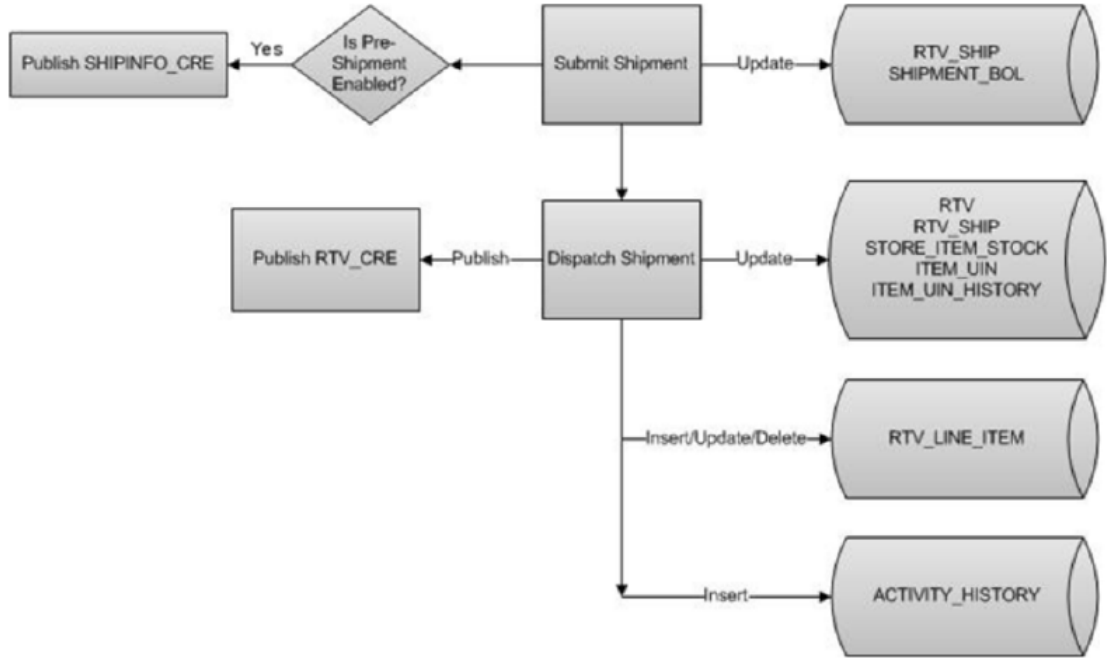
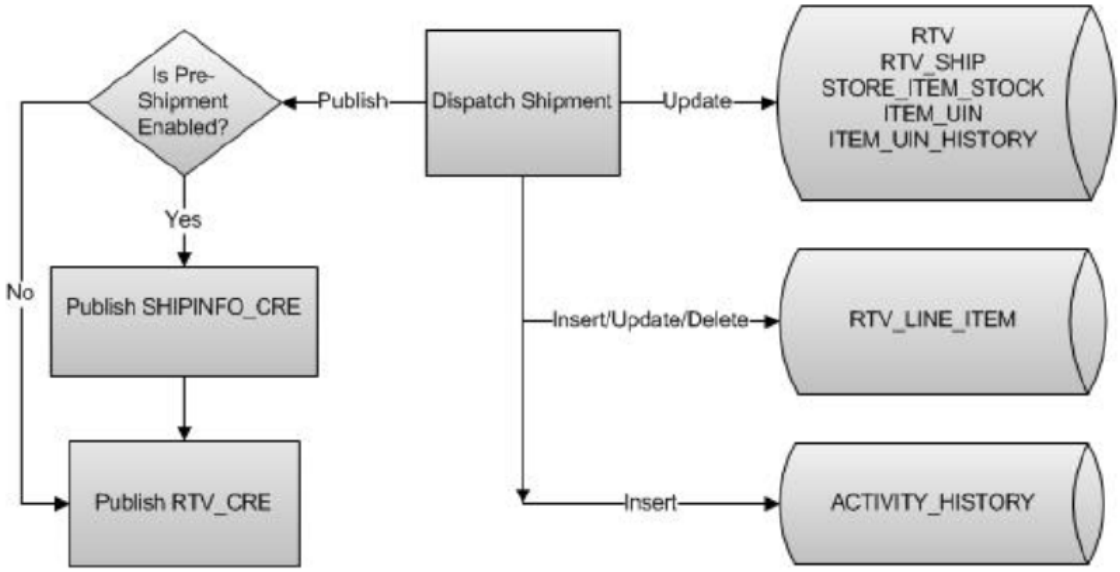


Figure 7-8 RTV Shipment Dispatch Flow



The following payloads are used in RTV operations.

RIB Payload	Description
RTVReqDesc	This payload is sent from an external system to indicate a request for a vendor return. It creates or updates a vendor return document within EICS. It contains a series of RTVReqDtl.

RIB Payload	Description
RTVReqDtl	This payload contains the detailed information about the items on the vendor return.
RTVReqRef	This payload contains reference information about a vendor return when an external system wishes to attempt to cancel the return.
RTVDesc	This payload is sent from EICS to external systems when an RTV shipment is dispatched. This payload is sent from external warehouse system for vendor returns originating at warehouse.

SOAP Web Services

EICS provides a large range of web services to manage the processing of information that is controlled within EICS. Each web service covers a topical area of functionality within EICS and contains numerous operations within to accomplish this functionality. This document is only meant as an outline or summary into using EICS web services and assumes the user has access to the fully documented APIs through the publishing of the web services themselves.

- Security Considerations
- Functionality
- Available Web Services
- Web Services Basic Design Principles
- Internally Managed vs Externally Managed
- Web Service Operation Basic Design Standards
- Interpreting Validation Errors

> **Note:**
>
> The WSDL files are available to download from My Oracle Support (MOS) Document 2614551.1.

Security Considerations

The SOAP web services provided by EICS are secured by Policy A using Oracle WebLogic WS-Policy configurations defined in the xml files included in Oracle WebLogic:

- Policy A
  - **Description**: Message must be sent over SSL and requires authentication of a plain text UsernameToken.
  - **Configuration**: Wssp1.2-2007-Https-UsernameToken-Plain.xml

Customers should create IDCS or OCI IAM user and the user should be assigned integration_users IDCS or OCI IAM application role to access the web-service endpoints.

See *Oracle Retail Enterprise Inventory Cloud Service Security Guide* and *Oracle Retail Enterprise Inventory Cloud Service User Guide* - Security chapter.

For REST web service security see REST WEB Services Security Considerations later in this guide.

Functionality

This document is intended to be used by someone who has read and understands all the functional areas and business functionality described in the *Oracle Retail EICS User Guide* and *Oracle Retail EICS Administration Guide.*

Available Web Services

The following list contains a summary of the web services available in EICS.

Web Service	Description
ActivityLock	This service is used to manage the locking of data within EICS. Data needs to be locked to be updated securely.
FulfillmentOrderDelivery	This service is used to manage fulfillment order deliveries (outgoing shipment to customers). It allows the creation, cancellation, and dispatch of deliveries.
FulfillmentOrderPick	This service is used to manage fulfillment order picking within EICS. It allows the creation, deletion, and confirmation of a pick to complete a fulfillment order.
FulfillmentOrderReversePick	This service is used to manage fulfillment order reverse picking within EICS. It allows the creation, update, deletion, and confirmation of a reverse pick.
InventoryAdjustment	This service is used to manage inventory adjustments within EICS. It allows the creation, update, cancellation, and confirmation of inventory adjustments.
ItemBasket	This service is used to manage item baskets within EICS. It allows the creation, update, and removal of item baskets.
OrderRequest	This service is used to create, read, update, approve, cancel and lookup store orders.
POSTransaction	This service processes external point-of-sale transactions updating the inventory accordingly. A point-of-sale is considered an externally managed transaction (internally and externally managed transaction are covered later in this document).
ProductGroup	This service is used to create or update a product group.
ProductGroupSchedule	This service is used to create, update, or cancel a product group schedule.
ReplenishmentGap	This service is used to create, update, or delete a replenishment gap.
RfidInventory	This service is used to create, update, or delete a RFID facility zone. It is also used to refresh inventory and to process RFID events.
ShelfAdjustment	This service is used to create, update, cancel or confirm a shelf adjustment.
ShelfReplenishment	This service is used to create, update, cancel or confirm a shelf replenishment.
StockCount	This service is used to retrieve the details of a stock count or a stock count child (section of stock count).
Store	This service is used to retrieve information about stores such as store detail, associated stores, or transfer zones.

Web Service	Description
StoreFulfillmentOrder	This service is used to manage fulfillment orders within EICS. It allows for the cancellation and rejection of orders and items.
StoreInventory	This service is used to lookup information about inventory positions and has several different operations to do so.
StoreInventoryISN	This service is used to create, update, or delete ISN data in EICS.
StoreInventoryUIN	This service is used to create, update, generate or read a UINs.
StoreItem	This service is used to lookup various information about an item within the store.
StoreItemPrice	This service is used to lookup prices about items within a store.
StoreNotification	This service is used to create new notifications within the system.
StoreShipmentManifest	This service is used to close documents based on shipped container information.
StoreShipmentReason	This service is used to retrieve shipment reasons codes to use when creating shipments.
StoreTicket	This service is used to create tickets and lookup ticket formats.
StoreTransfer	This service is used to create, update, and request a transfer, which describes the intent to ship items to another store or to a warehouse. It is also used to approve or reject that request. It can be used to directly create, update, approve, cancel, or close an actual transfer.
TransferDelivery	This service is used to update, receive, or confirm a transfer delivery (delivery arriving from another store or warehouse). It is also used to create, update, receive, cancel, or confirm the containers on that delivery.
TransferShipment	This service is used to create, update, submit, or dispatch a transfer shipment (shipment going out to another store or warehouse). It is also used to create, update, cancel, or confirm the containers on that shipment.
VendorDelivery	This service is used to update, receive, reject, or confirm a vendor delivery (delivery arriving from a supplier). It is also used to create, update, cancel, or confirm the containers on that delivery.
VendorReturn	This service is used to create, update, approve, cancel, or close a vendor return document, which describes the intent to ship items to a supplier.
VendorShipment	This service is used to create, update, open, submit, cancel submit or dispatch a vendor shipment (outgoing shipment to a supplier). It is also used to create, update, cancel, submit, or confirm the containers on that shipment.

Web Services Basic Design Principles

Empty Response

In the cast that a web service does not return any information (an empty list), the external system needs to understand that this is a valid response that indicates no item, transaction or queried information was found or retrieved. For example, performing a lookup in which the search criteria entered matched no input.

Error Return Key

Errors returned through a web service will be in the form of a key. This key should be translated into correct language and verbiage by the external system. EICS will not do this translation or provide English verbiage for the encountered web service error.

Boolean Data Type

If a Boolean is the data type on the interface to EICS, and no value is provided, EICS will default the value to False.

Configured System Options in EICS

Web services apply system configurations to the request that are coming in through the web service but assumes that all input validation that requires user interaction to confirm has been completed by the consumer of the web service (the third party system). This system configuration user-interaction option will be assumed to have been confirmed during the web service processing. In case the system option is a fixed restriction that does not require user interaction, and the input fails this restriction, the web service will return an error. For example:

- Shipping inventory when inventory is less than 0 can be allowed by the user of EICS. The web service assumes that the third party application did prompt the user or that their business always allows the user to do this activity.
- Adding a non-ranged item requires both a system configuration option to be enabled and the user to confirm the process. If the system configuration does not allow it, the web service will block the transaction and return an error. If the system configuration does allow adding non-ranged items, it is automatically assumed that a user confirmed its addition, and the web service adds the item.
- Allowing Receiver Unit Adjustments are dependent on a period of time. If a receiver unit adjustment were to come into EICS after that period, it would automatically be rejected, and the web service would return an error regardless of presentation or confirmation of user done by the external system.

Internally Managed vs Externally Managed

Internally Initiated

Internally initiated indicates the EICS was responsible for the original creation of the transaction being processed. A web service that creates a new transaction within EICS to be managed creates an internally initiated transaction.

Externally Initiated

Externally initiated indicates that another system created the transaction, has information about it, and notifies EICS of its creation through a notification system, not by requesting EICS create new information. EICS might manage the data after the notification but did not create the data.

Internally Managed

Internally managed data is information in which EICS is responsible for tracking its state and processing its life cycle. Our deliveries and shipments are primary examples of this. They may be externally initiated or internally initiated, but either way, they are internally managed. EICS is responsible for approving, picking, packing, manifesting, and dispatching the system and internally manages that process.

Externally Managed

Externally managed data is information that EICS does not process or track and is simply informed about after the externally managed data is complete. Point-of-sale transactions are a perfect example of this. We do not manage the sale, but once it is complete, EICS is notified and adjusts the inventory accordingly.

Web Services

EICS web services are intended for integration to allow a system using those services to control the flow and processing within EICS. Our web services are primarily designed (almost all of them) to internally manage the information. The services are intended to be used real time with the steps such as approving, picking, and dispatching occurring with real time access to EICS web services while the process is happening.

EICS web services are not designed for externally managed information. If a system is controlling the state managements itself and not informing EICS until later, this will produce out-of-sync inventory. For example, if you create a shipment, pack the shipment, and send it out and then a day later use the web service, to create, update, and dispatch the shipment, all dates and processing of inventory movements will be tagged with the later date as if they occurred real time when the web service is used.

The point-of-sale service is an externally managed service, where the timestamp on the service can be any date and EICS handles the logic of dating things according to that timestamp. Inventory Adjustment also has an "adjustment date" which represents the time the adjustment took place and so the movement of inventory can be controlled externally.

Web Service Operation Basic Design Standards

This section discusses the general approach and design standards for naming and intent regarding operations within a web service.

Lookup

Lookup operations take either an identifier of a set of criteria and find all the relevant records associated to it. A thin or light view of the data being asked for is returned giving reference to information you can do further interrogation on.

Read

Read operations take an identifier and return all relevant information to it. It may only be one level, however. For example, reading a transfer shipment returns only all the information at the shipment level and does not read information at the container or item level. Usually, the entity that contains items will also retrieve the items. Reading a container will return the container information and the item information within.

Create

Create usually inserts and generates something new and returns an identifier, reference, or handle to that information. Create normally does not take a great deal of information, such as items or anything, but rather gives you a set of IDs that then lets you update the transaction with that reference.

Save or Update

Save or update is used to modify the data usually without changing state on the transaction. The save or update operation is used to add items, remove items, edit attributes, change quantities and all the other tasks one does during a process.

Approve, Cancel, Confirm or Dispatch

Activities that change state take in a simple identifier and then process that state change. To dispatch a shipment, you pass in a reference only to the shipment and it becomes shipped, updating the inventory. This means all changes are done through the save operations prior to making the state change.

Interpreting Validation Errors

If some data could not be processed, the web service will return a fault or a validation fault. The general form that a fault will take is to be a series of problem detail nodes containing a key and value that describes the fault. The first problem detail node will have the key ERROR and the value will be a description of the error type such as INVALID_INPUT. This will be followed by a series of nodes where the KEY is an object class name (ex: Transfer) and the value is its identifier (ex: 123) describing the hierarchy of data the error took place in. For example, a transfer container fault would have two nodes (Transfer:123) and then (TransferCarton:456). If a specific attribute is known, the final node in any problem detail series, it will have the key ATTRIBUTE and the value will be the name of the attribute of the error (ex: ITEM_ID:A5X).

Problem Detail Name	Value
ERROR	This describes the error (for example: INVALID_INPUT)
ATTRIBUTE	Identifies the specific attribute that had an error.

EICS follows the same business rules when processing information from a web service as it does from any of its clients, so the same business rules and functionality that exist in the User's Guide also exists for the web service. Understanding the basic functionality will help interpret why the validation or processing error occurred.

Common Error Codes

The following codes are paired as values to the ERROR Key:

Error Code	Description
ACTIVITY_LOCK_NOT_GRANTED	Indicates that a requested activity lock on a piece of data was not granted.
DUPLICATE_INPUT	Indicates the service would create a duplication of input that should be unique.
INVALID_DATE_RANGE	Indicates the end date of a date range is prior to the start date.
INVALID_INPUT	Indicates that the input is invalid. This error is usually followed by object and attribute information.
INVALID_ITEM	Indicates the item does not exist in the system.
INVALID_STATE_FOR_UPDATE	Indicates the transaction or data specified is not in a state that allows it to be updated (such as canceled).

Error Code	Description
INPUT_MISMATCH	Indicates the input to the web service has been altered incorrectly when compared to existing data. For example, the store identifier is different on the web service request than the currently existing transaction.
INPUT_TOO_LARGE	Indicates the input in the web service is larger than is allowed in the transaction date.
ITEM_NOT_RANGED	Indicates the item has not been activated in the location for which the request is made.
MULTIPLE_STORE	Indicates a batch of input data (such as a point-of-sale transaction) was for more than one store in a single web service call.
TIMEZONE_NOT_GMT	Indicates the time input of the web services was not in GMT.
UOM_MISMATCH	Indicates a mismatch of unit of measure information between the input and currently existing data that does not allow the information to be accurately merged.

Validation Error (Fault Example)

<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>VALIDATION_ERROR</faultstring>

<detail>

<ns0:ValidationWSFaultException xmlns:ns0="http://www.oracle.com/retail/integration/services/exception/v1">

<ns0:shortErrorMessage>VALIDATION_ERROR</ns0:shortErrorMessage>

<ns0:BusinessProblemDetail>

<ns0:problemDescription>VALIDATION_ERROR</ns0:problemDescription>

<ns0:ProblemDetailEntry>

<ns0:name>ERROR</ns0:name>

<ns0:value>INVALID_INPUT</ns0:value>

</ns0:ProblemDetailEntry>

<ns0:ProblemDetailEntry>

<ns0:name>ShlfAdjRef</ns0:name>

<ns0:value>1</ns0:value>

</ns0:ProblemDetailEntry>

<ns0:ProblemDetailEntry>

<ns0:name>ATTRIBUTE</ns0:name>

<ns0:value>shelfAdjustmentId</ns0:value>

</ns0:ProblemDetailEntry>

</ns0:BusinessProblemDetail>

</ns0:ValidationWSFaultException>

</detail>

</ns0:Fault>

</S:Body>

</S:Envelope>

Business Error (Fault Example)

<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>BUSINESS_ERROR</faultstring>

<detail>

<ns0:ValidationWSFaultException xmlns:ns0="http://www.oracle.com/retail/integration/services/exception/v1">

<ns0:shortErrorMessage>BUSINESS_ERROR</ns0:shortErrorMessage>

<ns0:BusinessProblemDetail>

<ns0:problemDescription>BUSINESS_ERROR</ns0:problemDescription>

<ns0:ProblemDetailEntry>

<ns0:name>ERROR CODE</ns0:name>

<ns0:value>ADJUSTMENT_NOT_FOUND</ns0:value>

</ns0:ProblemDetailEntry>

</ns0:BusinessProblemDetail>

</ns0:ValidationWSFaultException>

</detail>

</ns0:Fault>

</S:Body>

</S:Envelope>

Web Services

Web services available in EICS:

ActivityLock

The following operations are available within the ActivityLock web service.

Operation	Description
lookupActivityLock	Retrieves information about one or more activity locks that match the input criteria.
readActivityLock	Retrieves detailed information about a single lock using its identifying reference.
createActivityLock	Created an activity lock on a transaction.
deleteActivityLock	Deletes an activity lock thereby releasing processing on a transaction.

Standard Usage

An activity lock is a record indicating the user, time, and a piece of information (a transaction) that should be considered "locked". All server processing validates that the accessing user has a lock on the information before updating, notifying the current user if someone else has modified the information while they were locked and preventing the stale update.

Developers should create locks on information prior to performing update calls and delete locks when the update if finished. For example, create a lock on inventory adjustment with ID 123 with the ActivityLock service, then use `saveInventoryAdjustment` in the Inventory Adjustment service with Adjustment 123, and then delete the activity lock using the ActivityLock service. If you do not gain the lock, you will receive an error when attempting to save an inventory adjustment.

FulfillmentOrderDelivery

The following operations are available within the FulfillmentOrderDelivery web service.

Operation	Description
lookupFulfillmentOrderDeliveryHeaders	Retrieves summary information for fulfillment order deliveries that match the search criteria input.
readFulfillmentOrderDeliveryDetail	Reads the complete detailed information about a fulfillment order including items and quantities.
createFulfillmentOrderDelivery	Creates a new fulfillment order delivery including items and quantities in an in-progress status to be further worked on.
cancelFulfillmentOrderDeliverySubmission	Cancels the fulfillment order review and moves it back into in-progress status for further work.
dispatchFulfillmentOrderDelivery	Dispatches the fulfilment order delivery completing the delivery and updating the inventory.
submitFulfillmentOrderDelivery	Submits the fulfillment order delivery for review prior to dispatching.
updateFulfillmentOrderDelivery	Updates a fulfillment order delivery including items and quantities. This operation requires an activity lock.
lookupCustomAttributeAdmins	Retrieves the custom attribute administration information that describes what customized attributes are available on the fulfillment order delivery.

Standard Usage

A user can create a delivery by using `createFulfillmentOrderDelivery` references the fulfillment order to make a delivery for. The user can then use `updateFulfillmentOrderDelivery` to fill in all the quantities that are going to be shipped and finally use `dispatchFullfillmentOrderDelivery` to indicate that the order has been shipped out, which moves the inventory appropriately.

FulfillmentOrderPick

The following operations are available within the FulfillmentOrderPick web service.

Operation	Description
lookupFulfillmentOrderPickHeaders	Retrieves summary information for fulfillment order picks that match the search criteria input.
readFulfillmentOrderPick	Reads the complete detailed information about a fulfillment order pick including items and quantities.
confirmFulfillmentOrderPick	Confirm the fulfillment order pick which allows it to move on to the delivery cycle.
deleteFulfillmentOrderPick	Deletes a fulfillment order pick.
createFulfillmentOrderPickByFulfillmentOrder	Generate a pick based on the information in a fulfillment order.
createFulfillmentOrderPickByBin	Generate a pick based on a number of bins selecting orders as needed to fill the bins.
updateFulfillmentOrderPick	Update the item and quantity information about a pick. This operation requires an activity lock.
lookupCustomAttributeAdmins	Retrieves the custom attribute administration information that describes what customized attributes are available on the fulfillment order pick.

Standard Usage

Picking is used to reserve or set aside quantities for a later delivery. The user can create a pick for an order using `createFulfillmentOrderPickByFulfillmentOrder` or create a bin to places multiple orders in with `createFulfillmentOrderPickByBin`. The picked quantities can be updated through the `updateFullfillmentOrderPick` operation and when the pick is finished, it can be finalized with `confirmFulfillmentOrderPick` which sets assigned the goods as reserved in inventory.

FulfillmentOrderReversePick

The following operations are available within the FulfillmentOrderReversePick web service.

Operation	Description
lookupReversePickHeaders	Retrieves summary information for fulfillment order reverse picks that match the search criteria input.
lreadReversePickDetail	Reads the complete detailed information about a fulfillment order reverse pick including items and quantities.
createReversePick	Creates a new fulfillment order reverse pick for the specified fulfillment order.
deleteReversePick	Deletes a fulfillment order reverse pick.

Operation	Description
updateFulfillmentOrderReversePick	Updates the items and quantities on a fulfillment order reverse pick. This operation requests an activity lock.
confirmReversePick	Confirms the fulfillment order reverse pick completing the process and assigning the inventory back to a location within the store system.
lookupCustomAttributeAdmins	Retrieves the custom attribute administration information that describes what customized attributes are available on the fulfillment order reverse pick.

Standard Usage

Reverse Picking is used to take reserved quantities and place them back into available inventory. The user can create a reverse pick with `createReversePick`. The quantities to return can be updated through the `updateFulfillmentOrderReversePick` operation and when the reverse pick is ready, it can be finalized with `confirmReversePick` which moves reserved inventory back into available inventory.

InventoryAdjustment

The following operations are available within the InventoryAdjustment web service.

Operation	Description
lookupInventoryAdjustmentReason	Retrieve a complete list of adjustment reasons that can be used when updating or saving an inventory adjustment. Reason codes are attached to each line item.
lookupNonSellableQuantityType	Retrieve a complete list of non-sellable quantity types. These codes indicate the reason that unavailable inventory in unavailable.
lookupInventoryAdjustmentHeader	Retrieve summary information about inventory adjustment transactions based on the search criteria sent.
readInventoryAdjustmentDetail	Retrieve the complete detailed information about an inventory adjustment, including its item information, based on a unique reference/id.
saveInventoryAdjustment	Creates or updates the information about an inventory adjustment in the data store. You can alter information about items and quantities using this operation. This operation requires having an activity lock.
confirmInventoryAdjustment	Confirms the inventory adjustment, updating all the inventory positions, and closing the adjustment.
saveAndConfirmInventoryAdjustment	Performs the functionality of saveInventoryAdjustment and immediately thereafter performs the confirmInventoryAdjustment functionality. See those operations.
cancelInventoryAdjustment	Cancel an inventory adjustment. This can only be done prior to the inventory adjustment being confirmed.
lookupCustomAttributeAdmins	Retrieves the custom attribute administration information that describes what customized attributes are available on the inventory adjustment.

Standard Usage

A new inventory adjustment can be created using the `saveInventoryAdjustment` operation. Alternatively, the user can `lookupInventoryAdjustmentHeader` to find a specific inventory adjustment to work on. Either way, `saveInventoryAdjustment` can be used to update the information on an open adjustment. The `lookupInventoryAdjustmentReasons` will retrieve the

reasons codes that need to be assigned to items when you update an adjustment. When the adjustment contains all the information you need, the `confirmInventoryAdjustment` operation will finalize the inventory adjustment and shift the inventory appropriately.

ItemBasket

The following operations are available within the Item Basket web service.

Operation	Description
lookupItemBasketHeaders	Retrieve a list of item basket headers based on search criteria which contain summary information about the item basket.
lookupItemBasketTypes	Retrieve a complete list of item basket types to use when creating a new item basket.
createItemBasket	Creates a new item basket.
readItemBasket	Retrieve the complete detailed information about an item basket based on an identifier.
deleteItemBasket	Cancels an item basket. The basket will no longer be usable and will be marked for eventual purge from the data store. This operation requires an activity lock.
saveItemBasket	Updates an item basket. This operation requires an activity lock.
copyItemBasket	Creates a new item basket with the same information as an existing item basket.
confirmItemBasket	Moves the item basket to a completed state and allows it to be used within logic throughout the system. This operation requires an activity lock.
lookupCustomAttributeAdmins	Retrieves the custom attribute administration information that describes what customized attributes are available on the item basket.

Standard Usage

A new item basket can be created using the `saveItemBasket` operation. Alternatively, the user can `lookupItemBasketHeader` and `readItemBasket` to find a specific item basket to work on. Either way, `saveItemBasket` can be used to update the information on an item basket. When the item basket contains all the information you need, the `confirmItemBasket` operation will finalize the item basket and make it available to use in other areas of the system.

OrderRequest

The following operations are available within the Order Request web service.

Operation	Description
lookupOrderRequestHeader	Retrieves store order request headers based on the query criteria.
readOrderRequest	Retrieves detailed information about a store order request.
createOrderRequest	Creates a new store order request.
updateOrderRequest	Updates an existing store order request.
approveOrderRequest	Approve a store order request.
cancelOrderRequest	Cancels a store order request.
lookupDeliveryTimeSlot	Retrieves delivery time slots.
lookupOrderContext	Retrieves contexts available for store order requests.

Operation	Description
lookupOrderArea	Retrieves store order request areas that could be used for restriction.
lookupCustomAttributeAdmins	Retrieves all the custom attributes admins configured for store order requests.

Standard Usage

A new store order can be created using the `createOrderRequest` operation. The information about store order can be read by `readOrderRequest`. The store order can be updated using `updateOrderRequest` and can be approved using `approveOrderRequest` or can be canceled using `cancelOrderRequest`. The `lookupOrderRequestHeader` is used to find the store orders.

POSTransaction

The following operations are available within the POSTransaction web service.

Operation	Description
processPOSTransactions	Processes a point-of-sale transaction or transactions through an asynchronous process. This is designed to optimize the processing at 500 PosTrnItm (across any number of transactions).

Standard Usage

POS may integrate its transactions to EICS using this web service. The service processes point-of-sale transactions through an asynchronous process. This service has a default limit of 1000 total PosTrnItms, though they may be distributed between any number of actual PosTrn transactions. Exceeding this limit causes a web service fault to occur. However, the web service is optimized for speed at greater than 400 and less than 500 total PosTrnItms per service call. These transactions may belong to multiple store identifiers. The processing operation validates the input, parses the payload information, creates a POSTransaction object within EICS, and stores these records to be processed later. See Sales Integration for additional information.

REST Web Service

A REST web service for POSTransaction exists and is the preferred service to use in order to process point-of-sale transactions (see REST WEB Services). This SOAP based web service will be deprecated and eventually removed.

ProductGroup

The following operations are available within the ProductGroup web service.

Operation	Description
lookupProductGroupHeader	Retrieves list of summary information about a product group that match the search criteria input.
readProductGroup	Retrieves the detailed information about a single product group based on its unique reference.
saveProductGroup	Creates or updates a product group. The input contains all the detailed information about the product group. An activity lock is needed for this operation.

Standard Usage

With this web service, the user can create or update the contents of a product group, a collection of items associated with a certain type of grouping, such as stock counts. The user can find the product group with `lookupProductGroupHeader`, read in the entire product group with `readProductGroup` and then, if the group is still open, update the contents of the product group with `saveProductGroup`.

ProductGroupSchedule

The following operations are available within the ProductGroupSchedule web service.

Operation	Description
lookupProductGroupScheduleHeader	Retrieves list of summary information about a product group schedule that match the search criteria input.
readProductGroupSchedule	Retrieves the detailed information about a single product group schedule based on its unique reference.
saveProductGroupSchedule	Creates or updates a product group. The input contains all the detailed information about the product group schedule. An activity lock is needed for this operation.
cancelProductGroupSchedule	Cancels the product group schedule.

Standard Usage

With this web service, the user can create or update the contents of schedule, which uses a product group to generate activity within EICS. The user can find the schedule with `lookupProductGroupScheduleHeader`, read in the entire schedule with `readProductScheduleGroup` and then, if the schedule is still open, update the contents of the schedule with `saveProductGroupSchedule`.

ReplenishmentGap

The following operations are available within the ReplenishmentGap web service.

Operation	Description
lookupReplenishmentGapHeaders	Retrieves list of summary information about replenishment gaps that match the search criteria input
readReplenishmentGap	Retrieves the detailed information about a single replenishment gap based on its unique reference.
saveReplenishmentGap	Creates a new replenishment gap or updates the detailed information about a replenishment gap. If update, this operation requires an activity lock.
deleteReplenishmentGap	Deletes a replenishment gap.
lookupCustomAttributeAdmins	Retrieves the custom attribute administration information that describes what customized attributes are available on the replenishment gap.

Standard Usage

With this web service, the user can create or update the contents of replenishment gap list which can then be used in creation of shelf replenishment within EICS. A new replenishment gap list can be created using `saveReplenishmentGap`. The user can update existing

replenishment gap list with `saveReplenishmentGap`, find replenishment gap lists with `lookupReplenishmentGapHeaders`, read in the entire replenishment gap list with `readReplenishmentGap` and delete a replenishment gap list with `deleteReplenishmentGap`.

RfidInventory

The following operations are available within the RfidInventory web service.

Operation	Description
deleteRfidZone	Deletes a zone within a facility. A zone cannot be deleted if RFID tags still exist within the zone.
lookupRfidZones	Returns details about all the zones within a particular facility.
processRfidEvents	Processes Radio-Frequency-Identification based events.
saveRfidZone	Creates or updates the details of a facility zone.

Standard Usage

With this web service, the user can create or update RFID zones within EICS. A new RFID zone can be created using `saveRfidZone`. The user can update an existing RFID zone with `saveRfidZone`, find RFID zones with `lookupRfidZones` and delete a RFID zone with `deleteRfidZone`. The user can process RFID based events using `processRfidEvents`.

ShelfAdjustment

The following operations are available within the ShelfAdjustment web service.

Operation	Description
lookupShelfAdjustmentHeaders	Retrieves list of summary information about shelf adjustments that match the search criteria input.
readShelfAdjustment	Retrieves the detailed information about a single shelf adjustment gap based on its unique reference.
saveShelfAdjustment	Creates a new shelf adjustment or updates the detailed information about a current shelf adjustment. If update, this operation requires an activity lock.
confirmShelfAdjustment	Confirms a shelf adjustment completing the workflow and moving inventory positions.
cancelShelfAdjustment	Deletes a shelf adjustment.
lookupCustomAttributeAdmins	Retrieves the custom attribute administration information that describes what customized attributes are available on the shelf adjustment.

Standard Usage

Shelf adjustments are used to adjust the shop-floor or backroom stock in case of any discrepancy. A new shelf adjustment can be created using `saveShelfAdjustment`. The user can update existing shelf adjustment with `saveShelfAdjustment`, find shelf adjustments with `lookupShelfAdjustmentHeaders`, read in the entire shelf adjustment with `readShelfAdjustment`, cancel a shelf adjustment with `cancelShelfAdjustment` and confirm a shelf adjustment with `confirmShelfAdjustment`.

ShelfReplenishment

The following operations are available within the ShelfReplenishment web service.

Operation	Description
lookupShelfReplenishmentHeaders	Retrieves list of summary information about shelf replenishments that match the search criteria input.
readShelfReplenishment	Retrieves the detailed information about a single shelf replenishment gap based on its unique reference.
createShelfReplenishment	Creates a new shelf replenishment.
updateShelfReplenishment	Updates the detailed information about a current shelf replenishment. This operation requires an activity lock.
confirmShelfReplenishment	Confirms a shelf replenishment completing the workflow and moving inventory positions.
cancelShelfReplenishment	Deletes a shelf replenishment.
lookupCustomAttributeAdmins	Retrieves the custom attribute administration information that describes what customized attributes are available on the shelf replenishment.

Standard Usage

Shelf replenishment is used to replenish shop-floor stock from backroom or delivery bay. A new shelf replenishment can be created with `createShelfReplenishment`. The user can find shelf replenishments with `lookupShelfReplenishmentHeaders`, read in the entire shelf replenishment with `readShelfReplenishment`, update the shelf replenishment with `updateShelfReplenishment`, confirm the shelf replenishment with `confirmShelfReplenishment` and cancel the shelf replenishment with `cancelShelfReplenishment`.

StockCount

The following operations are available within the StockCount web service.

Operation	Description
lookupStockCountHeaders	Retrieves list of summary information about a stock count that match the search criteria input.
readStockCountDetail	Retrieves the detailed information about a single stock count based on its unique reference. This contains a list of summary information about the child counts.
readStockCountChild	Retrieves the detailed information about a single stock count child.
activateStockCount	This activates are starts the stock counting process including taking a snapshot of current inventory positions.
completeStockCountChild	Completes the counting or recounting of a stock count child, depending on which phase the stock count is in. This process will calculate discrepancies and move the child to the next phase.
updateCountQuantities	Updates the counted or recounted quantity fields for a stock count child based on the current phase of the stock count.
lookupCustomAttributeAdmins	Retrieves the custom attribute administration information that describes what customized attributes are available on the stock count.

Standard Usage

The stock count web services are design primarily to export information for third party counting. You first lookup the headers, choose your stock count, and then retrieve all the details for the stock count. These details do not contain item information but rather a list of child count references. You can use these references to grab the full details of a child count which includes items and quantities, and then update those quantities.

REST Web Service

A StockCount REST web service exists that allows for the snapshot of a stock count (see REST WEB Services).

Store

The following operations are available within the Store web service.

Operation	Description
lookupAutoReceiveStore	Retrieves all stores that allow auto-receiving of inventory from the input store.
lookupAssociatedStore	Retrieves all stores that are associated to the input store. They are sometimes called buddy stores.
lookupStoresInTransferZone	Retrieves all stores in the same transfer zone as the input store.
readStoreDetail	Retrieves the detailed information about a single store from the input unique reference.

Standard Usage

The Store web service is used to retrieve information about stores. There are no updates. They are used to determine such information as whether you can ship to certain stores (such as those in transfer zones).

StoreFulfillmentOrder

The following operations are available within the StoreFulfillmentOrder web service.

Operation	Description
lookuFulfillmentOrdersHeaders	Retrieves summary information for fulfillment orders that match the search criteria input.
readFulfillmentOrderDetail	Reads the complete detailed information about a fulfillment order including items and quantities.
createFulfillmentOrderDetail	Creates a new fulfillment order with detailed information, including items and quantities.
cancelFulfillmentOrderDetail	Cancels quantities on a fulfillment order. This may cancel the entire order or just reduce or cancel quantities for specific items.
rejectFulfillmentOrder	Rejects the fulfillment order indicating that the store will be unable to fulfill that order.
lookupCustomAttributeAdmins	Retrieves the custom attribute administration information that describes what customized attributes are available on the fulfillment order.

Standard Usage

Unlike some of the other web services, fulfillment order is not managed within EICS. Instead, EICs manages the picking and delivery, but the order itself is managed by an external order management system.

Oracle Retail Order Broker (OB) calls SIOCS for inventory availability.

Web services are supplied to find and read the details of a fulfillment order, but updates are not allowed. Instead, the external system uses `createFulfillmentOrderDetail` to notify EICS of a new order to ship, `cancelFulfillmentOrderDetail` to reduce or cancel quantities (note that they cannot be increased) or call `rejectFulfillmentOrder` to notify EICS that the order has been rejected.

StoreInventory

The following operations are available within the StoreInventory web service.

Operation	Description
lookupAvailableInventory	Retrieves basic availability information for multiple items at multiple locations. Only transaction-levels items are processed (UPCs are not allowed) and only current inventory is returned. The service supports up to 200 items at 150 locations.
lookupAvailableInventoryAllStores	Retrieves basic availability information for a single item at all store locations. Only transaction-levels items are processed (UPCs are not allowed) and only current inventory is returned.
lookupAvailableInventoryAllWarehouses	Retrieves inventory information for a single item at multiple warehouses. Only transaction-level items are processed, and only current inventory is returned.
lookupInventoryInStore	Retrieves a broad set of inventory information for several items at several stores, broken down into various inventory groupings.
lookupInventoryInTransferZone	Retrieves a broad set of inventory information for items within the specific transfer zone, broken down into various inventory groupings.
lookupInventoryForBuddyStores	Retrieves a broad set of inventory information for associated or buddy stores, broken down into various inventory groupings.
lookupFutureInventory	Retrieves the future inventory information (such as inbound, ordered quantities and expected dates) for an item and store location.

Standard Usage

The StoreInventory is meant to retrieve inventory position information. Available inventory lookups are much smaller and quicker to respond than full inventory lookups. Future inventory is separated from current positions as it is much more time consuming to retrieve. Those who access the web services should consider the purpose before choosing which operation to use.

REST Web Service

An InventoryInquiry REST web service exists for inventory lookup and is the preferred service to use in order to retrieve inventory information (see REST WEB Services). This SOAP based web service will be deprecated and eventually removed.

StoreInventoryISN

The following operations are available within the StoreInventoryISN web service.

Operation	Description
lookupIsnTypes	Returns a complete list of Item Scan Number types.
lookupIsn	Returns details about matching Item Scan Numbers in store inventory.
createIsn	Create a new Item Scan Number without changing store inventory.
updateIsn	Updates an existing Item Scan Number without changing store inventory.
deleteIsn	Deletes an Item Scan Number without changing store inventory.
lookupCustomAttributeAdmins	Retrieves all the custom attribute admins configured for ISNs.

Standard Usage

This web service is used to create, update, or delete ISN in store inventory. An item scan number is any number meant to be scanned to find an item, and potentially a Unique Identification Number, that is not already an item, UPC, UIN, VPN, or other value. Items Scan Numbers are only used to find information and are not tracked as inventory.

StoreInventoryUIN

The following operations are available within the StoreInventoryUIN web service.

Operation	Description
createUIN	Create a new UIN without changing store inventory.
generateUIN	Generate new UINs without changing store inventory.
lookupUINDetails	Returns details about all the UINs in store inventory for a particular item and store. This is limited to 1000 UINs for a particular item and store.
readUINDetail	Returns details about a UIN in store inventory. A UIN reference is not unique, so this may return detailed information for UINs across multiple items.
updateUIN	Updates an existing UIN without changing store inventory.

Standard Usage

This web service is used to create, generate, update, find, or read UINs in store inventory.

StoreItem

The following operations are available within the StoreItem web service.

Operation	Description
lookupItemHeaderByItem	Retrieves list of summary information about an item that match the item-based search criteria input.
lookupItemHeaderBySource	Retrieves list of summary information about an item that match the source or location-based search criteria input.
lookupItemHeaderByUDA	Retrieves list of summary information about an item that match the UDA (User Defined Attribute)-based search criteria input.
lookupItemHeaderByInventory	Retrieves list of summary information about an item that match the inventory-based search criteria input.

Operation	Description
lookupItemCfa	Retrieve a list of custom flexible attributes for the specified item and store.
lookupItemUda	Retrieve a list of user defined attributes for the specified item and store.
readItemDetail	Retrieves the complete detailed information a single item based on its unique reference.
lookupRelatedItem	Retrieves a list of summary information about items related to the item used as input criteria.
saveItemImage	Inserts a new display image or QR code image for the specified item. The service returns immediately, and the information is processed asynchronously.

Standard Usage

This web service is used to find items and retrieve information about items. The only exception is the ability to create new image-based information about an item.

StoreItemPrice

The following operations are available within the StoreItemPrice web service.

Operation	Description
lookupItemPriceHeader	Retrieve a summary list of item price information based on input criteria. This only retrieves information known to EICS and has no access to a pricing system.
readItemPrice	Retrieves the full details a single item price record based on its unique reference.
lookupItemPriceOnEffectiveDate	Retrieves the item price of an item for a specific date.

Standard Usage

This web service is used to retrieve information about prices that are known to EICS. Integration with pricing systems updates EICS information about item prices on a continual basis. These web services give a view into EICS information only.

StoreNotification

The following operations are available within the StoreNotification web service.

Operation	Description
createNotification	Creates a new notification within the system. These notifications are displayed in the client applications.

Standard Usage

This web service is designed for external system that handle related activities to EICS. With this web service, they can send notifications into EICS of activity that needs to take place based on something that has occurred in another system.

StoreShipmentManifest

The following operations are available within the StoreShipmentManifest web service.

Operation	Description
closeManifest	Closes the manifest shipments.

Standard Usage

This web service is designed to close manifest shipments. All manifest shipments matching the input criteria, such like carrier code, and carrier service code will be closed.

StoreShipmentReason

The following operations are available within the StoreShipmentReason web service.

Operation	Description
lookupAllShipmentReasons	Retrieves all the shipment reasons configured for store shipments.

Standard Usage

This web service exists to allow customers to retrieve information about shipment reasons that can be assigned to line items on outgoing shipments. The shipment based web services taking the code identifier and thus, you will need to read in these shipment reasons to be able to select and apply valid reason codes.

StoreTicket

The following operations are available within the StoreTicket web service.

Operation	Description
createTickets	Create a new group of up to 999 tickets to be managed and printed.
lookupTicketFormats	Retrieves available ticket formats for the criteria specified.

Standard Usage

The `createTickets` operation is used to create a new group up to 999 tickets to be managed and printed. The ticket formats can be retrieved using `lookupTicketFormats` operation based on the criteria specified.

StoreTransfer

The following operations are available within the StoreTransfer web service.

Operation	Description
lookupTransferHeader	Retrieve a summary list of transfers that matches the input criteria.
lookupTransferContext	Retrieves all the transfer context options available to assign to a transfer.

Operation	Description
readTransfer	Retrieves the detailed information about transfer, including its items and quantities, based on a unique reference.
createTransferRequest	Creates a brand new transfer request (Location 1 requesting a transfer from Location 2).
saveTransferRequest	Updates a transfer request allowing user to change items and quantities. This must be done prior to requesting it, which finalizes the transfer request. This requires an activity lock.
createTransfer	Generates a new transfer that you can add details to. The saveTransfer method must be used to update details such as items and quantities of the transfer.
saveTransfer	Updates a previously approved transfer item and quantity details. This operation requires an activity lock.
saveTransferApproval	Updates items and quantities on a transfer in requested status that is currently in the process of being approved but has not yet been approved. This operation requires an activity lock.
requestTransfer	Updates the status to Requested, finally the transfer request. This allows the opposite location to view the new request for transfer of goods. This operation requires an activity lock.
approveTransfer	Approves a transfer request converted the transfer request into an approved transfer. This operation requires having an activity lock.
rejectTransfer	Rejects a transfer in request status which prevents the transfer request from becoming a transfer. This operation requires having an activity lock.
cancelTransfer	Cancels an approved transfer. This operation requires having an activity lock.
closeTransfer	Closes a processed or partially processed transfer finalizing the state of the transfer. This operation requires having an activity lock.
lookupCustomAttributeAdmins	Retrieves the custom attribute administration information that describes what customized attributes are available on the transfer.

Standard Usage

The process is started by one store creating a transfer request from a shipping store using `createTransferRequest`. The requesting store can continue modifying the transfer request using `saveTransferRequest` until it is ready to notify the shipping store, when it then uses the `requestTransfer` to send the request to the shipping store. The shipping store can then begin picking items for the transfer and updating the transfer using the `saveTransferApproval` operation. When all the quantities the shipping store are willing to ship are determined, the shipping store uses `approveTransfer` to finalize the approval of the transfer. Alternatively, they can choose to reject the transfer using `rejectTransfer`. It is possible for a shipping store to create a transfer document without going through the request and approval process by using `createTransfer` and `saveTransfer`.

TransferDelivery

The following operations are available within the TransferDelivery web service.

Operation	Description
lookupTransferDeliveryHeaders	Retrieves basic information about one or more transfer deliveries that match the criteria specified. This operation is used to find a delivery arriving at the store.
readTransferDeliveryDetail	Retrieves the entire set of information about a transfer delivery header based on the identifier you pass to it.
updateTransferDelivery	Updates the header information on a transfer delivery. This operation requires an activity lock.
receiveTransferDelivery	Receives all the currently open and active containers on a transfer delivery by defaulting quantities into all the unreceived items. This does not move inventory, only defaults quantities. This operation requires an activity lock.
confirmTransferDelivery	Confirms a transfer delivery receiving the goods into inventory and updating all the inventory positions. This moves the transfer delivery to a completed status. This operation requires an activity lock.
lookupTransferDeliveryContainerHeaders	Retrieves summary information about every container on a transfer delivery based on the unique delivery reference.
readTransferDeliveryContainerDetail	Reads the entire details of a container including items and quantities based on a unique container reference.
createTransferDeliveryContainer	Generates a new container on the transfer delivery and returns a reference to use so that items and quantity can be added later.
updateTransferDeliveryContainer	Updates the items and quantities on a transfer delivery container. This operation requires an activity lock.
receiveandConfirmTransferDeliveryContainer	It first defaults receiving quantity on the items within the container and then executes the same locking as the confirmTransferDeliveryContainer. This operation requires an activity lock.
confirmTransferDeliveryContainer	Confirms a transfer delivery container as received and updates all the inventory positions. This operation requires an activity lock.
cancelTranferDeliveryContainer	Cancels a transfer delivery container moving it to missing status. Changes cannot be made to a canceled container.
openTransferDeliveryContainer	Re-opens an already confirmed container moving it back into in-progress status.
lookupTransferDeliveryOrders	Retrieves any customer orders associated with the transfer delivery based on the delivery's unique reference.
lookupMisdirected TransferDeliveryContainers	Retrieves summary information about containers that may have been misdirected based on a set of search criteria as input into the operation.
lookupCustomAttributeAdmins	Retrieves the custom attribute administration information that describes what customized attributes are available on the transfer delivery.

Standard Usage

After reading a transfer delivery using `lookupTransferDeliveryHeader`, you can read the header detail with `readTransferDelivery` or container list with `lookupTransferDeliveryContainers`. You can then use `updateTransferDelivery` to update header attributes and `updateTransferDeliveryContainer` to update items and quantities in the container. To quickly receive the quantities, `receiveTransferDeliveryContainer` automatically fills in quantities, and when quantities are entered `confirmTransferDeliveryContainer` finalizes the container (and if appropriate configurations and business rules apply) immediately updates the inventory. If `receiveTransferDelivery` or

confirmTransferDelivery is used, then all containers will either be received or confirmed respectively.

TransferShipment

The following operations are available within the TransferShipment web service.

Operation	Description
lookupTransferShipmentHeader	Retrieves basic information about one or more transfer shipments that match the criteria specified. This operation is used to find a shipment.
readTransferShipmentDetail	Retrieves the entire set of information about a transfer shipment header based on a unique reference.
createTransferShipment	Creates a new and empty transfer shipment and returns a reference to the shipment.
saveTransferShipment	Updates the information on a transfer shipment header.
submitTransferShipment	Submits the transfer shipment for review before final dispatch.
cancelSubmittedTransferShipment	Cancels the submission of the transfer shipment for review.
dispatchTransferShipment	Dispatches a transfer shipment. This moves the shipment to dispatched state and updates the inventory. A transfer shipment cannot be modified after dispatch. Dispatch should occur only after all containers are confirmed.
cancelTransferShipment	Cancels a transfer shipment.
lookupTransferShipmentContainer	Finds all the containers on a specific shipment and retrieves basic identification information about each container.
readTransferShipmentContainer	Reads the specific and complete contents of a container.
createTransferShipmentContainer	Creates a new transfer shipment container on the shipment and returns a reference to it.
saveTransferShipmentContainer	Updates the information about a transfer shipment container including adding and removing items and quantities.
confirmTransferShipmentContainer	Confirms that a transfer shipment container is ready for shipment and marks the container as no longer editable.
cancelTransferShipmentContainer	Cancels a transfer shipment container on the shipment.
openTransferShipmentContainer	Re-opens a confirmed container on a shipment prior to the shipment being dispatched so that changes can be made to the container.
lookupCustomAttributeAdmins	Retrieves the custom attribute administration information that describes what customized attributes are available on the transfer shipment.

Standard Usage

To create a shipment for a transfer document, lookup the transfer shipment using lookupTransferShipmentHeader. If it does not exist, you may create one for the document using createTransferShipment. Create a container on the shipment using createTransferShipmentContainer and update the container with items and quantities using saveTransferShipmentContainer. Confirm the container using confirmTransferShipmentContainer. Repeat the process for each container as needed. Once all containers are confirmed, if configured to require submittal, submit the shipment using submitTransferShipment and finally, dispatch the shipment using dispatchTransferShipment. Dispatching the shipment finalizes the shipment and relieves the inventory.

VendorDelivery

The following operations are available within the VendorDelivery web service.

Operation	Description
lookupVendorDeliveryHeaders	Retrieves basic information about one or more vendor deliveries that match the criteria specified. This operation is used to find a delivery from a supplier.
lookupPurchaseOrderHeaders	Retrieves basic information about one or more purchase orders that match the criteria specified.
readVendorDeliveryDetail	Retrieves the entire set of information about a vendor delivery header based on a unique reference.
createVendorDelivery	Generate a new vendor delivery heaver and returns a referenced to the delivery.
updateVendorDelivery	Updates the information on a vendor delivery header. This does not include containers, items, or quantities. This operation requires an activity lock.
receiveVendorDelivery	Updates the quantities on a vendor delivery filling in any unreceived items within the containers of the delivery with a default value. It "receives" missing quantities, but no inventory positions are updated. This operation requires an activity lock.
confirmVendorDelivery	Confirms the vendor delivery updating inventory positions and completing the delivery. This operation requires an activity lock.
rejectVendorDelivery	Rejects the vendor delivery placing it in rejected status. This operation requires an activity lock.
cancelVendorDelivery	Cancels the vendor delivery placing it in canceled status. This operation requires an activity lock.
lookupVendorDeliveryContainerHeaders	Retrieves summary information about every container on a vendor delivery based on the unique delivery reference.
readVendorDeliveryContainerDetail	Reads the entire details of a container including items and quantities based on a unique container reference.
createVendorDeliveryContainer	Generates a new container on the vendor delivery and returns a reference to use so that items and quantity can be added later.
updateVendorDeliveryContainer	Updates the items and quantities on a vendor delivery container. This operation requires an activity lock.
confirmVendorDeliveryContainer	Confirms a vendor delivery container as received and updates all the inventory positions. This operation requires an activity lock.
cancelVendorDeliveryContainer	Cancels a vendor delivery container moving it to missing status. Changes cannot be made to a canceled container.
openVendorDeliveryContainer	Open Vendor delivery container. This will re-open a container after receipt allowing it to be received again.
lookupVendorDeliveryOrders	Retrieves any customer orders associated with the vendor delivery based on the delivery's unique reference.
lookupVendorDeliveryAdjustments	Retrieves any external receipt adjustments that exist for the delivery based on the specified unique reference.
cancelSubmitVendorDeliveryContainer	Opens a submitted container for further updates, moving the status to in-progress.
submitVendorDeliveryContainer	Moves the status of the container to submitted and prevents further updates. The container may still be confirmed. No inventory positions are updated via this operation.

Operation	Description
lookupCustomAttributeAdmins	Retrieves the custom attribute administration information that describes what customized attributes are available on the vendor delivery.

Standard Usage

After reading a vendor delivery using `lookupVendorDeliveryHeader`, you can read the header detail with `readVendorDelivery` or container list with `lookupVendorDeliveryContainers`. Use `updateVendorDelivery` to update header attributes and `updateVendorDeliveryContainer` to update items and quantities in the container. To quickly receive the quantities, `receiveVendorDeliveryContainer` automatically fills in quantities, and when quantities are complete `confirmVendorDeliveryContainer` finalizes the container and if appropriate configurations and business rules apply, immediately updates the inventory. If `receiveVendorDelivery` or `confirmVendorDelivery` is used, then all containers will either be received or confirmed respectively. Re-opening a container can be done using `openVendorDeliveryContainer`. To prevent further updates to the container without confirming it, use `submitVendorDeliveryContainer`. Submitted container can be re-opened and moved to in-progress status for further updates using `cancelSubmitVendorDeliveryContainer`.

VendorReturn

The following operations are available within the VendorReturn web service.

Operation	Description
lookupVendorReturnHeader	Retrieves basic information about one or more vendors return documents that match the criteria specified.
readVendorReturnDetail	Retrieves the entire set of information about a vendor return, including items and quantities, based on a unique reference.
saveVendorReturn	Updates the entire set of information about a vendor return, including items and quantities. This operation requires an activity lock.
approveVendorReturn	This marks an in-progress vendor return as approve for shipment. This operation requires an activity lock.
cancelVendorReturn	Cancels a vendor return indicating no further items and quantities should be shipped for the return.
closeVendorReturn	Closes a vendor return document moving it from in-progress to canceled, rejected, or complete status depending on the state of shipped quantities.
lookupCustomAttributeAdmins	Retrieves the custom attribute administration information that describes what customized attributes are available on the vendor return.

Standard Usage

The user may access `lookupVendorReturnHeader` to find vendor returns to deal with. Once the proper vendor return is found, `readVendorReturnDetail` will retrieve all the details of the vendor return including items and quantities. The `saveVendorReturn` operation is then used to update quantities that are expected to ship. Once the vendor return reaches its final state, the operation `approveVendorReturn` will approve the return and get it ready for shipment.

VendorShipment

The following operations are available within the VendorShipment web service.

Operation	Description
lookupVendorShipmentHeaders	Retrieves basic information about one or more vendor shipment headers that match the criteria specified.
lookupReturnContext	Retrieves all the context options that are available to assign to a vendor return shipment.
readVendorShipmentDetail	Retrieves the detailed information about a vendor return header based on a unique reference. It does not include information about containers or items.
saveVendorShipment	Creates a new vendor shipment header if not identifying reference is set or updates the vendor shipment header information if a unique reference is sent as part of the date. When used as an update, an activity lock is needed.
submitVendorShipment	Submits the vendor shipment for review before final dispatch.
cancelVendorShipmentSubmission	Cancels the submission of the vendor shipment for review.
cancelVendorShipment	Cancels a vendor shipment. This moves the shipment to canceled status. Changes cannot be made to a canceled shipment.
dispatchVendorShipment	Dispatches a vendor shipment. This moves the shipment to dispatched state and updates the inventory. A vendor shipment cannot be modified after dispatch. Dispatch should occur only after all containers are confirmed. This operation requires an activity lock.
closeVendorShipment	Closes a vendor shipment using business logic to determine its final state. It cancels the shipment of remaining quantities. Changes cannot be made after a shipment is closed.
lookupVendorShipmentContainerHeaders	Retrieves summary information about all containers within a vendor shipment based on the unique reference of the shipment.
readVendorShipment ContainerDetail	Reads the specific details, including items and quantities, about a container specified by its unique reference.
saveVendorShipmentContainer	Update the details of a container, including items and quantities. This operation requires an activity lock.
confirmVendorShipmentContainer	Confirms that the container is ready for shipment. A confirmed container cannot be modified. This operation requires an activity lock.
cancelVendorShipmentContainer	Cancels a container on the shipment removing it from the shipment.
openVendorShipmentContainer	Opens a confirmed container placing it back into in-progress status so that items can be added or removed from the container.
lookupCustomAttributeAdmins	Retrieves the custom attribute administration information that describes what customized attributes are available on the vendor shipment.

Standard Usage

To create a shipment for a vendor return document, lookup the vendor shipment using `lookupVendorShipmentHeader`. If it does not exist, create one using `createVendorShipment`. Next, create a container on the shipment using `createVendorShipmentContainer`. Update the container with items and quantities using `saveVendorShipmentContainer`. Confirm the container using `confirmVendorShipmentContainer`. Repeat the process for each container as needed. Once all containers are confirmed, if configured to require submit, then submit using `submitVendorShipment` or dispatch the shipment using `dispatchVendorShipment`. Dispatching the shipment finalizes the shipment and relieves the inventory.

Enterprise Documentation

Full web service API document in the form of web services description language files can be downloaded in .zip format. Version specific files are available under 'Web Services Description Language Files' section within MOS Document 2614551.1.

REST Web Services

Web services are intended for integration to allow a system using those services to control the flow and processing of data within EICS. There are multiple types of data involved in this integration. Data that is managed by other systems and needs to get into our system, but that EICS does not manage. This includes such concepts as item, stores, and point-of-sale transaction. Data that is managed by EICS includes such ideas as inventory adjustments, transfers, deliveries, and stock counts. Some services will provide ability for external data to get into EICS, some are intended to be used real time such as approving, picking, and dispatching shipments.

- REST WEB Services Security Considerations
- REST WEB Services Basic Design Principles
- Hypertext Transfer Protocol Status Codes
- JSON Error Element and Error Codes
- Integration Error Codes
- Error Code Data Elements

REST WEB Services Security Considerations

The REST web services provided by EICS are secured using OAuth2 tokens and require SSL.

The supported OAuth2 security requires a token requested for the *client_credentials* grant with the EICS integration scope (for example, *rgbu:siocs:integration*).

Note that the scope name differs for each environment.

Please see the REST Web Service OAuth2 Requests for details on requesting tokens.

REST WEB Services Basic Design Principles

- Requests and Responses
- API Versioning
- Content-Type
- JSON Validation
- Synchronous vs Asynchronous
- Configured System Options In EICS
- External vs Internal Attributes
- Dates In Content
- Links In Content

Requests and Responses

When making requests and processing responses from REST web services it is important for the client to handle headers correctly.

The client should always use *Accept* for the appropriate content type when making requests.

The client should always check the response status code and *Content-Type* header before processing a response body.

When reading a payload from the response body, the *Content-Length* header must be used safely and securely along with the *Content-Type*.

This is important even for error responses. It is possible for errors to occur outside of the REST API layer, which may produce different content for the error. In these cases, it is common to get text or HTML content for the response body.

API Versioning

Accept-Version

The REST end points have an optional API versioning feature allowing the client to specify an API version to be accepted.

This may be used by the client to ensure that no calls may be made to a web service that uses an incorrect version number.

For example: *Accept-Version: 22.1.301*

If the web service does not support this API version, then the server will produce a *400 Bad Request* error response.

Content-Type

application/json

The content type of both REST input and returned output is *application/json*.

In the case that no content is included, a content type may not be assigned.

When handling REST service responses, the client must always check the returned *Content-Type* and *Content-Length* before processing the payload.

JSON Validation

When consuming a REST service end point that requires a request payload as JSON, the client is responsible for verifying that the JSON is valid. If invalid data is sent in a request, there may be a server error processing the JSON or it may ignore some fields if the JSON is valid but does not map correctly to the API payload definition.

Always make sure that the client sends valid JSON that is designed to satisfy the API payload definition.

Synchronous vs Asynchronous

Each service API will be defined as synchronous or asynchronous. Both perform JSON validation as described above. If the API is synchronous, the remaining data validation and live updating of the data will take place immediately and the call will be rejected if any business

errors occur. If the API is asynchronous, the data is set aside to be processed later and the REST service is successfully returned noting that the data has been accepted. In the case of asynchronous processing, business error and failed data recovery is monitored and the dealt with outside of the REST web service.

Configured System Options In EICS

Web services apply system configuration to the request that are coming in through a web service but assumes that all in-put validation that requires user interaction to confirm has been completed by the third party system prior to accessing the service. It operates as if the user confirmed any activity. However, if a system option is a fixed restriction that does not require user interaction, and the input fails the restriction, this is always considered an error.

Examples of configurations being applied include:

Shipping inventory when inventory is less than 0 can be allowed by a user of EICS. The web services assumes that the application accessing the service did prompt the user or that their business always allows the user to this activity.

Adding a non-ranged item requires both a system configuration option to be enabled and the user to confirm the addition of the item. If the system configuration does not allow it, the web service will block the transaction and return an error (un-less processing asynchronously). If the system configuration does allow adding non-ranged items, it will automatically assume that a user confirmed this addition and processing will allow the addition of the item.

Allowing Receiver Unit Adjustments is dependent on a period of time. If a receiver unit adjustment were to come into EICS after that period of time, it would automatically be rejected, and the web service would return an error regardless of presentation or confirmation of user done by the external system.

External vs Internal Attributes

EICS web services are EICS centric and track information from an internal application point-of-view. This has ramifications on three types of data: identifiers, dates, and users.

Almost all paths and information will contain an identifier. In almost all cases, this will be an EICS internal identifier generated within our system. If external identifiers also exist for the date, they will be defined as such in the information. For example, you might encounter transferId and externalTransferId as attributes. In some cases, an API only takes an internal identifier, and you may need to use lookup APIs to retrieve an internal identifier using an external identifier as search criteria.

Timestamps are captured at the time an event occurs within EICS as part of EICS's internal tracking and state management. For example, we capture the timestamp when a shipment is created, last updated, and when it is dispatched. These timestamps occur at the time this occurred within EICS. When a REST service is called to create a transaction, such as a shipment, the create timestamp of the shipment will be the moment that service is called. If the shipment is dispatched using the web service, the dispatch date will be the moment that service is called. So if an external system dispatched a shipment two days earlier, and is just now calling the web service, it will not capture the external dispatch time. In some places, you will encounter a date that can be entered as part of the input information (for example, an externalDispatchDate, or simple a transactionTimestamp). If it is part of the input information, then it will be captured as that attribute defined in the API.

The user responsible for actions is often captured as part of transaction information with EICS. Some examples might be the user that created the data, the user the last updated it, or perhaps the user that approved it. In these cases, the user is considered an internal user as is assigned the current session user at the time the activity takes place. When accessing the

REST service, the user will be fiwed as an “External User” to indicate it came from an external system, and not the user that manipulated the information in an external system. If external users are to be captured by the data, there will be independent attribute fields such as externalCreateUser that capture the identity of a user in an external system.

Dates In Content

Dates includes in JSON must be in the following format: 2022-04-19T23:59:59-05:00

Dates included as a query parameter must be in the following format: 220227152543-0700

Links In Content

JSON information for a data object may include links. These are self-referential APIs that defined other APIs that are available with the information. In the example below, when reading an activity lock, the following links were included that define a path to accomplish other calls. HRef lists the basic reference and the “rel” the remainder of the path. So, when you read activity lock (1), you get a delete reference that maps to */activitylock/1/delete*, defining the REST path to delete that lock.

[{

"links" : [{

"href" : "/activitylocks/1",

"rel" : "self"

}, {

"href" : "/activitylocks/1",

"rel" : "delete"

}],

Hypertext Transfer Protocol Status Codes

The following information documents the HTTP status codes that Oracle returns via web services calls.

Success Codes

Successful codes are returned whenever the accessing client call was made without any error in the form or content.

Code	Description
200 OK	The information supplied by the customer was in a correct form. This code is returned when reading a resource or querying information about an existing resource or schema. This response code is used when the access is synchronous.
202 Accepted	The information supplied by the customer as in a correct form. This code is returned when access is asynchronous.
204 No Content	The information supplied by the customer was in a correct form. The request was successful but the API itself never provides information as a response.

Client Failure Codes

Client failure codes indicate the client made an error in their service access and must correct their code or its content to fix the failure.

Code	Description
400 Bad Request	If this code is returned, it indicates the customer made a call with invalid syntax or violated the defined properties of the input information. Detailed information may be returned that further identifies the error.
401 Unauthorized 403 Forbidden	If either of these codes are returned, it indicates the access to service was denied. This may occur if no OAuth2 token was provided, or the token had expired, or the identity the token was generated for did not have sufficient access.
404 Not Found	If this code is returned, it indicates the customer made an erroneous access call against a resource or schema that is not defined.
405 Method Not Allowed	If this code is returned, it indicates that the wrong HTTP method was used to make the call. Please check the API.
406 Not Acceptable	If this code is returned, it indicates the wrong Accept header value was used to make the call. Please check the API.
409 Too Many Requests	If this code is returned, it indicates that the web service has received too many service requests recently. This may indicate a cloud issue requiring support to address or the client is making too many calls too frequently and a solution may be required to avoid the issue.

System Failure Codes

System failure codes are returned whenever the processing server encounters an unexpected or severe failure.

Code	Description
500 Internal Server Code	A server error occurred that did not allow the operation to complete.
502 Bad Gateway 503 Service Unavailable 504 Gateway Timeout	If any of these codes are returned, it indicates an issue with the network or cloud services, which may occur due to either client or cloud networking or infrastructure issues, such as outages.

JSON Error Element and Error Codes

If an error occurs in the form of the content, or during processing of the content, an HTTP error code will be returned along with a series of JSON Error Elements as described here.

Example Error

HTTP Response: 400 Bad Request

{

"errors": [

{

"code": 7,

"description": "Missing Attribute",

"dataElement": "storeId",

"referenceElement": "transactionId",

"referenceValue": "1236"

},

{

"code": 11,

"description": "Element Too Large",

"dataElement": "transactionId",

"dataValue": 128,

"referenceElement": "transactionId",

"referenceValue": "1236"

}

]

}

Error Attribute Definitions

Attribute	Definition
Code	A numeric code indicates the issue. See Integration Error Codes table.
Description	The name of the error or issue.
DataElement	The name of an attribute or element of the JSON structure that failed.
DataValue	The value of the attribute or element that failed, or a piece of information about the element that failed (such as a maximum value).
ReferenceElement	The name of an attribute or element that will help further identify the data element. Most often the containing element one level above the failed elements (such as a transaction header).
ReferenceValue	The value of the attribute or element that will help further identify the data element.

Integration Error Codes

The following table contains a listing of the error codes that can be found within returned error information.

Code	Name	Issue
1	Business Error	A business processing error prevent the service from completing.
2	Date Range Error	The date range has a problem (usually indicates end date is earlier than start date in a date range.
3	Duplicate Error	Indicates duplicate element within the data is not permitted.
4	Forbidden	Access is not allowed to the service.
5	Internal Server Error	A severe error occurred with the service attempting to process the request.

Code	Name	Issue
6	Invalid Input	Most often this indicates that input was included that is not allowed or not needed, however it also doubles a kind of catch-all category.
7	Invalid Format	An input was in an invalid format (most often a date string in a query parameter not being in a valid date format).
8	Invalid Status	A transaction or entity is not in a valid status to proceed with the request.
9	Missing Path Element	A path element defining the path of the resource URL was not present.
10	Missing Attribute	A required attribute was missing on the input to the service.
11	Not Found	A data element in the input could not be found in the system (most often an invalid identifier).
12	No Data Input	No input exists for a service that requires input.
13	No Query Input	No query input exists at all for a query that requires at least one input.
14	Element Too Large	An input was too large (exceeded maximum count or maximum size).
15	Results Too Large	The results of the service were too large to return.

Error Code Data Elements

The following table contains a listing of likely or possible data elements that would be matched with a code. Data element and value may not be returned in all cases.

Code	Name	Data Element	Data Value
1	Business Error	Business exception name/key	Data Value
2	Date Range Error	Date element name	Value of date
3	Duplicate Error	Duplicate element name	Duplicated Value
4	Forbidden	N/A	-
5	Internal Server Error	N/A	-
6	Invalid Input	Element name	Value of element
7	Invalid Format	Element name	Value of element
8	Invalid Status	Element name	Status of element
9	Missing Path Element	Missing element	-
10	Missing Attribute	Required element name	-
11	Not Found	Element not found	Value of element not found
12	No Data Input	Missing element	-
13	No Query Input	N/A	-
14	Element Too Large	Element name	Allowed size limit
15	Results Too Large	N/A	-

REST Service: Activity Lock

This service allows the creation, removal, and finding of activity locks.

An activity lock is a record indicating the user, time, and a piece of information (a transaction) that should be considered "locked". All server processing validates that the accessing user has a lock on the information before updating, notifying the current user if someone else has modified the information while they were locked and preventing the stale update.

Developers should create locks on transactional information prior to performing update calls and delete locks when the update if finished. For example, create a lock on inventory adjustment with ID 123 with the ActivityLock service, then use `saveInventoryAdjustment`in the Inventory Adjustment service with Adjustment 123, and then delete the activity lock using the ActivityLock service. If you do not gain the lock, you will receive an error when attempting to save an inventory adjustment.

Service Base URL

The Cloud service base URL follows the format:

https://<external_load_balancer>/<cust_env>/siocs-int-services/api/activitylocks

API Definitions

API	Description
Find Lock	Search for activity lock information based on a set of criteria.
Create Lock	Create a user activity lock.
Delete Lock	Remove a user activity lock.
Read Lock	Retrieve complete information about an activity lock.

API: Find Lock

Searches for locks based on input criteria. At least one input criteria should be provided.

If the number of activity locks found exceeds 10,000, a maximum limit error will be returned. Additional or more limiting search criteria will be required.

API Basics

Endpoint URL	{base URL}/find
Method	POST
Successful Response	200 OK
Processing Type	Synchronous
Input	Criteria
Output	List of activity locks
Max Response Limit	10,000

Input Data Definition

Attribute	Data Type	Required	Description
activityType	String (40)	No	A type of activity that is locked (see Additional Data Definition).
sessionId	String (128)	No	The unique identifier of the session that owns the lock.

Attribute	Data Type	Required	Description
deviceType	Integer (4)	No	The device type (see Additional Data Definition).
userName	String (128)	No	The unique identifier of the user that owns the lock.
lockDateFrom	Date	No	Start date of a range during which the activity was locked.
lockDateTo	Date	No	End date of a range during which the activity was locked.

Example Input

{

"activityType": "3",

"sessionId": "sessionTest",

"deviceType":3

}

Output Data Definition

Attribute	Data Type	Description
lockId	Long	The identifier of an activity lock.
sessionId	String	The identifier of the session that owns the lock.
activityId	String	The identifier of the activity that is locked.
activityType	Integer	The type of activity that is locked.
deviceType	Integer	The device type.
userName	String	The identifier of the user that owns the lock.
lockDate	Date	The date the activity was locked.

Example Output

[{

"links" : [{

"href" : "/activitylocks/1",

"rel" : "self"

}, {

"href" : "/activitylocks/1",

"rel" : "delete"

}],

"lockId" : 1,

"sessionId" : "sessionTest",

"activityId" : "2",

"activityType" : 3,

"deviceType" : 3,

"userName" : "admin",

"lockDate" : "2023-01-04T08:59:41-06:00"

}]

Additional Data Definitions

Location Type

Value	Definition
1	Bill Of Lading
2	Direct Delivery Invoice
3	Fulfilment Order
4	Fulfilment Order Delivery
5	Fulfilment Order Pick
6	Fulfilment Order Reverse Pick
7	Inventory Adjustment
8	Inventory Adjustment Reason
9	Item Basket
10	Item Request
11	POS Transaction Resolution
12	Price Change
13	Product Basket
14	Product Group
15	Product Group Schedule
16	Replenishment Gap
17	Shelf Adjustment
18	Shelf Replenishment
19	Shipment Reason
20	Stock Count Child
21	Store Order
22	Ticket
23	Ticket Format Basket
24	Transaction Event
25	Transfer
26	Transfer Delivery Carton
27	Transfer Shipment Carton
28	Vendor Delivery Carton
29	Vendor Shipment Carton
30	Vendor Return

Device Type

Value	Definition
1	Client
2	Server
3	Integration Service

API: Create Lock

Used to create a new user transaction activity lock. This prevents two users from simultaneously changing the same data.

API Basics

Endpoint URL	{base URL}
Method	POST
Successful Response	200 OK
Processing Type	Synchronous
Input	Activity Lock object
Output	Activity lock identifier
Max Response Limit	N/A

Input Data Definition

Attribute	Data Type	Required	Description
sessionId	String(128)	Yes	The unique identifier of the session that owns the lock.
activityId	String(128)	Yes	The unique identifier of the activity that is locked. This is often a primary identifier of a transaction.
activityType	Integer	Yes	The type of activity that is locked. This is often a transaction type (see Additional Data Definition).

Example Input

{

"activityType": 5,

"sessionId": "session01",

"activityId":"35"

}

Output Data Definition

Attribute	Data Type	Required	Description
lockId	Long	Yes	The unique identifier if a new activity lock is created, or null if the lock already exists.

Example Output

{

"lockId" : 2

}

API: Delete Lock

Used to remove a lock and indicates the activity should no longer be restricted and another user can now begin activity on that data.

It will remove the lock if it exists and perform no action if the lock does not currently exist. In either case, it returns 204 No Content.

API Basics

Endpoint URL	{base URL}/{activityLockId}/delete
Method	DELETE
Successful Response	204 No Content
Processing Type	Synchronous
Input	None
Output	None

Attribute	Description
activityLockId	The activity lock identifier to be removed.

API: Read Lock

Used to retrieve full information about a lock.

API Basics

Endpoint URL	{base URL}/{activityLockId}
Method	GET
Successful Response	200 OK
Processing Type	Synchronous
Input	None
Output	Activity Lock record
Max Response Limit	N/A

Attribute	Description
activityLockId	The activity lock identifier to be read.

Output Data Definition

Attribute	Data Type	Description
lockId	Long	The identifier of an activity lock.
sessionId	String	The identifier of the session that owns the lock.
activityId	String	The identifier of the activity that is locked.
activityType	Integer	The type of activity that is locked.
deviceType	Integer	The device type.
userName	String	The identifier of the user that owns the lock.
lockDate	Date	The date the activity was locked.

Example Output

{

"lockId": 4,

"sessionId": "session01",

"activityId": "37",

"activityType": 5,

"deviceType": 3,

"userName": "dev",

"lockDate": "2023-01-04T23:52:08-06:00"

REST Service: Address

This service integrates address foundation data. Asynchronous address integration is processed through staged messages and is controlled by the MPS Work Type: DcsStore.

Service Base URL

The Cloud service base URL follows the format:

https://<external_load_balancer>/<cust_env>/siocs-int-services/api/addresses

API Definitions

API	Description
readAddress	Read the information about a single address.
importAddress	Create or update the information about a single address.
deleteAddress	Deletes a single address.

API: Import Address

Import a series of addresses. This allows up to 1,000 addresses before an input too large error is returned.

A "Forbidden" response will occur if application is integrated with MFCS or RMS.

API Basics

Endpoint URL	{base URL}/import
Method	POST
Successful Response	202 Accepted
Processing Type	Asynchronous
Input	Addresses list
Output	None
Max Response Limit	N/A

Input Data Definition

Attribute	Data Type	Required	Description
addresses	List of details	Yes	A list of addresses to import.

Address Detail Data Definition

Attribute	Data Type	Required	Description
addressId	String(25)	Yes	The external identifier of the address.
entityType	Integer	Yes	Donates the type of location of the entity (see Additional Data Definition).
entityId	Long(10)	Yes	The external identifier of the entity (this will also match internal identifiers).
addressType	Integer	Yes	The type of address: (see Additional Data Definition).
primary	Boolean		True if this is the primary address of the entity, false otherwise
addressLine1	String(240)	Yes	The first line of the address
addressLine2	String(240		The second line of the address
addressLine3	String(240)		The third line of the address
city	String(120)	Yes	The city of the address
state	String(3)		The state of the address
countryCode	String(3)	Yes	The country code of the address (used by supplier)
postalCode	String(30)		The postal code of the address

county	String(250)	The county of the address
companyName	String(120)	A company name associated with that address
contactName	String(120)	Contact name for that address
contactPhone	String(20)	Contact phone number for that address
contactFax	String(20)	Contact fax number for that address
contactEmail	String(100)	Contact email for that address
firstName	String(120)	A first name of a contact at that address
lastName	String(120)	A last name of the contact at that address
phoneticFirstName	String(120)	A phonetic spelling of a first name of a contact at that address
phoneticLastName	String(120)	A phonetic spelling of a last name of a contact at that address
supplierLocation	String(120)	Supplier location information

API: Delete Address

Deletes a single address.

A "Forbidden" response will occur if application is integrated with MFCS or RMS.

API Basics

Endpoint URL	{base URL}/{addressId}/delete
Method	POST
Successful Response	202 Accepted
Processing Type	Asynchronous
Input	None
Output	None
Max Response Limit	N/A

Path Parameter Definitions

Attribute	Description
addressId	The address identifier to be removed

API: Read Address

Used to read a single address.

API Basics

Endpoint URL	{base URL}/{addressId}
Method	GET
Successful Response	200 OK
Processing Type	Synchronous
Input	None
Output	Address record
Max Response Limit	N/A

Path Parameter Definitions

Attribute	Description
addressId	The address identifier to be removed

Output Data Definition

Attribute	Data Type	Description
addressId	String	The external identifier of the address.
entityType	Integer	Donates the type of location of the entity (see Additional Data Definition).
entityId	Long	The external identifier of the entity (this will also match internal identifiers).
addressType	Integer	The type of address: (see Additional Data Definition)
primary	Boolean	True if this is the primary address of the entity, false otherwise
addressLine1	String	The first line of the address
addressLine2	String	The second line of the address
addressLine3	String	The third line of the address
city	String	The city of the address
state	String	The state of the address
countryCode	String	The country code of the address (used by supplier)
postalCode	String	The postal code of the address
county	String	The county of the address
companyName	String	A company name associated with that address
contactName	String	Contact name for that address
contactPhone	String	Contact phone number for that address
contactFax	String	Contact fax number for that address
contactEmail	String	Contact email for that address
firstName	String	A first name of a contact at that address
lastName	String	A last name of the contact at that address

phoneticFirstName	String	A phonetic spelling of a first name of a contact at that address
phoneticLastName	String	A phonetic spelling of a last name of a contact at that address
supplierLocation	String	Supplier location information

Additional Data Definitions

Entity Type

Value	Definition
1	Store
2	Supplier
3	Warehouse
4	Finisher

Address Type

Value	Definition
1	Business
2	Postal
3	Return
4	Order
5	Invoice
6	Remittance
7	Billing
8	Delivery
9	External

REST Service Batch

This service allows an external system to schedule an *adhoc* batch job for execution.

Service Base URL

The Cloud service base URL follows the format:

https://<external_load_balancer>/<cust_env>/siocs-int-services/api/batches

API Definitions

API	Description
executeBatch	Schedules the batch for immediate execution.
findBatchJobs	Finds all the batch jobs available to schedule.

API: Execute Batch

Schedules the specified batch job for immediate execution.

If parameter date and/or parameter identifier are entered, they are passed as parameters to the batch job identified by the batch name.

See Batch guide for definition of data set identifiers for various batches.

API Basics

Endpoint URL	{base URL}
Method	POST
Successful Response	202 Accepted
Processing Type	Asynchronous
Input	Batch information
Output	None
Max Response Limit	N/A

Input Data Definition

Attribute	Data Type	Required	Description
batchName	String (256)	Yes	The name of the batch job to execute.
storeId	Long(10)		A store identifier to run the batch for. If included, it will run the batch processing for a single store. If not included, the batch will run for all stores based on functional description of the batch processing.
parameterDate	Date		A parameter date passed to the batch job (see SIOCS adhoc batch documentation).
parameterId	String		A parameter identifier passed to the batch job (see SIOCS adhoc batch documentation).

API: Find Batch Jobs

Finds all the batch jobs available to schedule.

API Basics

Endpoint URL	{base URL}
Method	GET
Successful Response	200 OK

Processing Type	Synchronous
Input	None
Output	List of batch jobs
Max Response Limit	N/A

Output Data Definition

Attribute	Data Type	Description
jobName	String	The job name used to execute the batch.
shortDescription	String	A short description of the batch job.
longDescription	String	A long description of the batch job.
jobParamHint	String	Some hint text for what parameter values might be.
jobInterval	Integer	The execution interval of the batch job (see Additional Data Definition).
batchType	Integer	The type of batch job (see Additional Data Definition).
storeRelated	Boolean	Y indicates the batch job requires store level processing.
enabled	Boolean	Y indicates the batch job is currently enabled and scheduled. This will not prevent batch execution via this service.
lastExecutionTime	String	The timestamp of the last execution of the batch job.
updateDate	String	The last time this record was updated by SIOCS.

Additional Data Definitions

Batch Type

Value	Definition
1	Cleanup
2	Operation
3	System
4	System Cleanup
5	Data Seed
6	Archive

Batch Interval Type

Value	Definition
1	30 Minutes
2	1 Hour
3	2 Hours
4	3 Hours

5	4 Hours
6	6 Hours
7	8 Hours
8	12 Hours
9	24 Hours
10	Monthly

REST Service: Differentiator

This service integrates differentiator foundation data. Asynchronous differentiator integration is processed through staged messages and is controlled by the MPS Work Type: DcsDiff.

This service replaces the RIB flow for differentiators.

Service Base URL

The Cloud service base URL follows the format:

https://<external_load_balancer>/<cust_env>/siocs-int-services/api/differentiators

API Definitions

API	Description
importDifferentiatorTypes	Create or update differentiator types.
importDifferentiators	Create or update differentiators.
deleteDifferentiatorType	Delete a differentiator type and all associated differentiators.
deleteDifferentiator	Delete a differentiator.

API: Import Differentiator Types

Imports a differentiator type by writing a staged message and processing through DCS consumer.

If the number of records exceed 1000, an input too large error is returned.

A "Forbidden" response will occur if application is integrated with MFCS or RMS.

API Basics

Endpoint URL	{base URL}/types/import
Method	POST
Successful Response	202 Accepted
Processing Type	Asynchronous
Input	Differentiator Type information
Output	None
Max Response Limit	N/A

Input Data Definition

Attribute	Data Type	Required	Description
differenatiorTypes	List of details	Yes	The differentiator types to import.

Detail Data Definition

differenatiorTypeId	String (10)	Yes	The differentiator type identifier.
description	String (255)	Yes	The differentiator type description (not translated).

API: Import Differentiators

Imports a differentiator.

If the number of records exceed 1000, an input too large error is returned.

A "Forbidden" response will occur if application is integrated with MFCS or RMS.

API Basics

Endpoint URL	{base URL}/import
Method	POST
Successful Response	202 Accepted
Processing Type	Asynchronous
Input	Differentiator information
Output	None
Max Response Limit	N/A

Input Data Definition

Attribute	Data Type	Required	Description
differenatiors	List of details	Yes	The differentiators to import.

Detail Data Definition

Attribute	Data Type	Required	Description
differenatiorId	String (10)	X	The differentiator type identifier.
differenatiorTypeId	String (10)	X	The differentiator type identifier.
description	String (255)	X	The differentiator type description (not translated).

API: Delete Differentiator Type

Deletes a differentiator type and all associated differentiators.

A "Forbidden" response will occur if application is integrated with MFCS or RMS.

API Basics

Endpoint URL	{base URL}/types/{differenatiorTypeId}/delete
Method	POST
Successful Response	202 Accepted
Processing Type	Asynchronous
Input	None
Output	None
Max Response Limit	N/A

Path Parameter Definitions

Attribute	Description
differentiatorId	The differentiator Id to be removed

REST Service: Finisher

This service integrates finisher and finisher item foundation data. Asynchronous finisher integration is processed through staged messages and is controlled by the MPS Work Type: DcsPartner. Asynchronous finisher item integration is processed through staged messages and is controlled by the MPS Work Type: DcsItemLocation.

Service Base URL

The Cloud service base URL follows the format:

https://<external_load_balancer>/<cust_env>/siocs-int-services/api/finishers

API Definitions

API	Description
importFinishers	Imports a collection of finishers.
deleteFinisher	Deletes a finisher.
importItems	Imports a collection of finisher items.
removeItems	Marks finisher items for deletion.

API: Import Finishers

Imports finishers. This allows 500 finishers per service call.

A "Forbidden" response will occur if application is integrated with MFCS or RMS.

API Basics

Endpoint URL	{base URL}/import
Method	POST
Successful Response	202 Accepted
Processing Type	Asynchronous
Input	List of finisher import
Output	None
Max Response Limit	N/A

Input Data Definition

Attribute	Data Type	Required	Description
finishers	List of details	Yes	A list of finishers to import

Detail Data Definition

Attribute	Data Type	Required	Description
finisherId	Long (10)	Yes	The finisher identifier.
name	String (240)		The finisher name.
status	Integer		Finisher Import Status (see Additional Data Definition).
currencyCode	String (3)		ISO currency code used by the finisher
countryCode	String (3)		The ISO country code assigned to the finisher.
languageCode	String (6)		The ISO language code of the finisher
contactName	String (120)		Name of the finisher's representative contact.
contactPhone	String (20)		Phone number of the finisher's representative contact.
contactFax	String (20)		Fax number of the finisher's representative contact.
contactTelex	String (20)		Telex number of the finisher's representative contact.
contactEmail	String (100)		Email address of the finisher's representative contact.
manufacturerId	String (18)		Manufacturer's identification number
taxId	String (18)		Tax identifier number of the finisher.

transferEntityId	String (20)	Identifier of the transfer entity that the finisher belongs to.
paymentTerms	String (20)	Payment terms for the partner
importCountryCode	String (3)	The ISO country code of the import authority.
importPrimary	Boolean	True Indicates the code is the primary import authority of the import country.

API: Delete Finisher

Deletes a finisher.

A "Forbidden" response will occur if application is integrated with MFCS or RMS.

API Basics

Endpoint URL	{base URL}/{finisherId}/delete
Method	POST
Successful Response	202 Accepted
Processing Type	Asynchronous
Input	None
Output	None
Max Response Limit	N/A

Path Parameter Definitions

Attribute	Description
finisherId	The finisher identifier to be removed.

API: Import Items

Imports finisher items. This allows 5000 items per service call.

A "Forbidden" response will occur if application is integrated with MFCS or RMS.

API Basics

Endpoint URL	{base URL}/{finisherId}/items/import
Method	POST
Successful Response	202 Accepted
Processing Type	Asynchronous
Input	Finisher Item list

Output	None
Max Response Limit	N/A

Input Data Definition

Attribute	Data Type	Required	Description
items	List of details	Yes	A list of items to import.

Detail Data Definition

Attribute	Data Type	Required	Description
itemId	String (25)	Yes	The item identifier.
status	Integer	Yes	Finisher Item Import Status (see Additional Data Definition).

API: Remove Items

Marks finisher items for later deletion. This allows 5000 items per service call.

A "Forbidden" response will occur if application is integrated with MFCS or RMS.

API Basics

Endpoint URL	{base URL}/{finisherId}/items/remove
Method	POST
Successful Response	202 Accepted
Processing Type	Asynchronous
Input	Items list
Output	None
Max Response Limit	N/A

Path Parameter Definitions

Attribute	Description
finisherId	The finisher identifier to be removed.

Input Data Definition

Attribute	Data Type	Required	Description
itemIds	List<String>	Yes	A list of items to remove.

Additional Data Definitions

Finisher Import Status

Value	Definition

1	Active
2	Inactive

Finisher Item Import Status

Value	Definition
1	Active
2	Discontinued
3	Inactive

Rest Service Inventory Adjustment

This service defines operations to manage Inventory Adjustment information.

Service Base URL

The Cloud service base URL follows the format:

https://external_load_balancer/cust_env/siocs-int-services/api/invadjustments

APIs

API	Description
Find Adjustments	Search for transactional store inventory adjustments summary headers based on input criteria.
Read Adjustment	Reads a transaction store inventory adjustment.
Confirm Adjustment	This API is used to confirm an inventory adjustment, finalizing it and processing the inventory movement.
Cancel Adjustment	This API is used to cancel an inventory adjustment.
Create Adjustment	This API is used to create a new manual inventory adjustment whose status is In Progress.

API: Find Adjustments

API is used to search for transaction headers for inventory adjustments. No items or detailed information is returned via this service.

API Basics

Endpoint URL	{base URL}
Method	GET
Successful Response	200 OK
Processing Type	Synchronous
Input	Query parameters

Output	Collection of adjustments
Max Response Limit	10,000

Query Params

Attribute	Type	Req	Definition
referenceId	Long(12)	No	Return adjustments containing this reference identifier.
status	Integer(2)	No	Returns adjustments containing this status: See Index.
adjustmentDateFrom	Date	No	Returns adjustments where the adjustment date is on or after this date.
adjustmentDateTo	Date	No	Returns adjustments where the adjustment date is on or before this date.
updateDateFrom	Date	No	Returns adjustments where the adjustment date is on or after this date.
updateDateTo	Date	No	Returns adjustments where the adjustment date is on or before this date.
itemId	String(25)	No	Returns adjustments containing this item.
reasonId	Long(12	No	Returns adjustments containing this reason code.

Output Data Definition

Attribute	Type	Definition
adjustmentId	Long(12)	Adjustment identifier
storeId	Long(10)	Store Identifier
referenceId	Long(12)	Identifier of original adjustment it was copied from
externalId	String(128)	A unique identifier of the adjustment in an external system
adjustmentDate	Date	The timestamp the inventory is official considered to have been adjusted
status	Integer(2)	The adjustment status: See Index.
createDate	Date	The date the adjustment was created in EICS
updateDate	Date	The date the adjustment was last updated by EICS
approveDate	Date	The date the adjustment was approved in EICS

API: Read Adjustment

This API is used to read an inventory adjustment.

API Basics

Endpoint URL	{base URL}/{adjustmentId}
Method	GET
Successful Response	200 OK
Processing Type	Synchronous
Input	N/A
Output	Inventory Adjustment
Max Response Limit	N/A

Path Parameter Definition

Payload	Type	Definition
adjustmentId	Long(12)	The unique inventory adjustment identifier

Output Data Definition

Attribute	Type	Definition
adjustmentId	Long(12)	The unique inventory adjustment identifier
storeId	Long(10	The unique identifier of the store
referenceId	Long(12)	Identifier of original adjustment it was copied from
externalId	String(128)	Identifier to the adjustment in an external system
adjustmentDate	Date	The timestamp the inventory is official considered to have been adjusted
status	Integer(2)	The adjustment status: See Index
comments	String(2000)	Comments associated to the adjustment
createDate	Date	The date the adjustment was created by EICS
createUser	String (128)	The user that created the adjustment in EICS
updateDate	Date	The date the adjustment was last updated by EICS
updateUser	String (128)	The user that last updated the adjustment in EICS
approveDate	Date	The date the adjustment was approved in EICS
approveUser	String (128)	The user that approved the adjustment in EICS
externalCreateUser	String (128)	The non-EICS user that created the inventory adjustment in the external system
externalUpdateUser	String (128)	The non-EICS user that updated the inventory adjustment in the external system
lineItems	Collection	A collection of Inventory Adjustment Line Items

Inventory Adjustment Line Item

Payload	Type	Definition

lineId	Long(12)	The unique SIOCS identifier of the line item
itemId	Long(25)	The sku level item identifier
reasonId	Long(12)	The unique internal identifier of the reason code
caseSize	BigDecimal(10,2)	Case size associated to the line item
quantity	BigDecimal(20,4)	Quantity associated to the line item
uins	Collection{String}	The UINs associated to the item quantities

API: Confirm Adjustment

This API is used to confirm an inventory adjustment, finalizing it and processing the inventory movement.

API Basics

Endpoint URL	{base URL}/{adjustmentId}/confirm
Method	POST
Successful Response	204 No Content
Processing Type	Synchronous
Input	N/A
Output	N/A
Max Response Limit	N/A

Path Parameter Definition

Payload	Type	Definition
adjustmentId	Long(12)	The unique inventory adjustment identifier

API: Cancel Adjustment

This API is used to cancel an inventory adjustment.

API Basics

Endpoint URL	{base URL}/{adjustmentId}/cancel
Method	POST
Successful Response	204 No Content
Processing Type	Synchronous
Input	N/A
Output	N/A
Max Response Limit	N/A

Path Parameter Definition

Payload	Type	Definition

adjustmentId	Long(12)	The unique inventory adjustment identifier

API: Create Adjustment

This API is used to create a new manual inventory adjustment whose status is In Progress.

API Basics

Endpoint URL	{base URL}
Method	POST
Successful Response	200 OK
Processing Type	Synchronous
Input	Inventory Adjustment
Output	Inventory Adjustment Reference
Max Response Limit	5,000 Items

Input Data Definition

Attribute	Type	Req	Definition
storeId	Long(10)	X	Store Identifier
referenceId	Long(12)		Identifier of original adjustment it was copied from
externalId	String(128)		A unique identifier to the adjustment in an external system
adjustmentDate	Date		The timestamp the inventory is official considered to have been adjusted
externalUser	String(128)		A non-EICS user that created the inventory adjustment in the external system
comments	String(2000)		Comments associated to the inventory adjustment
lineItems	Collection	X	collection of inventory adjustment line items

Inventory Adjustment Line Item

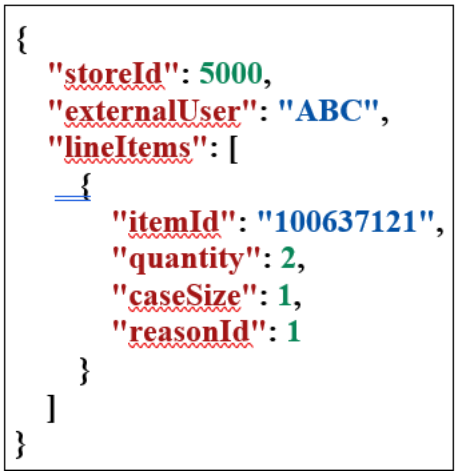
Payload	Type	Req	Definition
itemId	String(25)	X	The sku level item identifier. It must be an inventoriable item.
reasonId	Long(12)	X	The unique internal identifier of the reason code.
quantity	BigDecimal(20,4)	X	Quantity associated to the line item
caseSize	BigDecimal(10,,2)		Case size associated to the line item
uins	Collection{String}		The UINs associated to the item quantities

Output Data Definition

Attribute	Type	Definition

adjustmentId	Long(12)	he newly created adjustment identifier

Example Input



Business Data Errors

Error	Definition
CASE_SIZE_NOT_WHOLE	The adjustment requires a non-decimal case size.
EXISTING_UIN_CANNOT_BE_ADDED	An already existing UIN cannot be added to the store.
NON_INVENTORIABLE_ITEM	The item cannot be adjusted because it cannot have inventory.
NON_MANAGED_STORE	The store does not manage its inventory.
QUANTITY_NOT_WHOLE	The adjustment requires a non-decimal quantity.

Additional Date Definition

Inventory Adjustment Status

ID	Status
1	In Progress
2	Completed
3	Canceled

REST Service Item

This service integrates the item foundation data with an external application. Asynchronous item integration is processed through staged messages and is controlled by the MPS Work Types.

Note that this is item level foundational data. To lookup or access item information, use the item inquiry REST service.

Service Base URL

The Cloud service base URL follows the format:

https://<external_load_balancer>/<cust_env>/siocs-int-services/api/items

API Definitions

API	Description
importItems	Import items into the system.
removeItems	Mark items for deletion at some point in the future when no records are using them.
importHierarchies	Imports item hierarchies into the system.
removeHierarchies	Marks hierarchies for deletion at some point in the future when no records are using them.
importRelatedItems	Imports the associations of related items.
deleteRelatedItems	Deletes an association of related items.
importImageUrls	Imports image URLs associated to the item.
deleteImageUrls	Delete image URLs associated to the item.

API: Import Items

Imports items.

This flow is managed in MPS system with the following family: DcsItem.

If the input exceeds more than 100 records, an input too large error will be returned.

A "Forbidden" response will occur if application is integrated with MFCS or RMS.

API Basics

Endpoint URL	{base URL}/import
Method	POST
Successful Response	202 Accepted
Processing Type	Asynchronous
Input	List of Items to import
Output	None
Max Response Limit	N/A

Input Data Definition

Attribute	Data Type	Required	Description
finishers	List of details	Yes	A list of items to import

Detail Data Definition

Attribute	Data Type	Required	Description
itemId	String (25)	Yes	The unique item identifier (sku number).
transactionLevel	Long	Yes	Number indicating which of the three levels transactions occur for the item. Items may only be used on transactions with inventory tracked if the transaction level and item level match.
itemLevel	Long	Yes	Number indicating which of the three levels an item resides at. Items may only be used on transactions with inventory tracked if the transaction level and item level match.
departmentId	Long (12)	Yes	The merchandise hierarchy department identifier.
classId	Long (12)		The merchandise hierarchy class identifier.
subclassId	Long (12)		The merchandise hierarchy subclass identifier.
shortDescription	String (255)		A short description of the item.
longDescription	String (400)		A long description of the item.
differentiator1	String (10)		The first differentiator identifier.
differentiator2	String (10)		The second differentiator identifier.
differentiator3	String (10)		The third differentiator identifier.
differentiator4	String (10)		The fourth differentiator identifier.
status	Integer		Item Import Status (see Additional Data Definition).
parentId	String (25)		The unique identifier of the item at the next level above this item.
pack	Boolean		True if the item is pack, false otherwise.
simplePack	Boolean		True if the item is a simple pack, false otherwise.
sellable	Boolean		True if the item is sellable, false otherwise.
orderable	Boolean		True if the item can be ordered from a supplier, false otherwise.
shipAlone	Boolean		True if the item must be shipped in separated packaging, false otherwise.
inventoriable	Boolean		True if the item is inventoried, false otherwise.

notionalPack	Boolean	True indicates the inventory is held at the component level. All notional pack are marked as inventoriable in SIOCS.
estimatePackInventory	Boolean	True if the item allows estimating pack inventory from component positions, false otherwise.
primaryReferenceItem	Boolean	True indicates it the primary sub-translation level item.
orderAsType	Boolean	True indicates a buyer pack is receivable at the pack level. N means at the component level.
standardUom	String (4)	The unit of measure that inventory is tracked in.
packageUom	String (4)	The unit of measure associated with a package size.
packageSize	Double	The size of the product printed (will be printed on the label).
eachToUomFactor	BigDecimal	The multiplication factor to convert 1 EA to the equivalent standard unit of measure.
barcodeFormat	String (4)	The format of a barcode (used for Type 2 barcode items).
barcodePrefix	Long (9)	The barcode prefix used in association with the Type 2 barcode of this item.
wastageType	Integer	Waste Type (see Additional Data Definition).
wastagePercent	BigDecimal	Wastage percent.
wastagePercentDefault	BigDecimal	Default wastage percent.
suggestedRetailCurrency	String (3)	The currency of the manufacturer suggested retail price.
suggestedRetailPrice	BigDecimal	Manufacturer suggested retail price.
brand	String (30)	Brand name of the brand the item belongs to.
brandDescription	String (120)	Brand description of the brand the item belongs to.
components	List of components	A list of components for the item (if the item is a pack)

Component Data Definition

Attribute	Data Type	Required	Description
componentItemId	String (25)	Yes	The item identifier of the component item within the pack.
quantity	BigDecimal	Yes	The quantity of component item within the pack

API: Remove Items

Deactivate items.

This flow is managed in MPS system with the following family: DcsItem.

If the input exceeds more than 500 records, an input too large error will be returned.

A "Forbidden" response will occur if application is integrated with MFCS or RMS.

API Basics

Endpoint URL	{base URL}/remove
Method	POST
Successful Response	202 Accepted
Processing Type	Asynchronous
Input	Items list to remove
Output	None
Max Response Limit	N/A

Input Data Definition

Attribute	Data Type	Required	Description
items	List of details	Yes	A list of item reference to update to a non-active status.

Detail Data Definition

Attribute	Data Type	Required	Description
itemId	String (25)	Yes	The unique item identifier
status	Integer	Yes	Item Remove Status (see Additional Data Definition).

API: Import Hierarchies

Imports item hierarchies.

This flow is managed in MPS system with the following family: DcsHierarchy.

If the input exceeds more than 500 records, an input too large error will be returned.

A "Forbidden" response will occur if application is integrated with MFCS or RMS.

API Basics

Endpoint URL	{base URL}/hierarchies/import
Method	POST
Successful Response	202 Accepted
Processing Type	Asynchronous

Input	Items hierarchy list
Output	None
Max Response Limit	N/A

Input Data Definition

Attribute	Data Type	Required	Description
hierarchies	List of details	Yes	The hierarchies to import.

Detail Data Definition

Attribute	Data Type	Required	Description
departmentId	Long (12)	Yes	The hierarchy department identifier.
departmentName	String (360)		The hierarchy department name.
classId	Long (12)		The hierarchy class identifier.
className	String (360)		The hierarchy class name.
subclassId	Long (12)		The hierarchy subclass identifier.
subclassName	String (360)		The hierarchy subclass name.

API: Remove Hierarchies

Deactivates item hierarchies. Once no information is associated to the item hierarchies, a cleanup batch will remove them from the database.

This flow is managed in MPS system with the following family: DcsHierarchy.

If the input exceeds more than 500 records, an input too large error will be returned.

A "Forbidden" response will occur if application is integrated with MFCS or RMS.

API Basics

Endpoint URL	{base URL}/hierarchies/remove
Method	POST
Successful Response	202 Accepted
Processing Type	Asynchronous
Input	Items hierarchy list
Output	None
Max Response Limit	N/A

Input Data Definition

Attribute	Data Type	Required	Description
hierarchies	List of details	Yes	The images to remove.

Detail Data Definition

Attribute	Data Type	Required	Description
departmentId	Long (12)	Yes	The hierarchy department identifier.
classId	Long (12)		The hierarchy class identifier.
subclassId	Long (12)		The hierarchy subclass identifier

API: Import Related Items

Imports item relationships that an item may belong to.

This flow is managed in MPS system with the following family: DcsItem.

If the input exceeds more than 500 records, an input too large error will be returned.

A "Forbidden" response will occur if application is integrated with MFCS or RMS.

API Basics

Endpoint URL	{base URL}/related/import
Method	POST
Successful Response	202 Accepted
Processing Type	Asynchronous
Input	Items Relationship list
Output	None
Max Response Limit	N/A

Input Data Definition

Attribute	Data Type	Required	Description
relationships	List of details	Yes	The relationships to import.

Detail Data Definition

Attribute	Data Type	Required	Description
relationshipId	Long (20)	Yes	The identifier of the relationship.
relationshipType	Integer	Yes	Relationship Type (see Additional Data Definitions).

name	String (120)		The name of the relationship.
itemId	String (25)	Yes	The item whose related records are being recorded.
mandatory	Boolean	Yes	True if the relationships are mandatory.
relatedItems	List of related items	Yes	The related items.

Related Item Data Definition

Attribute	Data Type	Required	Description
itemId	String (25)	Yes	The item that is related.
effectiveDate	Date		Date at which this relationship becomes active.
endDate	Date		Last date at which this relationship is active.
priorityNumber	Long (4)		Number defining priority in the case of multiple substitute items.

API: Delete Related Items

Deletes relationships between items.

This flow is managed in MPS system with the following family: DcsItem.

If the input exceeds more than 500 records, an input too large error will be returned.

A "Forbidden" response will occur if application is integrated with MFCS or RMS.

API Basics

Endpoint URL	{base URL}/related/delete
Method	POST
Successful Response	202 Accepted
Processing Type	Asynchronous
Input	None
Output	None
Max Response Limit	N/A

Input Data Definition

Attribute	Data Type	Required	Description
relationshipIds	List<Long>	Yes	The relationship identifiers to remove.

API: Import Image Urls

Import image URLs associated to the item.

This flow is managed in MPS system with the following family: DcsItem.

If the input exceeds more than 500 records, an input too large error will be returned.

A "Forbidden" response will occur if application is integrated with MFCS or RMS.

API Basics

Endpoint URL	{base URL}/images/import
Method	POST
Successful Response	202 Accepted
Processing Type	Asynchronous
Input	Items Image list
Output	None
Max Response Limit	N/A

Input Data Definition

Attribute	Data Type	Required	Description
images	List of details	Yes	The images to import.

Detail Data Definition

Attribute	Data Type	Required	Description
itemId	String (25)	Yes	The item identifier/sku number.
imageType	Integer	Yes	Image Type (see Additional Data Definitions).
storeId	Long (10)		The store identifier. This is required only if the image type is QR_CODE.
imageName	String (120)	Yes	The name of the image.
imageSize	String (6)		The size of the image: (T) thumbnail. Other than (T), any text is accepted and there is no definition to validate against, but the text has no meaning.
url	String (1000)	Yes	The universal resource locator of the image.
displaySequence	Integer (2)		The sequence the item should be displayed in.
startDate	Date		The date the image becomes active.

endDate	Date	The date the image ceases being active.

API: Delete Image Urls

Deletes image URLs.

This flow is managed in MPS system with the following family: DcsItem

If the input exceeds more than 500 records, an input too large error will be returned.

A "Forbidden" response will occur if application is integrated with MFCS or RMS.

API Basics

Endpoint URL	{base URL}/images/delete
Method	POST
Successful Response	202 Accepted
Processing Type	Asynchronous
Input	Items Image list
Output	None
Max Response Limit	N/A

Input Data Definition

Attribute	Data Type	Required	Description
images	List of details	Yes	The images to remove.

Detail Data Definition

Attribute	Data Type	Required	Description
itemId	String (25)	Yes	The item identifier/sku number.
imageName	String (120)	Yes	The name of the image.
imageType	Integer	Yes	Image Type (see Additional Data Definitions).

Additional Data Definitions

Item Status

Value	Definition
1	Active
2	Discontinued
3	Inactive
4	Deleted
5	Auto Stockable

6	Non Ranged

Item Import Status

Value	Definition
1	Active
5	Auto Stockable
6	Non Ranged

Item Remove Status

Value	Definition
2	Discontinued
3	Inactive
4	Deleted

Wastage Type

Value	Definition
1	Sales Wastage
2	Spoilage Wastage

Relationship Type

Value	Definition
1	Related
2	Substitute
3	Up-Sell
4	Cross-Sell

Image Type

Value	Definition
1	Image
2	QRCode

REST Service: Item Inquiry

This service allows the customer to retrieve information about items. These services are intended to find item themselves and do not retrieve inventory. For inventory queries, see inventory inquiry services.

Service Base URL

The Cloud service base URL follows the format:

https://<external_load_balancer>/<cust_env>/siocs-int-services/api/iteminquiries

API Definitions

API	Description
findItemBySearchScan	Searches for summary item information based on basic item scan information such as UPC, barcode, and so on.
findItemBySource	Searches for summary item information based on hierarchy or source location search criteria.
findItems	Searches item information based on multiple items (and an optional store).

API: Find Item by Search Scan

Search for item information by on unknown identifier, most likely a scan. It first searches for the item using the scan itself as the potential item, and if found will return record(s). It then searches as a UPC, barcode, or UIN in that order halting if it finds potential matches.

API Basics

Endpoint URL	{base URL}/scan/{searchScan}
Method	GET
Successful Response	200 OK
Processing Type	Synchronous
Input	None
Output	List of items scanned
Max Response Limit	N/A

Path Parameter Definitions

Attribute	Description
searchScan	An Item, UPC, Barcode or UIN take from a scan.

Query Parameter Definition

Attribute	Data Type	Required	Description
storeId	Long		A store to verify if found item is ranged to.

Output Data Definition

Attribute	Data Type	Description
itemId	String	The item identifier.
type	Integer	Item Type (see Additional Data Definition).
status	String	Item Status (see Additional Data Definition).

shortDescription	String	A short description of the item.
longDescription	String	A long description of the item.
departmentId	Long	The identifier of the department the item belongs to.
classId	Long	The identifier of the class the item belongs to.
subclassId	Long	The identifier of the subclass the item belongs to.
storeId	Long	The store identifier passed in as the query parameter.
ranged	Boolean	True if the item is ranged to the requested store, false otherwise.

API: Find Item by Source

Searches for summary information about an item based on search criteria, primarily hierarchy and source location.

If the number of items found exceeds 10000, a maximum limit error will be returned. Additional or more limiting search criteria will be required.

At least one query parameter is required. If source type location is entered, then a source id for that type must also be entered.

API Basics

Endpoint URL	{base URL}/source
Method	GET
Successful Response	200 OK
Processing Type	Synchronous
Input	None
Output	List of Items summary
Max Response Limit	N/A

Query Parameter Definition

Attribute	Data Type	Description
description	String	Include only items with description text matching this description.
sourceType	Integer	Include only items available from this source type (see Additional Data Definition).
sourceId	String	Include only items available from this source identifier.
departmentId	Long	Include only items associated to this merchandise hierarchy department.

classId	Long	Include only items associated to this merchandise hierarchy class.
subclassId	Long	Include only items associated to this merchandise hierarchy subclass.
storeId	Long	Include only items ranged to this particular store.

Output Data Definition

Attribute	Data Type	Description
itemId	String	The item identifier.
type	Integer	Item Type (see Additional Data Definition).
status	String	Item Status (see Additional Data Definition).
shortDescription	String	A short description of the item.
longDescription	String	A long description of the item.
departmentId	Long	The identifier of the department the item belongs to.
classId	Long	The identifier of the class the item belongs to.
subclassId	Long	The identifier of the subclass the item belongs to.

API: Find Items

Searches for detailed information about the items using the specified input.

If the number of items found exceeds 10000, a maximum limit error will be returned. Additional or more limiting input criteria will be required.

API Basics

Endpoint URL	{base URL}
Method	POST
Successful Response	200 OK
Processing Type	Synchronous
Input	Criteria
Output	List of Items
Max Response Limit	N/A

Input Data Definition

Attribute	Data Type	Required	Description
storeId	Long	Yes	The store to retrieve item information for

itemIds	List<String>	Yes	A list of items to retrieve item information for

Output Data Definition

Attribute	Data Type	Description
itemId	String	The unique item identifier (sku number).
transactionLevel	Long	Number indicating which of the three levels transactions occur for the item. Items may only be used on transactions with inventory tracked if the transaction level and item level match.
itemLevel	Long	Number indicating which of the three levels an item resides at. Items may only be used on transactions with inventory tracked if the transaction level and item level match.
departmentId	Long	The merchandise hierarchy department identifier.
classId	Long	The merchandise hierarchy class identifier.
subclassId	Long	The merchandise hierarchy subclass identifier.
shortDescription	String	A short description of the item.
longDescription	String	A long description of the item.
differentiator1	String	The first differentiator identifier.
differentiator2	String	The second differentiator identifier.
differentiator3	String	The third differentiator identifier.
differentiator4	String	The fourth differentiator identifier.
status	Integer	The status (see Additional Data Definition).
parentId	String	The unique identifier of the item at the next level above this item.
pack	Boolean	True if the item is pack, false otherwise.
simplePack	Boolean	True if the item is a simple pack, false otherwise.
sellable	Boolean	True if the item is sellable, false otherwise.
orderable	Boolean	True if the item can be ordered from a supplier, false otherwise.
shipAlone	Boolean	True if the item must be shipped in separated packaging, false otherwise.
inventoriable	Boolean	True if the item is inventoried, false otherwise.

notionalPack	Boolean	True indicates the inventory for the pack is tracked at the component level.
estimatePackInventory	Boolean	True if the item allows estimating pack inventory from component positions, false otherwise.
primaryReferenceItem	Boolean	True indicates it the primary sub-translation level item.
orderAsType	Boolean	True indicates a buyer pack is receivable at the pack level. N means at the component level.
standardUom	String	The unit of measure that inventory is tracked in.
packageUom	String	The unit of measure associated with a package size.
packageSize	Double	The size of the product printed (will be printed on the label).
eachToUomFactor	BigDecimal	The multiplication factor to convert 1 EA to the equivalent standard unit of measure.
barcodeFormat	String	The format of a barcode (used for Type 2 barcode items).
barcodePrefix	Long	The barcode prefix used in association with the Type 2 barcode of this item.
wastageType	Integer	Type of wastage (see Additional Data Definition).
wastagePercent	BigDecimal	Wastage percent.
wastagePercentDefault	BigDecimal	Default wastage percent.
suggestedRetailCurrency	String	The currency of the manufacturer suggested retail price.
suggestedRetailPrice	BigDecimal	Manufacturer suggested retail price.
brand	String	Brand name of the brand the item belongs to.
brandDescription	String	Brand description of the brand the item belongs to.
createDate	Date	The date the item was created in EICS.
updateDate	Date	The last date the item was updated in EICS.

(RANGED INFO)

storeId	Long	The store identifier.
status	String	The item status. (see Additional Data Definition).
primarySupplierId	Long	The unique identifier of the primary supplier of the item to this store location.
storeControlPricing	Boolean	True indicates the item price can be controlled by the store.

rfid	Boolean	True indicates the item is RFID tagged.
defaultCurrencyCode	String	The default currency of the item's price at this store.
purchaseType	Long	Purchase Type (see Additional Data Definition).
uinType	Integer	UIN Type (see Additional Data Definition).
uinCaptureTime	Integer	UIN Capture Time (see Additional Data Definition).
uinLabelId	Long	The UIN label unique identifier.
uinExternalCreateAllowed	Boolean	True if an external system can create a UIN, false otherwise.
replenishmentMethod	String	The replenishment method: (SO) Store Orders (has meaning), otherwise meaningless text.
rejectStoreOrder	Boolean	True indicates store orders must be on or after the next delivery date or should be rejected.
multipleDeliveryPerDayAllowed	Boolean	True indicates the item allows multiple deliveries per day at the location.
nextDeliveryDate	Date	The next delivery date of the time based on its replenishment type.

Additional Data Definitions

Item Status Type

Value	Definition
1	Active
2	Discontinued
3	Inactive
4	Deleted
5	Auto Stockable
6	Non Ranged

Item Type

Value	Definition
1	Item
2	Simple Pack
3	Complex Pack
4	Simple Breakable Pack
5	Complex Breakable Pack

Source Type

Value	Definition
1	Supplier
2	Warehouse
3	Finisher

Item Purchase Type

Value	Definition
1	Consignment
2	Concession

Item UIN Type

Value	Definition
1	Serial
2	AGSN

Item UIN Capture Time Type

Value	Definition
1	Sale
2	Store Receiving

Item Wastage Type

Value	Definition
1	Sales Wastage
2	Spoilage

REST Service: Item Inventory

This service retrieves information about item inventory.

Service Base URL

The Cloud service base URL follows the format:

https://<external_load_balancer>/<cust_env>/siocs-int-services/api/inventory

API Definitions

API	Description
Find Available Inventory	Search for available inventory information by multiple items and multiple locations.
Find Inventory	Searches for standard inventory information by multiple items and multiple locations.

API	Description
Find Expanded Inventory	Searches for expanded inventory information by multiple items at a single store.
Find Future Inventory	Searches for future inventory delivery information by a single item and a single store.
Find Inventory In Buddy Stores	Searches for inventory information at buddy stores by single input store and multiple items.
Find Inventory In Transfer Stores	Searches for inventory information at transfer zone stores by single input store and multiple items.

API: Find Available Inventory

Searches for available inventory quantity about an item in requested locations. Only transaction-level items are processed, and only current available inventory is returned. The multiplied combination of items and locations within the input criteria cannot exceed 10,000. Invalid items or locations will not cause this API to fail. Inventory is returned for any item and locations found and is not returned invalid or not found items or locations.

API Basics

Endpoint URL	{base URL}/available
Method	POST
Successful Response	200 OK
Processing Type	Synchronous
Input	Criteria
Output	List of items
Max Response Limit	10,000

Input Data Definition

Attribute	Data Type	Required	Description
itemIds	List of Strings	Yes	A list of items to retrieve available inventory for.
locationIds	List of Longs	Yes	A list of location identifiers to retrieve available inventory for.
locationType	Integer	Yes	A location type: See Location Type

Example Input

{

"itemIds": [

"100637156",

"100637172",

"100653105"

],

"locationIds": [

5000,

5001,

5005

],

"locationType": 1

}

Output Data Definition

Attribute	Data Type	Required	Description
itemId	String	Yes	The item identifier.
locationId	Long	Yes	The location identifier.
locationType	Integer	Yes	The location type: See Location Type.
availableQuantity	BigDecimal	Yes	The amount of available inventory.
unitOfMeasure	String	Yes	The unit of measure of the available inventory.
estimatedPack	Boolean	Yes	True if this is an estimated pack quantity, false otherwise.

Example Output

[

{

"itemId": "100637113",

"locationId": 5000,

"locationType": 1,

"availableQuantity": 200.0000,

"unitOfMeasure": "EA",

"estimatedPack": false

},

{

"itemId": "100637113",

"locationId": 5001,

"locationType": 1,

"availableQuantity": 200.0000,

"unitOfMeasure": "EA",

"estimatedPack": false

},

}

Additional Data Definitions

Location Type

Value	Definition
1	Store
2	Warehouse

API: Find Inventory

Query lookup of detailed inventory information about a multiple item in multiple stores. The multiplied combination of items and locations within the input criteria cannot exceed 10,000. Invalid items or locations will not cause this API to fail. Inventory is returned for any item and locations found and is not returned invalid or not found items or locations.

API Basics

Endpoint URL	{base URL}/positions
Method	POST
Successful Response	200 OK
Processing Type	Synchronous
Input	Criteria
Output	List of inventory of item at stores
Max Response Limit	10,000

Input Data Definition

Attribute	Data Type	Required	Description
itemIds	List of Strings	Yes	A list of items to retrieve inventory for.
storeIds	List of Longs	Yes	A list of store identifiers to retrieve inventory for.
sellingUnitOfMeasure	Boolean	-	True indicates an attempt to use the selling unit of measure of the item, false indicates to use the standard unit of measure. If conversion cannot take place, it defaults back to standard unit of measure.

Example Input

{

"itemIds": [

"100637156",

"100637172",

"100668091"

],
"storeIds": [
5000,
5001,
5002
],
"sellingUnitOfMeasure": true
}

Output Data Definition

Attribute	Data Type	Required	Description
itemId	String	Yes	The item identifier.
storeId	Long		The store identifier if the item is ranged to a store.
ranged	Boolean	Yes	True if the item is ranged to the store, false otherwise.
estimated	Boolean	Yes	True if the quantities are estimated, false otherwise.
unitOfMeasure	String	Yes	The unit of measure of the quantities.
caseSize	BigDecimal	Yes	The default case size of the item.
quantityStockOnHand	BigDecimal	Yes	The stock on hand quantity.
quantityBackroom	BigDecimal	Yes	The quantity located in the back room area.
quantityShopfloor	BigDecimal	Yes	The quantity located on the shop floor.
quantityDeliveryBay	BigDecimal	Yes	The quantity located in the delivery bay.
quantityAvailable	BigDecimal	Yes	The available to sell quantity.
quantityUnavailable	BigDecimal	Yes	The unavailable to sell quantity.
quantityNonSellable	BigDecimal	Yes	The total non-sellable quantity.
quantityInTransit	BigDecimal	Yes	The quantity currently in transit.
quantityCustomerReserved	BigDecimal	Yes	The quantity reserved for customer orders.
quantityTransferReserved	BigDecimal	Yes	The quantity reserved for transfers.
quantityVendorReturn	BigDecimal	Yes	The quantity reserved for vendor returns.
nonSellableQuantities	List of Non-Sellable Quantities	-	A collection containing the specific quantity in each non-sellable quantity type bucket.

Non-Sellable Quantity Data Definition

Attribute	Data Type	Required	Description
nonsellableTypeId	Long	Yes	The non-sellable type unique identifier.
quantity	quantity	Yes	The quantity in this particular non-sellable type bucket.

Example Output

[
{
"itemId": "100637156",
"storeId": 5000,
"ranged": true,
"estimated": false,
"unitOfMeasure": "EA",
"caseSize": 100.00,
"quantityStockOnHand": 10.0000,
"quantityBackroom": 10.0000,
"quantityShopfloor": 0.0000,
"quantityDeliveryBay": 0.0000,
"quantityAvailable": 10.0000,
"quantityUnavailable": 0.0000,
"quantityNonSellable": 0.0000,
"quantityInTransit": 0.0000,
"quantityCustomerReserved": 0.0000,
"quantityTransferReserved": 0.0000,
"quantityVendorReturn": 0.0000
},
{
"itemId": "100637172",
"storeId": 5000,
"ranged": true,
"estimated": false,
"unitOfMeasure": "EA",
"caseSize": 100.00,
"quantityStockOnHand": 10.0000,
"quantityBackroom": -10.0000,
"quantityShopfloor": 0.0000,
"quantityDeliveryBay": 0.0000,
"quantityAvailable": -10.0000,

"quantityUnavailable": 20.0000,
"quantityNonSellable": 20.0000,
"quantityInTransit": 0.0000,
"quantityCustomerReserved": 0.0000,
"quantityTransferReserved": 0.0000,
"quantityVendorReturn": 0.0000,
"nonSellableIdos": [
{
"nonsellableTypeId": 1,
"quantity": 15.0000
},
{
"nonsellableTypeId": 2,
"quantity": 5.0000
}
]
}
}

API: Find Expanded Inventory

Searches for expanded inventory information about multiple items within a single store.

API Basics

Endpoint URL	{base URL}/{storeId}/expanded
Method	POST
Successful Response	200 OK
Processing Type	Synchronous
Input	Criteria
Output	List of inventory of items
Max Response Limit	2,500

Path Parameter Definitions

Attribute	Description
storeId	The store identifier of the store to process items for.

Input Data Definition

Attribute	Data Type	Required	Description
itemIds	List of Strings	Yes	A list of items to retrieve expanded inventory for.

Example Input

{

"itemIds": [

"100637156",

"100637172",

"100695081"

]

}

Output Data Definition

Attribute	Data Type	Required	Description
itemId	String	Yes	The item identifier.
storeId	Long		The store identifier if the item is ranged to a store.
ranged	Boolean	Yes	True if the item is ranged to the store, false otherwise.
estimated	Boolean	Yes	True if the quantities are estimated, false otherwise.
unitOfMeasure	String	Yes	The unit of measure of the quantities.
caseSize	BigDecimal	Yes	The default case size of the item.
quantityStockOnHand	BigDecimal	Yes	The stock on hand quantity.
quantityBackroom	BigDecimal	Yes	The quantity located in the back room area.
quantityShopfloor	BigDecimal	Yes	The quantity located on the shop floor.
quantityDeliveryBay	BigDecimal	Yes	The quantity located in the delivery bay.
quantityAvailable	BigDecimal	Yes	The available to sell quantity.
quantityUnavailable	BigDecimal	Yes	The unavailable to sell quantity.
quantityNonSellable	BigDecimal	Yes	The total non-sellable quantity.
quantityInTransit	BigDecimal	Yes	The quantity currently in transit.
quantityCustomerReserved	BigDecimal	Yes	The quantity reserved for customer orders.
quantityTransferReserved	BigDecimal	Yes	The quantity reserved for transfers.
quantityVendorReturn	BigDecimal	Yes	The quantity reserved for vendor returns.
firstReceivedDate	Date	-	The first date the item was received into stock.

Attribute	Data Type	Required	Description
lastReceivedDate	Date	-	The date the item last received inventory into stock.
lastReceivedQuantity	BigDecimal	-	Total amount of inventory received on the last date it was received.
openStockCounts	Integer	-	The number of stock counts open for the item at this store.
lastStockCountType	Integer	-	The type of stock count (see Additional Data Definition).
lastStockCountApprovedDate	Date	-	The date this item was last approved on a stock count at this store.
lastStockCountTimeframe	Integer	-	The stock count timeframe (see Additional Data Definition).
uinProblemLine	Boolean	Yes	True indicates it is UIN problem line item, false otherwise.
lastRequestedQuantity	BigDecimal	-	The quantity last requested for this item.
lastUpdateDate	Date	Yes	The timestamp of the last time this record was updated.
nonSellableQuantities	Collection of Non-Sellable Quantities	-	The specific quantities in each non-sellable quantity type bucket.

Non-Sellable Quantity Data Definition

Attribute	Data Type	Required	Description
nonsellableTypeId	Long	Yes	The non-sellable type unique identifier.
quantity	quantity	Yes	The quantity in this particular non-sellable type bucket.

Example Output

[

{

"itemId": "100637113",

"storeId": 5000,

"ranged": true,

"estimated": false,

"unitOfMeasure": "EA",

"caseSize": 100.00,

"quantityStockOnHand": 200.0000,

"quantityBackroom": 200.0000,

"quantityShopfloor": 0.0000,

"quantityDeliveryBay": 0.0000,

"quantityAvailable": 200.0000,

"quantityUnavailable": 0.0000,
"quantityNonSellable": 0.0000,
"quantityInTransit": 0.0000,
"quantityCustomerReserved": 0.0000,
"quantityTransferReserved": 0.0000,
"quantityVendorReturn": 0.0000,
"quantityLastReceived": 0.0000,
"quantityLastRequested": 0.0000,
"openStockCounts": 0,
"lastStockCountTimeframe": 3,
"uinProblemLine": false,
"lastUpdateDate": "2022-07-15T06:23:27-05:00"
},
{
"itemId": "100637121",
"storeId": 5000,
"ranged": true,
"estimated": false,
"unitOfMeasure": "EA",
"caseSize": 100.00,
"quantityStockOnHand": 200.0000,
"quantityBackroom": 180.0000,
"quantityShopfloor": 0.0000,
"quantityDeliveryBay": 0.0000,
"quantityAvailable": 180.0000,
"quantityUnavailable": 20.0000,
"quantityNonSellable": 20.0000,
"quantityInTransit": 0.0000,
"quantityCustomerReserved": 0.0000,
"quantityTransferReserved": 0.0000,
"quantityVendorReturn": 0.0000,
"quantityLastReceived": 0.0000,
"quantityLastRequested": 0.0000,
"openStockCounts": 0,

"lastStockCountTimeframe": 3,
"uinProblemLine": false,
"lastUpdateDate": "2022-07-15T06:23:27-05:00",
"nonSellableIdos": [
{
"nonsellableTypeId": 1,
"quantity": 15.0000
},
{
"nonsellableTypeId": 2,
"quantity": 5.0000
}
]
}
}

API: Find Future Inventory

Searches for future delivery records for a single store and single item.

API Basics

Endpoint URL	{base URL}/{storeId}/{itemId}/future
Method	GET
Successful Response	200 OK
Processing Type	Synchronous
Input	None
Output	List of delivery records
Max Response Limit	N/A

Path Parameter Definitions

Attribute	Description
storeId	The store identifier to retrieve future inventory for.
itemId	The item identifier to retrieve future inventory for.

Output Data Definition

Attribute	Data Type	Required	Description
itemId	String (25)	Yes	The item identifier.
storeId	Long	Yes	The store identifier.

Attribute	Data Type	Required	Description
deliveries	List of deliveryIds	-	A list of delivery information if it exists.

Delivery Data Definition

Attribute	Data Type	Required	Description
sourceLocationType	Integer	Yes	Item Location Type (see Additional Data Definition).
sourceLocationId	Long	Yes	The unique identifier of the source location of the delivery.
deliveryType	Integer	Yes	Item Delivery Type (see Additional Data Definition).
expectedDate	Date	Yes	The date the inventory is expected to arrive.
quantityInbound	BigDecimal	Yes	Amount of inventory inbound on the delivery.
quantityOrdered	BigDecimal	Yes	Amount of inventory on order.

Example Output

{

"itemId": "100637121",

"storeId": 5000,

"deliveryIdos": [

{

"sourceLocationType": 1,

"sourceLocationId": 5001,

"deliveryType": 3,

"quantityInbound": 30.0000,

"quantityOrdered": 0.0000

}

]

}

Additional Data Definitions

Item Delivery Type

Value	Definition
1	Allocation
2	Purchase Order
3	Transfer

Value	Definition
-	-

Item Location Type

Value	Definition
1	Store
2	Supplier
3	Warehouse
4	Finisher

API: Find Inventory in Buddy Stores

Searches for inventory information at buddy stores by single input store and multiple items.

API Basics

Endpoint URL	{baseUrl}/{storeId}/associated
Method	POST
Successful Response	200 OK
Processing Type	Synchronous
Input	List of items
Output	List of inventory records
Max Response Limit	N/A

Path Parameter Definitions

Attribute	Description
storeId	The store identifier to find buddy stores for.

Input Data Definition

Attribute	Data Type	Required	Description
itemIds	List of Strings	Yes	A list of items to retrieve inventory for.
sellingUnitOfMeasure	Boolean	-	True indicates an attempt to use the selling unit of measure of the item, false indicates to use the standard unit of measure. If conversion cannot take place, it defaults back to standard unit of measure.

Example Input

{

"itemIds": [

"100637156",

"100637172",

"100668091"

],

"sellingUnitOfMeasure": true

}

Output Data Definition

Attribute	Data Type	Required	Description
itemId	String	Yes	The item identifier.
storeId	Long		The store identifier if the item is ranged to a store.
ranged	Boolean	Yes	True if the item is ranged to the store, false otherwise.
estimated	Boolean	Yes	True if the quantities are estimated, false otherwise.
unitOfMeasure	String	Yes	The unit of measure of the quantities.
caseSize	BigDecimal	Yes	The default case size of the item.
quantityStockOnHand	BigDecimal	Yes	The stock on hand quantity.
quantityBackroom	BigDecimal	Yes	The quantity located in the back room area.
quantityShopfloor	BigDecimal	Yes	The quantity located on the shop floor.
quantityDeliveryBay	BigDecimal	Yes	The quantity located in the delivery bay.
quantityAvailable	BigDecimal	Yes	The available to sell quantity.
quantityUnavailable	BigDecimal	Yes	The unavailable to sell quantity.
quantityNonSellable	BigDecimal	Yes	The total non-sellable quantity.
quantityInTransit	BigDecimal	Yes	The quantity currently in transit.
quantityCustomerReserved	BigDecimal	Yes	The quantity reserved for customer orders.
quantityTransferReserved	BigDecimal	Yes	The quantity reserved for transfers.
quantityVendorReturn	BigDecimal	Yes	The quantity reserved for vendor returns.
nonSellableQuantities	List of Non-Sellable Quantities	-	A collection containing the specific quantity in each non-sellable quantity type bucket.

Non-Sellable Quantity Data Definition

Attribute	Data Type	Required	Description
nonsellableTypeId	Long	Yes	The non-sellable type unique identifier.
quantity	quantity	Yes	The quantity in this particular non-sellable type bucket.

Example Output

[

{

"itemId": "100637156",
"storeId": 5000,
"ranged": true,
"estimated": false,
"unitOfMeasure": "EA",
"caseSize": 100.00,
"quantityStockOnHand": 10.0000,
"quantityBackroom": 10.0000,
"quantityShopfloor": 0.0000,
"quantityDeliveryBay": 0.0000,
"quantityAvailable": 10.0000,
"quantityUnavailable": 0.0000,
"quantityNonSellable": 0.0000,
"quantityInTransit": 0.0000,
"quantityCustomerReserved": 0.0000,
"quantityTransferReserved": 0.0000,
"quantityVendorReturn": 0.0000
},
{
"itemId": "100637172",
"storeId": 5000,
"ranged": true,
"estimated": false,
"unitOfMeasure": "EA",
"caseSize": 100.00,
"quantityStockOnHand": 10.0000,
"quantityBackroom": -10.0000,
"quantityShopfloor": 0.0000,
"quantityDeliveryBay": 0.0000,
"quantityAvailable": -10.0000,
"quantityUnavailable": 20.0000,
"quantityNonSellable": 20.0000,
"quantityInTransit": 0.0000,
"quantityCustomerReserved": 0.0000,

"quantityTransferReserved": 0.0000,
"quantityVendorReturn": 0.0000,
"nonSellableIdos": [
{
"nonsellableTypeId": 1,
"quantity": 15.0000
},
{
"nonsellableTypeId": 2,
"quantity": 5.0000
}
]
}
}

API: Find Inventory in Transfer Zone Stores

Searches for inventory at transfer zone stores by single input store and multiple items.

API Basics

Endpoint URL	{baseUrl}/{storeId}/transferzone
Method	POST
Successful Response	200 OK
Processing Type	Synchronous
Input	List of items
Output	List of inventory records
Max Response Limit	N/A

Path Parameter Definitions

Attribute	Description
storeId	The store identifier to find transfer zone stores for.

Input Data Definition

Attribute	Data Type	Required	Description
itemIds	List of Strings	Yes	A list of items to retrieve inventory for.

Attribute	Data Type	Required	Description
sellingUnitOfMeasure	Boolean	-	True indicates an attempt to use the selling unit of measure of the item, false indicates to use the standard unit of measure. If conversion cannot take place, it defaults back to standard unit of measure.

Example Input

{

"itemIds": [

"100637156",

"100637172",

"100668091"

],

"sellingUnitOfMeasure": true

}

Output Data Definition

Attribute	Data Type	Required	Description
itemId	String	Yes	The item identifier.
storeId	Long		The store identifier if the item is ranged to a store.
ranged	Boolean	Yes	True if the item is ranged to the store, false otherwise.
estimated	Boolean	Yes	True if the quantities are estimated, false otherwise.
unitOfMeasure	String	Yes	The unit of measure of the quantities.
caseSize	BigDecimal	Yes	The default case size of the item.
quantityStockOnHand	BigDecimal	Yes	The stock on hand quantity.
quantityBackroom	BigDecimal	Yes	The quantity located in the back room area.
quantityShopfloor	BigDecimal	Yes	The quantity located on the shop floor.
quantityDeliveryBay	BigDecimal	Yes	The quantity located in the delivery bay.
quantityAvailable	BigDecimal	Yes	The available to sell quantity.
quantityUnavailable	BigDecimal	Yes	The unavailable to sell quantity.
quantityNonSellable	BigDecimal	Yes	The total non-sellable quantity.
quantityInTransit	BigDecimal	Yes	The quantity currently in transit.
quantityCustomerReserved	BigDecimal	Yes	The quantity reserved for customer orders.
quantityTransferReserved	BigDecimal	Yes	The quantity reserved for transfers.

Attribute	Data Type	Required	Description
quantityVendorReturn	BigDecimal	Yes	The quantity reserved for vendor returns.
nonSellableQuantities	List of Non-Sellable Quantities	-	A collection containing the specific quantity in each non-sellable quantity type bucket.

Non-Sellable Quantity Data Definition

Attribute	Data Type	Required	Description
nonsellableTypeId	Long	Yes	The non-sellable type unique identifier.
quantity	quantity	Yes	The quantity in this particular non-sellable type bucket.

Example Output

[

{

"itemId": "100637156",

"storeId": 5000,

"ranged": true,

"estimated": false,

"unitOfMeasure": "EA",

"caseSize": 100.00,

"quantityStockOnHand": 10.0000,

"quantityBackroom": 10.0000,

"quantityShopfloor": 0.0000,

"quantityDeliveryBay": 0.0000,

"quantityAvailable": 10.0000,

"quantityUnavailable": 0.0000,

"quantityNonSellable": 0.0000,

"quantityInTransit": 0.0000,

"quantityCustomerReserved": 0.0000,

"quantityTransferReserved": 0.0000,

"quantityVendorReturn": 0.0000

},

{

"itemId": "100637172",

"storeId": 5000,
"ranged": true,
"estimated": false,
"unitOfMeasure": "EA",
"caseSize": 100.00,
"quantityStockOnHand": 10.0000,
"quantityBackroom": -10.0000,
"quantityShopfloor": 0.0000,
"quantityDeliveryBay": 0.0000,
"quantityAvailable": -10.0000,
"quantityUnavailable": 20.0000,
"quantityNonSellable": 20.0000,
"quantityInTransit": 0.0000,
"quantityCustomerReserved": 0.0000,
"quantityTransferReserved": 0.0000,
"quantityVendorReturn": 0.0000,
"nonSellableIdos": [
{
"nonsellableTypeId": 1,
"quantity": 15.0000
},
{
"nonsellableTypeId": 2,
"quantity": 5.0000
}
]
}
}

Rest Service: Item ISN

This rest service defines operations to manage Item ISN information.

Service Base URL

The Cloud service base URL follows the format:

https://<external_load_balancer>/<cust_env/siocs-int-services>/api/isns

APIs

API	Description
Create ISN	Create a new ISN
Update ISN	Updates an existing ISN
Delete ISN	Delete an existing ISN
Find ISNs	Search for ISNs based on a set of criteria
Read ISN Types	Read all Item ISN types

API: Create ISN

This API will create ISN information.

API Basics

Endpoint URL	{base URL}
Method	POST
Successful Response	200 OK
Processing Type	Synchronous
Input	The ISN
Output	The ISN including ID
Max Response Limit	N/A

Input Data Definition

Attribute	Type	Req	Definition
isn	String(128)	X	A scan number used to scan the item
isnTypeId	Long(12)	X	The unique identifier of the item ISN type
itemId	String(25)	X	The unique item identifier (sku number)
uin	String(128)		A universal identification number
externalId	String(128)		An identifier from an external system

Example

```
{
  "isn": "ABC123",
  "itemId": "100700500",
  "isnTypeId": 1,
  "uin": "123456789",
  "externalId": "TestId"
}
```

Output Data Definition

Attribute	Type	Definition
isnId	Long(12)	The unique identifier of the ISN record
isn	String(128)	This is a scan number used to scan the item
externalId	String(128)	An identifier from an external system

API: Update ISN

This API allows the ISN information to be modified. An optional value left blank will update the ISN information to blank for that value.

API Basics

Endpoint URL	{base URL}/isnId
Method	POST
Successful Response	204 No Content
Processing Type	Synchronous
Input	ISN Update Information
Output	N/A
Max Response Limit	N/A

Path Data Definiton

Attribute	Type	Definition
isnId	Long(12)	The unique identifier of the item ISN.

Input Data Definition

Attribute	Type	Req	Definition
isnTypeId	Long(12)	X	The unique identifier of the item ISN type
itemId	String(25)	X	The unique item identifier (sku number)
uin	String(128)		A universal identification number
externalId	String(128)		An identifier from an external system

Example


```
{
   "itemId": "100700500",
   "isnTypeId": 1,
   "uin": "123456789",
   "externalId": "TestId"
}
```

API: Delete ISN

This API deletes an item ISN.

API Basics

Endpoint URL	{base URL}/{isnId}/delete
Method	POST
Successful Response	204 No Content
Processing Type	Synchronous
Input	N/A
Output	N/A
Max Response Limit	N/A

Path Data Definition

Attribute	Type	Definition
isinId	Long(12)	The unique identifier of the item ISN.

API: Find ISNs

This API is used to lookup or find ISNs.

API Basics

Endpoint URL	{base URL}
Method	GET
Successful Response	200 OK
Processing Type	Synchronous
Input	N/A
Output	List of ISNs
Max Response Limit	10,000

Query Parameters

Attribute	Type	Definition
isn	String(128)	Retrieves records containing this ISN, which could be for multiple items.
ItemId	String(25)	Retrieves records associated to this unique item identifier (sku number).
uin	String(128)	Retrieves records associated to this universal identification number.
updateDateFrom	String	Retrieves records on or after this date.
updateDateTo	String	Retrieves records on or before this date.

Output Data Definition

Attribute	Type	Definition
itemIsnId	Long(12)	The unique identifier of the record
isn	String(128)	Retrieves records containing this ISN, which could be for multiple items.
ItemId	String(25)	Retrieves records associated to this unique item identifier (sku number).
uin	String(128)	Retrieves records associated to this universal identification number.
itemIsnTypeId	Long(12)	The unique identifier of the item ISN type
externalId	String(128)	An identifier of this ISN from an external system
createDate	Date	The date this ISN record was first created
createUser	String(128)	The user that created this ISN record
updateDate	Date	The last date this ISN record was updated
updateUser	String(128)	The user that last updated this ISN record.

API: Read ISN Types

This API is used to lookup all the ISN types available for an ISN.

API Basics

Endpoint URL	{base URL}/types
Method	GET
Successful Response	200 OK
Processing Type	Synchronous
Input	N/A
Output	List of ISN Types
Max Response Limit	N/A

Output Data Definition

Attribute	Type	Definition
isnTypeId	Long(12)	The unique ISN type identifier
labelKey	String(128)	A label for the ISN type (not translated)
restricted	Boolean	Y if this represents secure data, N otherwise.

REST Service: Item Price

This service allows for the search and retrieval of item pricing information stored within EICS.

Service Base URL

The Cloud service base URL follows the format:

https://<external_load_balancer></cust_env>/siocs-int-services/api/prices

API Definitions

API Definitions

API	Description
findPrices	This API can be used to search for price summary information matching filter criteria.
FindPriceByIds	Find extended price information based on a list of potential unique price identifiers.

API: findPrices

This API can be used to search for price summary information matching filter criteria.

At least one input criteria is required or a bad request error will be returned.

API Basics

Endpoint URL	{base URL}
Method	GET
Success Response	200 OK
Processing Type	Synchronous
Input	Query Parameters
Output	List of Prices
Maximum Results Allowed	10,000

Input Data Definition

Attribute	Data Type	Definition
storeId	Long(10)	Only retrieve item price information for this store.
effectiveDateFrom	String	Only retrieve item price information on or after this date.
effectiveDateTo	String	Only retrieve item price information on or before this date.
itemId	String(25)	Only retrieve item price information for this item.

Output Data Definition

Attribute	Data Type	Definition
priceId	Long(12)	The unique identifier of the price.
itemId	String(25)	The unique identifier of the item.
storeId	Long(10)	The unique identifier of the store.
status	Integer	The status of the price.
priceType	Integer	The type of the price.
effectiveDate	Date	The effective date of the price.
endDate	Date	The end date of the price.
priceValue	BigDecimal(20,4)	The price amount.
priceCurrency	String(3)	The price currency.

unitOfMeasure	String(4)	The item unit of measure associated with the price.

API: findPriceByIds

Find extended price information based on a list of potential unique price identifiers.

It will return information only for price identifiers that are found. It will not return errors or fail to process if invalid identifiers occur. It is up to the accessing information to determined prices not found using this API.

API Basics

Endpoint URL	{base URL}/find
Method	POST
Success Response	200 OK
Processing Type	Synchronous
Input	ID List
Output	List of Prices
Maximum Input Allowed	5,000

Input Data Definition

Attribute	Type	Definition
priceIds	List<Long(12)>	A list of price identifiers.

Output Data Definition

Attribute	Type	Definition
priceId	Long(12)	The unique identifier of the price.
itemId	String(25)	The unique identifier of the item.
storeId	Long(10)	The unique identifier of the store.
Status	Integer	The status of the price.
priceType	Integer	The type of the price
effectiveDate	Date	The effective date of the price.
endDate	Date	The end date of the price.
priceValue	BigDecimal(20, 4)	The price amount.
priceCurrency	String(3)	The price currency.
unitOfMeasure	String(4)	The item unit of measure associated with the price.
externalPriceId	Long(12)	The unique identifier of the external price or price event.
clearanceId	Long(15)	The unique identifier of the clearance price change from the pricing engine.
promotionId	Long(10)	The unique identifier of the promotion.
regularPriceChangeId	Long(15)	The unique identifier of the regular price change from the pricing engine.
resetClearanceId	Long(15)	The identifier of the clearance reset.
storeRequested	Boolean	True indicates it is store requested, false indicates it is not store requested.

sellingUnitPriceChange	Boolean	True indicates the selling unit retail price has changed, false indicates it has not.
multiUnitPriceChange	Boolean	True indicates the multi-unit pricing has changed, false indicates it has not.
multiUnitRetail	BigDecimal(20, 4)	The multi-unit retail price.
multiUnitRetailCurrency	String(3)	The currency type of the multi-unit retail price in the multi-selling unit of measure.
multiUnits	BigDecimal(12, 4)	The number of multi-units.
multiUnitUom	String(4)	The unit of measure of the multi-unit retail price.
promotionName	String(160	The promotion name.
promotionCompDtlId	Long(15)	The unique identifier of the promotion component detail from the pricing engine.
promotionType	Integer	Promotion Component Type (See Index)
promotionDurationType	Integer	Promotion Duration Type (See Index)
promotionCompId	Long(10)	The unique identifier of the promotion component.
promotionCompName	String(160)	The promotion component name.
promotionDescription	String(640)	The promotion description.
updateDate	Date	The date that the update took place.

Example Input:

```
{
"priceIds": [
123,
 456,
 789,
 012
  ]
}
```

Additional Data Definitions

Price Type

ID	Status
1	Permanent
2	Promotional
3	Clearance
4	Clearance Reset

Price Status

ID	Status
1	New
2	Pending
3	Approved

4	Completed
5	Rejected
6	Ticket List
7	Extract Failed
8	Deleted
99	Default

Promotion Component Type

ID	Status
1	Complex Promotion
2	Simple Promotion
3	Threshold Promotion
4	Credit (Finance) Promotion
5	Transaction Promotion

Promotion Duration Type

ID	Status
1	All Day Promotion
2	Partial Day Promotion
3	Multiple Day Promotion

REST Service Item Uda

This service integrates user defined attribute foundation data. Asynchronous item UDA integration is processed through staged messages and is controlled by the MPS Work Types.

REST Service: Item Price

This service allows for the search and retrieval of item pricing information stored within EICS.

Service Base URL

The Cloud service base URL follows the format:

https://<external_load_balancer>/<cust_env>/siocs-int-services/api/udas

API Definitions

API	Description
importUdas	Imports user defined attributes.
deleteUdas	Deletes user defined attributes.
importItemUdas	Imports an association between items and user defined attributes.
deleteItemUdas	Deletes the association between items and user defined attributes.
readItemUdas	Retrieves all the user defined attributes for a particular item.

API: Import Udas

Imports user defined attributes. It is managed by DcsUda work type. If the input exceeds 500 UDAs an input too large exception will be returned.

A "Forbidden" response will occur if application is integrated with MFCS or RMS.

API Basics

Endpoint URL	{base URL}/import
Method	POST
Successful Response	202 Accepted
Processing Type	Asynchronous
Input	UDA import List
Output	None
Max Response Limit	N/A

Input Data Definition

Attribute	Data Type	Required	Description
udas	List of details	Yes	A list of user defined attributes.

Detail Data Definition

Attribute	Data Type	Required	Description
udaId	Integer (5)	Yes	The user defined attribute identifier.
type	Integer	Yes	See Index: UdaType
description	String (120)	Yes	The description of the user defined attribute.
printTicket	Boolean		True indicates tickets are printed for this user defined attribute.
printLabel	Boolean		True indicates labels are printed for this user defined attribute.
lovId	String (25)		The unique identifier of a list of values UDA.
lovDescription	String (250)		The description of the list of values UDA.

API: Delete Udas

Deletes user defined attributes. It is managed by DcsUda work type. If the input exceeds 500 UDAs an input too large exception will be returned.

A "Forbidden" response will occur if application is integrated with MFCS or RMS.

API Basics

Endpoint URL	{base URL}/delete
Method	POST
Successful Response	202 Accepted
Processing Type	Asynchronous
Input	UDA delete list
Output	None
Max Response Limit	N/A

Input Data Definition

Attribute	Data Type	Required	Description
udaIds	List<Long>	Yes	A list of user defined attribute.

API: Import Item Udas

Imports associations between items and user defined attributes. This is controlled by the work type: DcsItem.

If the input exceeds 500 Item UDAs an input too large exception will be returned.

A "Forbidden" response will occur if application is integrated with MFCS.

API Basics

Endpoint URL	{base URL}/items/import
Method	POST
Successful Response	202 Accepted
Processing Type	Asynchronous
Input	Item UDA list
Output	None
Max Response Limit	N/A

Input Data Definition

Attribute	Data Type	Required	Description
itemUdas	List of details	x	A list of associations between an item and user defined attributes.

Detail Data Definition

Attribute	Data Type	Required	Description
itemId	String	X	The item identifier.

udaId	Integer	X	The user defined attribute identifier.
udaDate	Date		Holds the value of the user defined attribute if it is a date.
udaText	String (250)		Holds the value of the user defined attribute if it is text.
udaLovId	String (25)		Holds the unique numeric identifier of the user defined attribute if it is a list of values selection.
print	Boolean		Y indicates printing is done for this item and user defined attribute (which is also controlled by the UDA).

API: Delete Item Udas

Deletes an association between item and user defined attributes. This is controlled by the work type: DcsItem.

If the input exceeds 500 Item UDAs an input too large exception will be returned.

A "Forbidden" response will occur if application is integrated with MFCS.

API Basics

Endpoint URL	{base URL}/items/delete
Method	POST
Successful Response	202 Accepted
Processing Type	Asynchronous
Input	UDA item delete list
Output	None
Max Response Limit	N/A

Input Data Definition

Attribute	Data Type	Required	Description
itemUdas	List of details	Yes	A list of associations between item and user defined attribute identifiers.

Detail Data Definition

Attribute	Data Type	Required	Description
itemId	String (25)	Yes	The item identifier.

udaId	Integer (5)	Yes	The user defined attribute identifier.
udaDate	Date		Holds the value of the user defined attribute if it is a date.
udaText	String (250)		Holds the value of the user defined attribute if it is text.
udaLovId	String (25)		Holds the unique numeric identifier of the user defined attribute if it is a list of values selection.

API: Find Item Udas

It will retrieve UDAs for the inputted items.

API Basics

Endpoint URL	{base URL}/items/find
Method	POST
Successful Response	200 OK
Processing Type	Synchronous
Input	Item list
Output	UDA item list
Max Response Limit	N/A

Input Data Definition

Attribute	Data Type	Required	Description
itemIds	List<String>	Yes	A list of items to find UDAs for.

Output Data Definition

Attribute	Data Type	Description
itemId	String	The item identifier.
udaId	Integer	The user defined attribute identifier.
type	String	The user defined attribute type: (LV) List of Value, (FF) Free Form Text, DT (Date)
description	String	The description of the user defined attribute.
udaDate	Date	Holds the value of the user defined attribute if it is a date.
udaText	String	Holds the value of the user defined attribute if it is text.

udaLovId	Long	Holds the unique numeric identifier of the user defined attribute if it is a list of values selection.
udaLoveDescription	String	Holds the value description of the UDA List of Values selection.
printTicket	Boolean	True indicates tickets are printed for this user defined attribute.
printLabel	Boolean	True indicates labels are printed for this user defined attribute.

Additional Data Definitions

UDA Type

Value	Definition
1	Date
2	Free Form Text
3	List of Value

Rest Service: Item UIN

Service Base URL

The Cloud service base URL follows the format:

https://<external_load_balancer>/<cust_env>/siocs-int-services/api/uins

API Definitions

API	Description
createUin	Create a new unique identification number.
readUin	Reads information about a Universal Identification Number.
findUins	Find unique identification number summary information based on search criteria.
findUinLabels	This API is used to find all the UIN labels available for a uin items.
findUinHistory	This API is used to find UIN historical information based on search criteria.
generateUins	This API generates new Type 2 (Auto Generated Serial Numbers) Universal Identification Numbers without changing store inventory positions.

REST Service: Activity Lock

This service allows the creation, removal, and finding of activity locks.

An activity lock is a record indicating the user, time, and a piece of information (a transaction) that should be considered "locked". All server processing validates that the accessing user has

a lock on the information before updating, notifying the current user if someone else has modified the information while they were locked and preventing the stale update.

Developers should create locks on transactional information prior to performing update calls and delete locks when the update if finished. For example, create a lock on inventory adjustment with ID 123 with the ActivityLock service, then use `saveInventoryAdjustment`in the Inventory Adjustment service with Adjustment 123, and then delete the activity lock using the ActivityLock service. If you do not gain the lock, you will receive an error when attempting to save an inventory adjustment.

API: createUin

Create a new unique identification number. Note that the combination of store and item determines the administrative information about a UIN (such as UIN Type).

The newly created UIN will be in "Unconfirmed" status and its transaction type will be "UIN Web Service." To move it into inventory, use inventory adjustment or another transaction.

API Basics

Endpoint URL	{base URL}
Method	POST
Success Response	200 OK
Processing Type	Synchronous
Input	UIN information
Output	UIN confirmation information

Input Data Definition

Payload	Type	Req	Definition
storeId	Long	X	The identifier of the store.
itemId	String	X	The identifier of the item.
uin	String	X	The universal identification number.

Output Data Definition

Payload	Type	Definition
itemUinId	Long	The unique identifier to the record.
storeId	Long	The identifier of the store.
itemId	String	The identifier of the item.
uin	String	The universal identification number.
status	Integer	The current status of the UIN. Valid values are in index.

Example

```
{
```

"storeId": 5000,,
"itemId": "100700500",
"uin": "1234"
}

API: readUin

Reads information about a Universal Identification Number.

API Basics

Endpoint URL	{base URL}/items/{itemId}/{uin}
Method	GET
Success Response	200 OK
Processing Type	Synchronous
Input	Path Parameters Item identifier and UIN
Output	UIN information

Output Data Definition

Payload	Type	Definition
itemUinId	Long	A unique identifier representing the record in the database.
itemId	String	The identifier of the item.
uin	String	The universal identification number.
type	Integer	The type of UIN. Valid values are: (1) Serial Number, (2) Auto Generated Serial Number
status	Integer	The current status of the UIN. See Index for valid values.
storeId	Long	The store identifier
transactionType	Integer	The business area that last contained the UIN,
transactionId	String	The transaction id of the transaction containing the UIN.
cartonId	String	The identifier of the carton containing the UIN.
nonsellableTypeId	Long	A non-sellable inventory bucket the UIN was within.
previousStatus	Integer	The previous status of the UIN. Valid values are in index.
previousStoreId	Long	The previous store identifier associated with the previous status.
previousTransactionType	Integer	The previous business area that contained the UIN for that previous status.
previousTransactionId	String	The transaction id of the transaction that previously contained the UIN for that previous status.
previousCartonId	String	The identifier of the carton that previously container the UIN for that previous status.
previousNonsellableTypeId	Long	A non-sellabable inventory bucket the UIN was last within for that previous status.
damaged	Boolean	True if the UIN is damaged, N otherwise.

createDate	Date	The date the UIN was first inserted into the system.
updateDate	Date	The last date the UIN was updated.
createUser	String	The user that first inserted the UIN into the system.
updateUser	String	The user that last updated the UIN in the system.

API: findUins

Find unique identification number summary information based on search criteria.

API Basics

Endpoint URL	{base URL}
Method	GET
Success Response	200 OK
Processing Type	Synchronous
Input	Query Parameters
Output	List of UINs (see ReadUIN API for Data Output)
Maximum Results Allowed	10,000

Input Data Definition

Attribute	Type	Definition
storeId	Long	Include only UINs for this store identifier.
itemId	String	Include only UINS for this item
status	Integer	Include only UINs with this current status.
updateDateFrom	Date/String	Include only UINs updated on or after this date.
updateDateTo	Date/String	Include only UINs updated on or before this date.

API: findUinLabels

This API is used to find all the UIN labels available for a uin items.

API Basics

Endpoint URL	{base URL}/labels
Method	GET
Success Response	200 OK
Processing Type	Synchronous
Input	N/A
Output	List of labels

Output Data Definition

Attribute	Type	Definition
labelId	Long	The unique identifier of the record.

labelCode	String	A unique code that defines the label.
description	String	The description or label associated to the code (not translated).

API: findUinHistory

This API is used to find UIN historical information based on search criteria.

API Basics

Endpoint URL	{base URL}/histories
Method	GET
Success Response	200 OK
Processing Type	Synchronous
Input	Query parameters
Output	List of UIN history records
Maximum Results Allowed	10,000

Input Data Definition

Attribute	Type	Definition
itemId	String	Include only UIN history for this item.
uin	Integer	Include only UIN history for this UIN.
createDateFrom	Date/String	Include only UIN history created on or after this date.
createDateTo	Date/String	Include only UIN history created on or before this date.

Output Data Definition

Payload	Type	Definition
itemId	String	The identifier of the item.
uin	String	The universal identification number.
type	Integer	The type of UIN. Valid values are: (1) Serial Number, (2) Auto Generated Serial Number
status	Integer	The current status of the UIN. Valid values are in index.
storeId	Long	The store identifier
transactionType	Integer	The business area that last contained the UIN,
transactionId	String	The transaction id of the transaction containing the UIN.
cartonId	String	The identifier of the carton containing the UIN.
nonsellableTypeId	Integer	A non-sellable inventory bucket the UIN was within.
createDate	Date	The date the UIN was first inserted into the system.

updateDate	Date	The last date the UIN was updated.
createUser	String	The user that first inserted the UIN into the system.
updateUser	String	The user that last updated the UIN in the system.

API: generateUins

This API generates new Type 2 (Auto Generated Serial Numbers) Universal Identification Numbers without changing store inventory positions.

If the UIN administrative data for the item and store used do not indicate Type 2, an error will be returned.

The new UINs generated will have a status of “Unconfirmed”.

API Basics

Endpoint URL	{base URL}/generate
Method	POST
Success Response	200 OK
Processing Type	Synchronous
Input	UIN generation information
Output	N/A

Input Data Definition

Payload	Type	Req	Definition
itemId	String	X	The identifier of the item.
storeId	Long	X	The identifier of the store
quantity	Integer	X	The amount of universal identification numbers to generate.
transactionType	Integer	X	See Index: UIN Functional Area
transactionId	String		A transaction reference identifier.

Example

```
{
   "storeId": 5000,
   "itemId": "100663071",
   "uin": "1234",
  "quantity": 5,
    "transactionType": 13
}
```

Additional Data Definitions

UIN Type

ID	Description
1	Serial Number
2	Auto-Generated Serial Number

UIN Status

ID	Description
1	In Stock
2	Sold
3	Shipped To Warehouse
4	Shipped To Store
5	Reserved For Shipping
6	Shipped To Vendor
7	Removed From Inventory
8	Unavailable
9	Missing
10	In Receiving
11	Customer Order Reserved
12	Customer Order Fulfilled
13	Shipped To Finisher
99	Unconfirmed

UIN Functional Area

ID	Description
1	Warehouse Delivery Receipt
2	Direct Delivery Receipt
3	Create Transfer
4	Dispatch Transfer
5	Receive Transfer
6	Receipt Adjustment
7	Create Return
8	Dispatch Return
9	Inventory Adjustment
10	Stock Count
11	Stock Recount
12	Stock Count Authorized
13	Manual
14	POS Sale
15	POS Return

16	POS Sales Void
17	POS Return Void
18	UIN Web Service
19	Customer Order
20	Direct Delivery ASN Inbound
21	Transfer ASN Inbound
22	Transfer Shipment

REST Service: Manifest

Service Base URL

The Cloud service base URL follows the format:

https://<external_load_balancer>/<cust_env>/siocs-int-services/api/manifests

API Definitions

API	Description
Close Manifest	Call this method to close all manifested shipments matching the input criteria.

API: Close Manifest

Call this method to close all manifested shipments for the carrier code and carrier services. A processing message is sent to the internal message processing system and the services returns an “Accepted” response. The closing of the manifest will occur later when the message is processed.

API Basics

Endpoint URL	{base URL}/close
Method	POST
Successful Response	202 Accepted
Processing Type	Asynchronous
Input	Criteria
Output	None
Max Response Limit	-

Input Data Definition

Attribute	Data Type	Required	Description
carrierCode	String (4)	Yes	A carrier code.
carrierServiceCode	String (6)	Yes	A carrier service code.

trackingNumber	List of Strings (120)	Yes	A list of tracking numbers associated to the contents of the manifest.
shipDate	Date	-	Indicates all items manifested prior to this date for the carrier have been shipped.

Example Input

```
{
    "carrierCode": "O",
    "carrierServiceCode": "O",
    "trackingIds": ["7861","45722"],
    "shipDate":"2022-04-19T23:59:59-05:00"
}
```

REST Service: POS Transaction

This service retrieves information about item inventory.

Service Base URL

The Cloud service base URL follows the format:

https://<external_load_balancer>/<cust_env>/siocs-int-services/api/postransactions

API Definitions

API	Description
Import POS Transactions	Imports point-of-sale transactions.

API: Import POS Transactions

POS may integration its transaction to EICS using this web service. The service imports and process point-of-sale transactions through an asynchronous process. The service has a default limit of 1000 total items, though they may be distributed across any number of transactions. Only one store is allowed across all the transaction sent in a single requires.

The web service is optimized for speed at greater than 400 items and less than 500 items per service call. The further above or below this optimized point, processing speed will be reduced, and it may take longer for the sales to be recorded in inventory.

Since this import is asynchronous, only the form is validated prior to the data being captured and processed later. See Sales Integration for additional information about processing.

API Basics

Endpoint URL	{base URL}
Method	POST

Successful Response	202 Accepted
Processing Type	Asynchronous
Input	List of transactions
Output	None
Max Input Limit	-

Input Data Definition

Attribute	Data Type	Required	Description
Transactions	List of PosTransactions	Yes	A list of transactions to process.

Pos Transaction Data Definition

Attribute	Data Type	Required	Description
storeId	Long	Yes	The unique identifier of the store that is the source of the transaction.
transactionId	String (128)	Yes	A unique identifier of the point-of-sale transaction.
transactionTimestamp	Date	Yes	The date and time of the transaction.
custOrderId	String (128)	-	An external customer order identifier. This attribute is required for customer order related point-of-sale transactions.
customerOrderComment	String (512)	-	A comment associated to the customer order.
externalUser	String (128)	-	User information from the external point-of-sale system.
items	List of Items	Yes	A collection of items belonging to the transaction.

Pos Transaction Item Data Definition

Attribute	Data Type	Required	Description
itemId	String (25)	Yes	The transaction-level SKU number of the item.
quantity	BigDecimal	Yes	The quantity of the item transacted.
unitOfMeasure	String (4)	Yes	Unit of measure of the quantity.
uin	String (128)	-	The unique identifier number (serial number). If not empty, the quantity will be overwritten with a quantity of one.
epc	String (256)	-	A complete SGTIN-96 EPC of the item.

Attribute	Data Type	Required	Description
reasonCode	Integer	-	A reason code associated to the line item. This reason code represents the inventory movement reason. If the field is left blank, the appropriate inventory movement reason code will be defaulted based on the type of sale. This field is required when non-sellable sub-level inventory tracking is active in the system.
dropShip	Boolean	-	True if this item is a drop ship, false if it is not. Drop ship sales do not impact stock positions.
fulfillmentOrderId	String (128)	-	If the transaction is associated to a customer order, this is the external fulfillment order identifier.
fulfillmentOrderLineNumber	Long	-	If the transaction is associated to a customer order, this is the line number of the order that this item transaction aligns with.
reservationType	Integer	-	If the transaction is a customer order, this is the type of reservation. See Reservation Type.
transactionCode	Integer	Yes	A code that indicates the transaction event that took place on the item. See Transaction Codes.
comments	String (512)	-	Comments associated to the line item.

Example Input

{

"transactions":

[

{

"storeId": 5000,

"transactionId": 1236,

"transactionTimestamp": "2022-04-19T23:59:59-05:00",

"externalUser": "ABC",

"custOrderId": "1111",

"items":

[

{

"itemId": 5678,

"transactionCode": 5,

"reservationType": 1,

"fulfillOrderId": "2222",

"dropShip": false

}

]

}

]

}

Possible Business Exception Codes

In addition to the normal REST error codes, the following business data element may be returned when a business error occurs.

Business Exception Data Definition

Name	Definition
DUPLICATE_TRANSACTION	One or more the transactions or transaction items are not unique. This will cause the entire request to be rejected.

Additional Data Definitions

Reservation Type

Value	Definition
1	Web Order
2	Special Order
3	Pickup or Delivery
4	Layaway
5	On Hold

Transaction Code

Value	Definition
1	Sale
2	Return
3	Void Sale
4	Void Return
5	Order New
6	Order Fulfill
7	Order Cancel

REST Service: Reason Code

This service allows and external system to retrieve available reason codes. Reason codes are attached to inventory adjustments or shipments.

Service Base URL

The Cloud service base URL follows the format:

https://<external_load_balancer>/<cust_env>/siocs-int-services/api/ reasoncodes

API Definitions

API	Description
Find Adjustment Reason Codes	Finds reason codes available for inventory adjustments.
Find Shipment Reason Codes	Finds reason codes available for shipments.

API: Find Adjustment Reason Codes

This API is used to find reason codes available for inventory adjustments.

API Basics

Endpoint URL	{base URL} /adjustments
Method	GET
Successful Response	200 OK
Processing Type	Synchronous
Input	None
Output	List of inventory adjustment reasons
Max Response Limit	N/A

Output Data Definition

Attribute	Data Type	Description
reasonId	Long	The unique identifier of the inventory adjustment reason code.
reasonCode	Integer	Unique reason code associated to external systems.
description	String	A description of the inventory reason code.
dispositionCode	Integer	The inventory disposition associated to the code.
dispositionDescription	String	A description of the inventory disposition associated to the code (not translated).
fromNonSellableType	Long	From unavailable sub-bucket (indicates the sub-bucket of disposition of stock movement)
fromNonSellableTypeDescription	String	The description of the from unavailable sub-bucket (not translated).

toNonSellableType	Long		To unavailable sub-bucket (indicates the sub-bucket of disposition of stock movement)
toNonSellableTypeDescription	String		The description of the to unavailable sub-bucket (not translated).
systemRequired	Boolean		True indicates the reason code is required for the system to function. A system required reason code cannot be deactivated.
displayable	Boolean		True indicates the reason code can be used by a transactional inventory adjustment created by an entity other than EICS internal service.
publish	Boolean	-	True indicating inventory movements with this reason code should be published to external systems

Example Input

[

{

"reasonId": 1,

"reasonCode": 1,

"description": “invAdjReason.1",

"dispositionCode": 4,

"dispositionDescription": "inventoryDisposition.ATS-DIST",

"systemRequired": true,

"displayable": false,

"publish": true

},

{

"reasonId": 2,

"reasonCode": 81,

"description": "invAdjReason.81",

"dispositionCode": 4,

"dispositionDescription": "inventoryDisposition.ATS-DIST",

"systemRequired": false,

"displayable": true,

"publish": true

}

]

API: Find Shipment Reason Codes

This API is used to find the reason codes available for shipments.

API Basics

Endpoint URL	{base URL} /shipments
Method	GET
Successful Response	200 OK
Processing Type	Synchronous
Input	None
Output	List of reasons codes
Max Response Limit	N/A

Output Data Definition

Attribute	Data Type	Description
reasonId	Long	The unique identifier of the inventory adjustment reason code.
reasonCode	String	Unique reason code associated to external systems.
description	String	A description of the inventory reason code (not translated).
type	Integer	The Shipment Reason Code Type: See the Shipment Reason Code Type.
useAvailable	Boolean	True if it should use available inventory, false otherwise.
nonSellableTypeId	Long	An identifier of associated unavailable sub-bucket (indicates the sub-bucket of disposition of stock movement)

Example Input

[

{

"reasonId": 1,

"reasonCode": "F",

"description": "shipmentReason.4.F",

"type": 4,

"useAvailable": true

},

{

“reasonId": 2,

"reasonCode": "O",
"description": "shipmentReason.4.O",
"type": 4,
"useAvailable": true
},
{
"reasonId": 3,
"reasonCode": "U",
"description": "shipmentReason.4.U",
"type": 4,
"useAvailable": false,
"nonSellableTypeId": 1
}
]

Additional Data Definitions

Shipment Reason Code Type

Value	Definition
1	Store
2	Supplier
3	Warehouse
4	Finisher
5	Customer

REST Service: Stock Count

The stock count services handle tasks related to a stock count.

Service Base URL

The Cloud service base URL follows the format:

https://<external_load_balancer>/<cust_env>/siocs-int-services/api/stockcounts

API Definitions

API	Description
Snapshot Count	Snapshots a stock count capturing the current stock on hand quantities.

API: Snapshot Stock Count

Executes a snapshot of the stock count capturing the current stock on hand quantity of each item on the count. The process of doing a snapshot first determines whether the stock count needs a snapshot and only snapshots those stock counts or stock count children that need a snapshot. If the stock count or stock count child does not need a snapshot, the service is considered successful.

API Basics

Endpoint URL	/{stockCountId}/snapshot
Method	POST
Successful Response	204 No Content
Processing Type	Synchronous
Input	None
Output	None

Path Parameter Definitions

Attribute	Definition
stockCountId	The internal identifier of the stock count header.

REST Service: Store

This service integrates the store foundation data with an external application. Store integration is controlled by the MPS Work Type: DcsStore.

Service Base URL

The Cloud service base URL follows the format:

https://<external_load_balancer>/<cust_env>/siocs-int-services/api/stores

API Definitions

API	Description
Import Stores	Imports a collection of stores.
Delete Store	Deactivate a single store.
Read Store	Read store information based on an identifier (or link)
Find Stores	Lookup store information based on a collection of store identifiers.
Find Associated Stores	Lookup store associated to the specified input store.
Find Auto Receive Stores	Lookup stores that are allowed to auto receive from the specified input store.
Find Transfer Zone Stores	Lookup stores that are in the same transfer zone as the specified input store.
Find Adjustment Reason Codes	Finds reason codes available for inventory adjustments.

API	Description
Find Shipment Reason Codes	Finds reason codes available for shipments.

API: Import Stores

Imports a collection of stores. This allows 1,000 stores per input call. All imported stores will be inventory-holding regular stores.

A "Forbidden" response will occur if application is integrated with MFCS or RMS.

API Basics

Endpoint URL	{base URL}/import
Method	POST
Successful Response	202 Accepted
Processing Type	Asynchronous
Input	List of Store to Import
Output	None
Max Response Limit	1000

Input Data Definition

Attribute	Data Type	Required	Description
stores	List of stores to import	Yes	A collection of up to 1000 stores to import.

Stores Data Definition

Attribute	Data Type	Required	Description
storeId	Long(10)	Yes	The store identifier
storeName	String(150)	Yes	The name of the store.
languageCode	String(3)	-	The language code of the store.
countryCode	String(3)	-	The country code of the store.
currencyCode	String(3)	-	The currency code of the store.
timezone	String(80)	-	The timezone of the store.
transferZoneId	String(128)	Yes	The transfer zone identifier of the store.
organizationUnitId	String(15)	-	The organization unit identifier of the store.
managedStore	Boolean	-	True indicates that EICS manages the inventory, false indicates it does not.
customerOrdering	Boolean	-	True indicates this store can take customer orders, false indicates it does not.

Example Output

{

"stores": [
{
"storeId": 5002,
"storeName": "Leamington Spa",
"languageCode": "EN",
"countryCode": "US",
"currencyCode": "USD",
"timezone": "America/Los_Angeles",
"transferZoneId": "1000",
"organizationUnitId": "1111",
"managedStore": true,
"allowsCustomerOrders": false
},
{
"storeId": 5003,
"storeName": "Leamington Spa",
"languageCode": "EN",
"countryCode": "US",
"currencyCode": "USD",
"timezone": "America/Los_Angeles",
"transferZoneId": "1000",
"organizationUnitId": "1111",
"managedStore": true,
"allowsCustomerOrders": false
}
]
}

API: Delete Store

Delete a store. Prior to placing the request to delete into the MPS queue, it validates that the store exists and that the store contains no items ranged to it.

A "Forbidden" response will occur if application is integrated with MFCS or RMS.

API Basics

Endpoint URL	{base URL}/{storeId}/delete

Method	POST
Successful Response	202 Accepted
Processing Type	Asynchronous
Input	None
Output	None
Max Response Limit	N/A

Path Parameter Definitions

Attribute	Definition
storeId	The internal identifier of the store.

API: Read Store

Retrieve information about a store based on a single unique store identifier or link.

API Basics

Endpoint URL	{base URL}/{storeId}
Method	GET
Successful Response	200 OK
Processing Type	Synchronous
Input	None
Output	Store
Max Response Limit	N/A

Path Parameter Definitions

Attribute	Definition
storeId	The internal identifier of the store.

Output Data Definition

Attribute	Data Type	Description
storeId	Long	The store identifier
storeName	String	The name of the store.
languageCode	String	The language code of the store
countryCode	String	The country code of the store.
currencyCode	String	The currency code of the store
timezone	String	The timezone of the store
transferZoneId	String	The transfer zone identifier of the store
organizationUnitId	String	The organization unit identifier of the store.
managedStore	boolean	True indicates that EICS manages the inventory, false indicates it does not.

customerOrdering	boolean	True indicates this store can take customer orders, false indicates it does not.

Example Output


```
{
   "links": [
      {
         "href": "/stores/5000",
         "rel": "self"
      },
      {
         "href": "/stores/5000/delete",
         "rel": "delete"
      }
   ],
   "storeId": 5000,
   "storeName": "Solihull",
   "languageCode": "EN",
   "countryCode": "US",
   "currencyCode": "USD",
   "timezone": "America/Chicago",
   "transferZoneId": "1000",
   "organizationUnitId": "1111",
   "managedStore": true,
   "allowsCustomerOrders": false
}
```


API: Find Stores

Find stores based on a list of potential unique store identifiers. It allows a maximum of 1500 store identifiers.

API Basics

Endpoint URL	{base URL}/find
Method	POST
Successful Response	200 OK

Processing Type	Synchronous
Input	List of stores ids
Output	List of stores
Max Response Limit	1500

Input Data Definition

Attribute	Data Type	Required	Description
stores	List of stores Ids	Yes	A collection of up to 1500 stores to read.

Example Input

```
{
"storeIds": [
5000,
5001
]
}
```

Output Data Definition

Attribute	Data Type	Description
Stores	List of Stores	A collection containing the store information

Stores Data Definition

Attribute	Data Type	Required	Description
storeId	Long	Yes	The store identifier
storeName	String		The name of the store.
languageCode	String		The language code of the store
countryCode	String		The country code of the store.
currencyCode	String		The currency code of the store
timezone	String		The timezone of the store
transferZoneId	String		The transfer zone identifier of the store
organizationUnitId	String		The organization unit identifier of the store.
managedStore	boolean		True indicates that EICS manages the inventory, false indicates it does not.
customerOrdering	boolean		True indicates this store can take customer orders, false indicates it does not.

Example Output


```
[
   {
      "links": [
         {
            "href": "/stores/5000",
            "rel": "self"
         },
         {
            "href": "/stores/5000/delete",
            "rel": "delete"
         }
      ],
      "storeId": 5000,
      "storeName": "Solihull",
      "languageCode": "EN",
      "countryCode": "US",
      "currencyCode": "USD",
      "timezone": "America/Chicago",
      "transferZoneId": "1000",
      "organizationUnitId": "1111",
      "managedStore": true,
      "allowsCustomerOrders": false
   },
   {
      "links": [
         {
            "href": "/stores/5001",
            "rel": "self"
         },
         {
            "href": "/stores/5001/delete",
            "rel": "delete"
```

```
        }
      ],
      "storeId": 5001,
      "storeName": "Nottingham",
      "languageCode": "EN",
      "countryCode": "US",
      "currencyCode": "USD",
      "timezone": "America/New_York",
      "transferZoneId": "1000",
      "organizationUnitId": "1111",
      "managedStore": true,
      "allowsCustomerOrders": false
    }
]
```


API: Find Associated Stores

Find potential associated stores (buddy stores) to the specified input store.

API Basics

Endpoint URL	{base URL}/{storeId}/associated
Method	GET
Successful Response	200 OK
Processing Type	Synchronous
Input	None
Output	List of stores
Max Response Limit	N/A

Path Parameter Definitions

Attribute	Definition
storeId	The internal identifier of the store.

Output Data Definition

Attribute	Data Type	Required	Description
Stores	List of Stores Ids	Yes	A collection containing the store Ids

Stores Data Definition

Attribute	Data Type	Required	Description

storeId	Long	Yes	The internal identifier of the store.

Example Output

[
{
"links": [
{
"href": "/stores/5001",
"rel": "self"
},
{
"href": "/stores/5001/delete",
"rel": "delete"
}
],
"storeId": 5001
},
{
"links": [
{
"href": "/stores/5002",
"rel": "self"
},
{
"href": "/stores/5002/delete",
"rel": "delete"
}
],
"storeId": 5002
}
]

API: Find Auto Receive Stores

Find stores that are allowed to auto receive from the specified input store.

API Basics

Endpoint URL	{base URL}/{storeId}/autoreceive
Method	GET
Successful Response	200 OK
Processing Type	Synchronous
Input	None
Output	List of stores
Max Response Limit	N/A

Path Parameter Definitions

Attribute	Definition
storeId	The internal identifier of the store.

Example Output

[
{
"links": [
{
"href": "/stores/5001",
"rel": "self"
},
{
"href": "/stores/5001/delete",
"rel": "delete"
}
],
"storeId": 5001
},
{
"links": [
{
"href": "/stores/5002",
"rel": "self"
},
{

"href": "/stores/5002/delete",

"rel": "delete"

}

],

"storeId": 5002

}

]

API: Find Transfer Zone Stores

Find stores that are available within the transfer zone of the input store.

API Basics

Endpoint URL	{base URL}/{storeId}/transferzone
Method	GET
Successful Response	200 OK
Processing Type	Synchronous
Input	None
Output	List of stores
Max Response Limit	N/A

Path Parameter Definitions

Attribute	Definition
storeId	The internal identifier of the store.

Output Data Definition

Attribute	Data Type	Required	Description
Stores	List of Stores Ids	Yes	A collection containing the store Ids

Stores Data Definition

Attribute	Data Type	Required	Description
storeId	Long	Yes	The internal identifier of the store.

Example Output

[

{

"links": [

{

"href": "/stores/5001",

"rel": "self"

},

{

"href": "/stores/5001/delete",

"rel": "delete"

}

],

"storeId": 5001

},

{

"links": [

{

"href": "/stores/5002",

"rel": "self"

},

{

"href": "/stores/5002/delete",

"rel": "delete"

}

],

"storeId": 5002

}

]

REST Service: Store Item

This service integrates the store item foundation data with an external application. Store integration is controlled by the MPS Work Type: DcsItemLocation.

Service Base URL

The Cloud service base URL follows the format:

https://<external_load_balancer>/<cust_env>/siocs-int-services/api/storeitems

API Definitions

API	Description
Import Store Items	Imports items at a store location.
Remove Store Items	Deactivate items at a store location.
Import Replenishment Items	Imports replenishment item information.
Remove Replenishment Items	Deactivate replenishment item information.

API: Import Store Items

Imports store items into the system.

If more than 10,000 items are included in a single call, an input too large error will be returned.

A "Forbidden" response will occur if application is integrated with MFCS or RMS.

API Basics

Endpoint URL	{base URL}/import
Method	POST
Successful Response	202 Accepted
Processing Type	Asynchronous
Input	List of store Items to Import
Output	None
Max Response Limit	10,000

Input Data Definition

Attribute	Data Type	Required	Description
items	List of stores items to import	Yes	A collection of up to 10,000 stores items to import.

Stores Items Data Definition

Attribute	Data Type	Required	Description
itemId	String(25)	Yes	The unique item identifier (sku number).
shortDescription	String(255)	-	A short description of the item at this store.
longDescirption	String(400)	-	A long description of the item at this store.
status	String	Yes	See Index (Item Status)
primarySupplierId	Long(10)	-	The unique identifier of the primary supplier of the item to this store location.
storeControlPricing	Boolean	-	True indicates the item price can be controlled by the store.
rfid	Boolean	-	True indicates the item is RFID tagged.
defaultCurrencyCode	String(3)	-	The default currency of the item's price at this store.

purchaseType	Long	-	See Index (Purchase Type)
uinType	Integer	-	See Index (UIN Type)
uinCaptureTime	Integer	-	See Index (UIN Capture Time)
uinLabelCode	Long	-	The UIN label unique identifier.
uinExternalCreateAllowed	Boolean	-	True if an external system can create a UIN, false otherwise.

Example Input

{

"items": [

{

"itemId": "100637121",

"defaultCurrencyCode": "USB",

"longDescription": "TestDescriptionAA",

"primarySupplierId": 1,

"purchaseType": 1,

"rfid": true,

"shortDescription": "TestShortAA",

"status": 1,

"storeControlPricing": false,

"uinCaptureTime": 1,

"uinExternalCreateAllowed": true,

"uinLabelCode": "SN",

"uinType": 1

}

]

}

Additional Data Definitions

Item Status

Value	Definition
1	Active
2	Discontinued
3	Inactive
4	Auto Stockable

Purchase Type

Value	Definition
1	Normal Merchandise
2	Consignment Stock
3	Concession Items

UIN Capture Time

Value	Definition
1	Sale
2	Store Receiving

UIN Type

Value	Definition
1	Serial Number
2	Auto-Generated Serial Number

API: Remove Store Items

This will mark items as no longer usable. When all data is cleared out of transactions that reference this data, later batch jobs will eventually delete the material.

If more than 1000 items are included in a single call, an input too large error will be returned.

A "Forbidden" response will occur if application is integrated with MFCS or RMS.

API Basics

Endpoint URL	{base URL}/{storeId}/remove
Method	POST
Successful Response	202 Accepted
Processing Type	Asynchronous
Input	None
Output	None
Max Input Limit	1000

Path Parameter Definitions

Attribute	Definition
storeId	The internal identifier of the store.

Input Data Definition

Attribute	Data Type	Required	Description
items	List of items to remove	Yes	A collection of up to 1000 items to remove.

Example Input

{
"itemIds": [
"100637121",
"100637113"
]
}

API: Import Replenishment Items

Imports item replenishment information for an item at a store location.

If more than 1000 items are included in a single call, an input too large error will be returned.

A "Forbidden" response will occur if application is integrated with MFCS or RMS.

API Basics

Endpoint URL	{base URL}/{storeId}/replenish/import
Method	POST
Successful Response	202 Accepted
Processing Type	Asynchronous
Input	None
Output	List of Items Replenishment to import
Max Input Limit	1000

Path Parameter Definitions

Attribute	Definition
storeId	The internal identifier of the store.

Input Data Definition

Attribute	Data Type	Required	Description
items	List of Item Replenishment Import	Yes	The item replenishment information to import.

Item Replenishment Import Data Definition

Attribute	Data Type	Required	Description
itemId	String (25)	Yes	The unique item identifier (sku number).
replenishmentMethod	String(6)		A code representing the replenishment method. (SO indicates Store Order).

rejectStoreOrder	Boolean	True indicates store orders must be on or after the next delivery date or should be rejected.
multipleDeliveryPerDayAllowed	Boolean	True indicates the item allows multiple deliveries per day at the location.
nextDeliveryDate	Date	The next delivery date of the time based on its replenishment type.

Example Input

{

"items": [

{

"itemId": "100637121",

"replenishmentMethod": "AB",

"rejectStoreOrder": false,

"multipleDeliveryPerDayAllowed": false,

"nextDeliveryDate": "2022-11-19T23:59:59-05:00"

}

]

}

API: Remove Replenishment Items

Clears the replenishment item information from within the store item setting the replenishment properties to empty, null, or default flag settings.

If more than 1000 items are included in a single call, an input too large error will be returned.

A "Forbidden" response will occur if application is integrated with MFCS or RMS.

API Basics

Endpoint URL	{base URL}/replenish/remove
Method	POST
Successful Response	200 OK
Processing Type	Asynchronous
Input	List of item ids
Output	None
Max Input Limit	1000

Input Data Definition

Attribute	Data Type	Required	Description
Items	List of item Ids	Yes	A collection of up to 1000 items to read.

Example Input

{
"itemIds": [
"100637121",
"100637113"
]

REST Service: Store Order

This service allows query and approval of store orders.

Service Base URL

The Cloud service base URL follows the format:

https://<external_load_balancer>/<cust_env>/siocs-int-services/api/storeorders

APIs

API	Description
Find Orders	Finds store order header records based on search criteria
Read Order	Reads a store order
Approve Order	Approves the store order and notifies external system
Cancel Order	Cancels the store order and notifies external system

API: Read Order

API Basics

Endpoint URL	{base URL}/{storeOrderId}
Method	GET
Successful Response	200 OK
Processing Type	Synchronous
Input	N/A
Output	The store order

Max Response Limit	NA

Output Data Definition

Attribute	Type	Definition
storeOrderId	Long(12)	The unique identifier of the record.
storeId	Long(10)	The unique identifier of the store.
externalTransferId	String(128)	An external transfer identifier that this store order is associated to.
externalPurchaseOrderId	String(128)	An external purchase order identifier that this store order is associated to.
externalReference	String(128)	A reference to the store order in an external system.
parentReference	Long(12)	Reference to a parent transfer or order.
description	String(2000)	A description of the store order or cause of the store order.
status	Integer(2)	The status: See Index
origin	Integer(2)	The origin: See Index
requestedDeliveryDate	Date	The date that the store requests the delivery arrive by.
autoApproveDate	Date	The date the record was automatically approved in EICS.
addItemsAllowed	Boolean	Y indicates new items can be added to the store order.
replenishmentItemsOnly	Boolean	Y indicates only store order replenishment items can be added to the store order.
contextId	Long(18)	A context identifier associated to the store order.
deliverySlotId	Long(15)	The unique identifier of the delivery time.
productGroupId	Long(12)	The unique identifier of a product group associated to the order.
productGroupDescription	String(250)	The description of the product group associated to the order.
restrictToSupplierId	Long(12)	Allow only items on the store order that are associated to this supplier.
restrictToWarehouseId	Long(12)	Allow only items on the store order that are associated to this warehouse.
restrictToHierarchyId	Long(12)	Allow only items on the store order that belong to this merchandise hierarchy.
restrictToAreaId	Long(12)	Allow only items on the store order that are associated to this store sequence area.
createDate	Date	The date the record was created in EICS.
createUser	String(128)	The user that created the record in EICS.
externalCreateDate	Date	The date the store order was created in an external system.
updateDate	Date	The date the record was last updated in EICS.
approvedDate	Date	The date the record was approved in EICS.
approvedUser	String(128)	The user that approved the record in EICS.
lineItems	Collection	A collection of Store Order Line Items

Store Order Line Item

Column	Type	Definition
lineId	Long(12)	The unique identifier of the record.
itemId	String(25)	The unique identifier of the item.
approvedQuantity	BigDecimal(20,4)	The quantity of item ordered.
suggestedQuantity	BigDecimal(20,4)	The quantity of item ordered from the external system.
caseSize	BigDecimal(10,2)	The case size of the item ordered.
unitCostCurrency	String(3)	The currency of the unit cost.
unitCostAmount	BigDecimal(12,4)	The value of the unit cost.
deliverySlotId	Long(14)	The unique identifier of the delivery time.

API Find Orders

API is used to find lightweight transaction headers for store orders.

API Basics

Endpoint URL	{base URL}
Method	GET
Successful Response	200 OK
Processing Type	Synchronous
Input	Query Parameters
Output	List of store order headers
Max Response Limit	5,000

Query Parameters

Attribute	Type	Definition
storeId	Long(12)	Include only records for this store.
externalReference	String(128)	Include only records with this external transfer identifier.
externalTransferId	String(128)	Include only records with this external transfer identifier.
externalPurchaseOrderId	String(128)	Include only records with this external transfer identifier.
status	Integer	Include only records with this status. See StockOrderCriteriaStatus Index.
itemId	String(25)	Include only records with this item on them.
updateDateFrom	Date	Include records with a last update date on or after this date.
updateDateTo	Date	Include records with a first update date on or before this date.

Output Data Definition

Column	Type	Definition
storeOrderId	Long(12)	The unique identifier of the record.
storeId	Long(10)	The unique identifier of the store.
externalTransferId	String(128)	An external transfer identifier that this store order is associated to.
externalPurchaseOrderId	String(128)	An external purchase order identifier that this store order is associated to.
externalReference	String(128)	A reference to the store order in an external system.
description	String(2000)	A description of the store order or cause of the store order.
status	Integer(2)	The status: See Index
requestedDeliveryDate	Date	The date that the store requests the delivery arrive by.
updateDate	Date	The date the record was last updated in EICS.

API: Approve Order

This operation will approve the store order and send notifications and updates of the approval to an external system.

API Basics

Endpoint URL	{base URL}/{storeOrderId}/approve
Method	POST
Successful Response	204 No Content
Processing Type	Synchronous
Input	N/A
Output	The store order
Max Response Limit	N/A

API: Cancel Order

API Basics

Endpoint URL	{base URL}/{storeOrderId}/cancel
Method	POST
Successful Response	204 No Content
Processing Type	Synchronous
Input	N/A
Output	The store order
Max Response Limit	N/A

Additional Data Definitions

Store Order Status

ID	Status
1	New
2	In Progress
3	Approved
4	Canceled
5	Submitted

Store Order Criteria Status

ID	Status
1	New
2	In Progress
3	Approved
4	Canceled
5	Submitted
99	Active

Store Order Origin

ID	Status
1	External
2	Manual
3	System

Rest Service: Store Sequencing

This service defines operations to manage Store Sequence information.

Service Base URL

The Cloud service base URL follows the format:

https://<external_load_balancer>/<cust_env>/siocs-int-services/api/sequences

APIs

API	Description
Find Sequence Areas	Finds sequence area header based on search criteria.
Read Sequence Area	Reads the details of a sequence area.
Create Sequence Area	Create a sequence area along with all its details.

Update Sequence Area	Replace the entire sequence area including all its details (a complete sequence area must be sent).
Delete Sequence Area	Delete a sequenced area and all its details.
Create Sequence Item	Create a single sequence item in a sequence area.
Update Sequence Item	Updates a single sequence item from a sequence area.
Delete Sequence Item	Removes a single sequence item from a sequence area.

API: Find Sequence Areas

This API is used to search for sequence area headers. No items or detailed information is returned by this API. At least one criteria is required in order to search.

API Basics

Table 7-1 API Basics

Endpoint URL	{base URL}
Method	GET
Successful Response	200 OK
Processing Type	Synchronous
Input	Query Parameters
Output	Collection of Sequence Areas
Max Response Limit	15,000

Query Parameters

Table 7-2 Query Parameters

Attribute	Type	Definition
storeId	Long(12)	Include only records for this store.
areaType	Integer(9)	Area Type (see Index).
departmentId	Long(12)	The unique identifier of a department associated to the area.
classId	Long(12)	The unique identifier of a class within the department associated to the area.
itemId	String(25)	Include only records with this item on them.

Output Data Definition

Table 7-3 Output Data Definition

Attribute	Type	Definition
sequenceAreaId	Long(10)	The unique identifier of the sequence area.
storeId	Long(10)	The unique identifier of the store.
description	String(255)	A description of this store sequence.

Table 7-3 (Cont.) Output Data Definition

areaType	Integer(9)	Area Type (see Index).
departmentId	Long(12)	The unique identifier of a department associated to the area.
classId	Long(12)	The unique identifier of a class within the department associated to the area.
sequenceOrder	Long(20	The order of this store sequence compared to other store sequences at the store.
unsequenced	Boolean	True indicates that this is a default sequence area that contains all the items that have not been sequenced within a different area.

API: Read Sequence Area

This API is used to read an entire sequence area including its details.

API Basics

Table 7-4 API Basics

Endpoint URL	{base URL}/{sequenceAreaId}
Method	GET
Successful Response	200 OK
Processing Type	Synchronous
Input	N/A
Output	Sequence Area
Max Response Limit	N/A

Path Parameter Definition

Table 7-5 Path Parameter Definition

Attribute	Type	Definition
sequenceAreaId	Long(10)	The unique identifier of the sequence area.

Output Data Definition

Table 7-6 Output Data Definition

Attribute	Type	Definition
sequenceAreaId	Long(10)	The unique identifier of the sequence area.
storeId	Long(10)	The unique identifier of the store.
description	String(255)	A description of this store sequence.
areaType	Integer(9)	Area Type (see Index).
departmentId	Long(12)	The unique identifier of a department associated to the area.

Table 7-6 (Cont.) Output Data Definition

classId	Long(12)	The unique identifier of a class within the department associated to the area.
sequenceOrder	Long(20	The order of this store sequence compared to other store sequences at the store.
unsequenced	Boolean	True indicates that this is a default sequence area that contains all the items that have not been sequenced within a different area.
Items	Collection	Contains all the items in the area (see Store Sequence Item)

Store Sequence Item

Table 7-7 Store Sequence Item

Payload	Type	Definition
sequenceItemId	Long(12)	The unique area item identifier.
itemId	String(25)	The unique identifier of the item
sequenceOrder	Long(20)	The order of the item within its sequence.
capacity	BigDecimal(11,2)	The amount of the item that can be contained in the area for that item at the unit of measure.
width	Long(12)	The number of items that can fit across the width of the shelf.
uomMode	Integer(2)	The mode of display of the unit of measure of the item.
primaryLocation	Boolean	Y indicates this is the primary sequence for the item.
ticketQuantity	BigDecimal(11,2)	The quantity of tickets that need to be printed for the item inventory area.
ticketFormatId	Long(10)	The unique identifier of the ticket format used to print the tickets.

API: Create Sequence Area

API Basics

Endpoint URL	{base URL}
Method	POST
Successful Response	200 OK
Processing Type	Synchronous
Input	Sequence Area Create Info
Output	Sequence Area Reference Info
Max Response Limit	1,000 items in the area

Input Data Definition

Attribute	Type	Req	Definition
storeId	Long(10)	X	The unique identifier of the store.
description	String(255)		A description of this store sequence.
areaType	Integer(9)	X	Area Type (see Index).
departmentId	Long(12)		The unique identifier of a department associated to the area.
classId	Long(12)		The unique identifier of a class within the department associated to the area.
sequenceOrder	Long(20)	X	The order of this store sequence compared to other store sequences at the store. The sequence order must be unique for the set of areas within the store.
items	Collections	X	All the items that belong to this sequence area (see Sequence Area Item Create)

Store Sequence Area Item Create

Payload	Type	Req	Definition
itemId	String(25)	X	The unique identifier of the item
sequenceOrder	Long(20)	X	The order of the item within its sequence.
capacity	BigDecimal(11,2)	X	The amount of the item that can be contained in the area for that item at the unit of measure.
width	Long(12)		The number of items that can fit across the width of the shelf.
uomMode	Integer(2)	X	The mode of display of the unit of measure of the item.
primaryLocation	Boolean		Y indicates this is the primary sequence for the item.
ticketQuantity	BigDecimal(11,2)		The quantity of tickets that need to be printed for the item inventory area.
ticketFormatId	Long(10)		The unique identifier of the ticket format used to print the tickets.

Output Data Definition

Attribute	Type	Definition
sequenceAreaId	Long(12)	The newly created adjustment identifier.
storeId	Long(10)	The store identifier.

Example

```
{
   "storeId": 5000,
   "areaType": 2,
   "description": "RestTest22",
   "sequenceOrder": 7,
   "items": [
      {
         "itemId": "100637121",
         "capacity": 2,
         "sequenceOrder": 1,
         "uomMode": 1
      }
   ]
}
```


API: Update Sequence Area

This API is used to update a sequence area within a store.

This API will **replace** the entire sequence area with the new data passed to the API. Only the area identifier and store identifier will be preserved from previous data.

API Basics

Table 7-8 API Basics

Endpoint URL	{base URL}
Method	POST
Successful Response	204 No Content
Processing Type	Synchronous
Input	Sequence Area Update Info
Output	N/A
Max Response Limit	1,000 items in the area

Path Parameter Definition

Table 7-9 Path Parameter Definition

Attribute	Type	Definition
sequenceItemId	Long(12)	The unique area item identifier

Input Data Definition

Table 7-10 Input Data Definition

Attribute	Type	Req	Definition

Table 7-10 (Cont.) Input Data Definition

description	String(255)		A description of this store sequence.
areaType	Integer(9)	X	Area Type (see Index).
departmentId	Long(12)		The unique identifier of a department associated to the area.
classId	Long(12)		The unique identifier of a class within the department associated to the area.
sequenceOrder	Long(20)	X	The order of this store sequence compared to other store sequences at the store. The sequence order must be unique for the set of areas within the store.
items	Collections	X	All the items that belong to this sequence area (see Sequence Area Item Update)

Store Sequence Area Item Update

Table 7-11 Store Sequence Area Item Update

Payload	Type	Req	Definition
itemId	String(25)	X	The unique identifier of the item
sequenceOrder	Long(20)	X	The order of the item within its sequence.
capacity	BigDecimal(11,2)	X	The amount of the item that can be contained in the area for that item at the unit of measure.
width	Long(12)		The number of items that can fit across the width of the shelf.
uomMode	Integer(2)	X	The mode of display of the unit of measure of the item.
primaryLocation	Boolean		Y indicates this is the primary sequence for the item.
ticketQuantity	BigDecimal(11,2)		The quantity of tickets that need to be printed for the item inventory area.
ticketFormatId	Long(10)		The unique identifier of the ticket format used to print the tickets.

Example

```
{
  "areaType": 2,
  "description": "RestTest22",
  "sequenceOrder": 7,
  "items": [
    {
      "itemId": "100637121",
      "capacity": 2,
      "sequenceOrder": 1,
      "uomMode": 1
    }
  ]
}
```


API: Delete Sequence Area

This API is used to delete a sequence area and all its items. This will not delete a sequence area currently being used by a sequenced stock count.

API Basics

Table 7-12 API Basics

Endpoint URL	{base URL}
Method	DELETE
Successful Response	204 No Content
Processing Type	Synchronous
Input	N/A
Output	N/A
Max Response Limit	N/A

API: Create Sequence Item

This API is used to create a new sequence item within a sequence area.

API Basics

Table 7-13 API Basics

Endpoint URL	{base URL}/{sequenceAreaId}/items
Method	POST
Successful Response	200 No Content
Processing Type	Synchronous
Input	Store Sequence Item Create

Table 7-13 (Cont.) API Basics

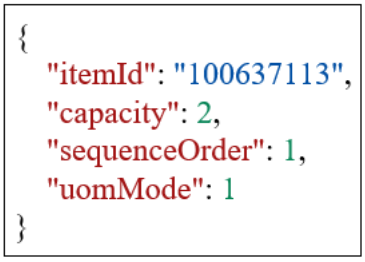
Output	Store Sequence Item
Max Response Limit	N/A

Input Data Definition

Table 7-14 Input Data Definition

Payload	Type	Req	Definition
itemId	String(25)	X	The unique identifier of the item
sequenceOrder	Long(20)	X	The order of the item within its sequence.
capacity	BigDecimal(11,2)	X	The amount of the item that can be contained in the area for that item at the unit of measure.
width	Long(12)		The number of items that can fit across the width of the shelf.
uomMode	Integer(2)	X	The mode of display of the unit of measure of the item. (see Index)
primaryLocation	Boolean		Y indicates this is the primary sequence for the item.
ticketQuantity	BigDecimal(11,2)		The quantity of tickets that need to be printed for the item inventory area.
ticketFormatId	Long(10)		The unique identifier of the ticket format used to print the tickets.

Example



API: Update Sequence Item

This API is used to update a sequence item within a sequence area.

API Basics

Table 7-15 API Basics

Endpoint URL	{base URL}/{sequenceAreaId}/items/{itemId}
Method	POST

Table 7-15 (Cont.) API Basics

Successful Response	204 No Content
Processing Type	Synchronous
Input	N/A
Output	N/A
Max Response Limit	N/A

Path Parameter Definition

Table 7-16 Path Parameter Definition

Attribute	Type	Definition
sequencerAreaId	Long(12)	The unique sequence area identifier.
itemId	String(25)	The item to be deleted from the sequence.

Input Data Definition

Table 7-17 Input Data Definition

Payload	Type	Req	Definition
sequenceOrder	Long(20)	X	The order of the item within its sequence.
capacity	BigDecimal(11,2)	X	The amount of the item that can be contained in the area for that item at the unit of measure.
width	Long(12)		The number of items that can fit across the width of the shelf.
uomMode	Integer(2)	X	The mode of display of the unit of measure of the item.
primaryLocation	Boolean		Y indicates this is the primary sequence for the item.
ticketQuantity	BigDecimal(11,2)		The quantity of tickets that need to be printed for the item inventory area.
ticketFormatId	Long(10)		The unique identifier of the ticket format used to print the tickets.

Example

```
{
  "capacity": 8,
  "sequenceOrder": 2,
  "uomMode": 1
}
```

API: Delete Sequence Item

This API is used to delete a sequence item from an area.

API Basics

Table 7-18 API Basics

Endpoint URL	{base URL}/{sequenceAreaId}/items/{itemId}
Method	DELETE
Successful Response	204 No Content
Processing Type	Synchronous
Input	N/A
Output	N/A
Max Response Limit	N/A

Path Parameter Definition

Table 7-19 Path Parameter Definition

Attribute	Type	Definition
sequencerAreaId	Long(12)	The unique sequence area identifier.
itemId	String(25)	The item to be deleted from the sequence.

Additional Data Definitions

Sequence Area Type

Table 7-20 Sequence Area Type

ID	Status
1	Shopfloor
2	Backroom
3	None

Sequence UOM Mode

Table 7-21 Sequence UOM Mode

ID	Status
1	Units
2	Cases

REST Service Supplier

This service integrates supplier and supplier item foundation data.

Asynchronous supplier integration is processed through staged messages and is controlled by the MPS Work Type: DcsSupplier.

Asynchronous supplier item integration is processed through staged messages and is controlled by the MPS Work Type: DcsSupplierItem.

Service Base URL

The Cloud service base URL follows the format:

https://<external_load_balancer>/<cust_env>/siocs-int-services/api/suppliers

API Definitions

API	Description
Import Suppliers	Imports supplier.
Delete Supplier	Deletes a supplier.
Import Items	Imports items for a supplier.
Delete Items	Deletes items from a supplier.
Import Item UOMs	Imports units of measure for a supplier item.
Delete Item UOMs	Deletes units of measure from a supplier item.
Import Item Countries	Imports countries for a supplier item.
Delete Item Countries	Deletes countries from a supplier item.
Import Item Dimensions	Imports items dimensions for a supplier item country.
Delete Item Dimensions	Deletes dimensions from a supplier item country.
Import Item Manufacturers	Imports manufacturers for a supplier item country.
Delete Item Manufacturers	Deletes manufacturers from a supplier item country.

API: Import Suppliers

Imports a list of suppliers.

If the import contains more than 500 suppliers, an input too large error will be returned.

A "Forbidden" response will occur if application is integrated with MFCS or RMS.

API Basics

Endpoint URL	{base URL}/import
Method	POST
Successful Response	202 Accepted
Processing Type	Asynchronous
Input	List of suppliers to Import
Output	None
Max Response Limit	500

Input Data Definition

Attribute	Data Type	Required	Description

suppliers	List of details	Yes	A collection of up to 500 suppliers to import.

Detail Data Definition

Attribute	Data Type	Required	Description
supplierId	Long (10)	Yes	The supplier identifier.
name	String (240)	-	The supplier's name.
status	Integer	Yes	The supplier’s status: See Supplier Status
dunsNumber	String (9)	-	The nine-digit number assigned and maintained by Dun and Bradstreet.
taxId	String (18)	-	The tax identification number of the supplier.
parentId	Long (128)	-	The parent supplier's identifier.
countryCode	String (3)	-	The 2-3 character ISO code of the country.
languageCode	String (3)	-	The 2-3 character ISO code of the language.
currencyCode	String (3)	-	The 2-3 character ISO code of the currency.
returnAllowed	Boolean	-	True is return is allowed to the supplier, N otherwise.
authorizationRequired	Boolean	-	True indicates an authorization number is required when merchandise is return to this supplier.
orderCreateAllowed	Boolean	-	True indicates at a purchase order can be created for the supplier when processing deliveries from that supplier.
vendorCheck	Boolean	-	True indicates that orders from this supplier requires vendor control.
vendorCheckPercent	BigDecimal	-	Indicates the percentage of items per receipt that will be marked for vendor checking.
quantityLevel	Integer	Yes	The Quantity Level: See Quantity Level
deliveryDiscrepancy	Integer	Yes	The delivery discrepancy: See Supplier Delivery Discrepancy Type
organizationUnitIds	List<Long>	Yes	A complete list of organization unit identifiers for the supplier.

Example Input

{

"suppliers": [

{

"supplierId": 5000,

"status": 1,

"quantityLevel": 1,

"deliveryDiscrepancy": 1,

"organizationUnitIds": [1]

},

{

"supplierId": 5001,

"status": 1,

"returnAllowed": false,

"quantityLevel": 1,

"deliveryDiscrepancy": 1,

"organizationUnitIds": [1]

}

]

}

Additional Data Definitions

Supplier Delivery Discrepancy Type

Value	Definition
1	Allow Any Discrepancy
2	Allow Overage But Not Short Receipts
3	Discrepancy Not Allowed

Supplier Status

Value	Definition
1	Active
2	Inactive

Quantity Level

Value	Definition
1	Eaches
2	Cases

API: Delete Supplier

Deletes a supplier.

A "Forbidden" response will occur if application is integrated with MFCS or RMS.

API Basics

Endpoint URL	{base URL}/{supplierId}/delete
Method	POST

Successful Response	202 Accepted
Processing Type	Asynchronous
Input	None
Output	None
Max Response Limit	1000

Path Parameter Definitions

Attribute	Definition
supplierId	The internal identifier of the supplier.

API: Import Items

Imports supplier items. Later during asynchronous processing, it validates that both the supplier and item exist before inserting new records.

If the import contains more than 500 supplier items, an input too large error will be returned.

A "Forbidden" response will occur if application is integrated with MFCS or RMS.

API Basics

Endpoint URL	{base URL}/items
Method	POST
Successful Response	202 Accepted
Processing Type	Asynchronous
Input	List of Supplier Items to import
Output	None
Max Response Limit	500

Path Parameter Definitions

Attribute	Definition
supplierId	The internal identifier of the supplier.

Input Data Definition

Attribute	Data Type	Required	Description
items	List of details	Yes	The supplier item information to import.

Detail Data Definition

Attribute	Data Type	Required	Description

supplierId	Long (10)	Yes	The supplier identifier.
itemId	String (25)	Yes	The item identifier.
vendorProductNumber	String (256)	-	Vendor product number.
primary	Boolean	-	True indicates this supplier is the primary supplier for the item at all locations.

Example Input

{
"items": [
{
"supplierId": 5000,
"itemId": "163715121",
"vendorProductNumber": "abc"
},
{
"supplierId": 5001,
"itemId": "163715121",
"vendorProductNumber": "def"
}
]
}

API: Delete Items

Deletes supplier items.

If the delete contains more than 500 supplier items, an input too large error will be returned.

A "Forbidden" response will occur if application is integrated with MFCS or RMS.

API Basics

Endpoint URL	{base URL}/items/delete
Method	POST
Successful Response	202 Accepted
Processing Type	Asynchronous
Input	List of supplier items
Output	None
Max Response Limit	500

Input Data Definition

Attribute	Data Type	Required	Description
items	List of details	Yes	The supplier item information to delete.

Detail Data Definition

Attribute	Data Type	Required	Description
supplierId	Long (10)	Yes	The supplier identifier.
itemId	String (15)	Yes	The item identifier.

Example Input

{

"items":

[

{

"supplierId":"9002",

"itemId": "100637130"

},

{

"supplierId":"9002",

"itemId": "100637148"

}

]

}

API: Import Item UOMs

Imports supplier item unit of measure information.

If the import contains more than 500 supplier item units of measure, an input too large error will be returned.

A "Forbidden" response will occur if application is integrated with MFCS or RMS.

API Basics

Endpoint URL	{base URL}/uoms
Method	POST
Successful Response	202 Accepted
Processing Type	Asynchronous
Input	List of supplier items UOMs to import

Output	None
Max Response Limit	500

Input Data Definition

Attribute	Data Type	Required	Description
uoms	List of details	Yes	The UOMs to import.

Detail Data Definition

Attribute	Data Type	Required	Description
supplierId	Long (10)	Yes	The supplier identifier.
itemId	String (25)	Yes	The item identifier.
unitOfMeasure	String (4)	Yes	The unit of measure.
Value	BigDecimal	Yes	Equivalent item/supplier shipping carton value in unit of measure

Example Input

{
"uoms": [
{
"supplierId": 5000,
"itemId": "163715121",
"unitOfMeasure": "KG",
"value": "1.0"
},
{
"supplierId": 5001,
"itemId": "163715121",
"unitOfMeasure": "KG",
"value": "1.0"
}
]
}

API: Delete Item UOMs

Deletes supplier item unit of measure information.

If the delete contains more than 500 supplier item units of measure, an input too large error will be returned.

A "Forbidden" response will occur if application is integrated with MFCS or RMS.

API Basics

Endpoint URL	{base URL}/uoms/delete
Method	POST
Successful Response	202 Accepted
Processing Type	Asynchronous
Input	List of supplier items UOMs to delete
Output	None
Max Response Limit	500

Input Data Definition

Attribute	Data Type	Required	Description
uoms	List of details	Yes	The UOMs to delete.

Detail Data Definition

Attribute	Data Type	Required	Description
supplierId	Long (10)	Yes	The supplier identifier.
itemId	String (25)	Yes	The item identifier.
unitOfMeasure	String (4)	Yes	The unit of measure.

Example Input

```
{
"uoms":
[
{
"supplierId":"9002",
"itemId": "100637130",
"unitOfMeasure":"EA"
},
{
"supplierId":"9002",
"itemId": "100637148",
"unitOfMeasure":"EA"
}
]
```

API: Import Item Countries

Imports supplier item country information. If the imported supplier item country is for the primary supplier of the item, the item's case size will be updated on the item master to reflect the imported case size.

If the import contains more than 500 supplier item countries, an input too large error will be returned.

A "Forbidden" response will occur if application is integrated with MFCS or RMS.

API Basics

Endpoint URL	{base URL}/countries
Method	POST
Successful Response	202 Accepted
Processing Type	Asynchronous
Input	List of supplier items country to import
Output	None
Max Response Limit	500

Input Data Definition

Attribute	Data Type	Required	Description
countries	List of details	Yes	The countries to import.

Detail Data Definition

Attribute	Data Type	Required	Description
supplierId	Long (10)	Yes	The supplier identifier.
itemId	String (25)	Yes	The item identifier.
countryCode	String (3)	Yes	The ISO country code of the supplier that produces the item.
caseSize	BigDecimal	Yes	Number of items with a case from the supplier.
unitCostCurrency	String (3)	-	The unit cost currency for that item and supplier in that country.
unitCostValue	BigDecimal	-	The unit cost of the item and supplier in that country.

Example Input

```
{
   "countries": [
      {
         "supplierId": 5100,
         "itemId": "163715121",
```

```
        "countryCode": "US",
        "caseSize":7
      },
      {
        "supplierId": 5200,
        "itemId": "163715121",
        "countryCode": "US",
        "caseSize":8
      }
   ]
}
```


API: Delete Item Countries

Deletes supplier item country information.

If the delete contains more than 500 supplier item countries, an input too large error will be returned.

A "Forbidden" response will occur if application is integrated with MFCS or RMS.

API Basics

Endpoint URL	{base URL}/countries/delete
Method	POST
Successful Response	202 Accepted
Processing Type	Asynchronous
Input	List of supplier items country to remove
Output	None
Max Response Limit	500

Input Data Definition

Attribute	Data Type	Required	Description
countries	List of details	Yes	The countries to remove.

Detail Data Definition

Attribute	Data Type	Required	Description
supplierId	Long (10)	Yes	The supplier identifier.
itemId	String (25)	Yes	The item identifier.
countryCode	String (3)	Yes	The ISO country code of the supplier that produces the item.

Example Input

{

"countries":

[

{

"supplierId":"9002",

"itemId": "100637130",

"countryCode":"US"

}

]

}

API: Import Item Dimensions

Imports supplier item country dimension information.

If the import contains more than 500 supplier item dimensions, an input too large error will be returned.

A "Forbidden" response will occur if application is integrated with MFCS or RMS.

API Basics

Endpoint URL	{base URL}/dimensions
Method	POST
Successful Response	202 Accepted
Processing Type	Asynchronous
Input	List of supplier items dimensions to import
Output	None
Max Response Limit	500

Input Data Definition

Attribute	Data Type	Required	Description
dimensions	List of details	Yes	The dimensions to import.

Detail Data Definition

Attribute	Data Type	Required	Description
supplierId	Long (10)	Yes	The supplier identifier.
itemId	String (25)	Yes	The item identifier.
countryCode	String (3)	Yes	The ISO country code of the supplier that produces the item.

dimensionName	String (6)	Yes	Dimension name
presentationMethod	String (6)	-	Describes the packaging method
Length	BigDecimal	-	The length in dimension unit of measure.
Width	BigDecimal	-	The width in dimension unit of measure.
Height	BigDecimal	-	The height in dimension unit of measure.
dimensionUom	String (4)	-	The unit of measure of the dimensions.
Weight	BigDecimal	-	The weight of the object in weight unit of measure.
netWeight	BigDecimal	-	The net weight of the goods without packaging in weight unit of measure.
weightUom	String (4)	-	The weight unit of measure.
liquidVolume	BigDecimal	-	The liquid value or capacity of the object.
liquidVolumeUom	String (4)	-	The liquid volume unit of measure.
statisticalCube	BigDecimal	-	A statistical value of the object's dimensions used for shipment loading purposes.

Example Input

{

"dimensions": [

{

"supplierId": 5100,

"itemId": "163715121",

"countryCode": "US",

"dimensionName": "Hello"

},

{

"supplierId": 7100,

"itemId": "163715121",

"countryCode": "US",

"dimensionName": "Bye"

}

]

}

API: Delete Item Dimensions

Deletes supplier item country dimension information.

If the delete contains more than 500 supplier items, an input too large error will be returned.

A "Forbidden" response will occur if application is integrated with MFCS or RMS.

API Basics

Endpoint URL	{base URL}/dimensions/delete
Method	POST
Successful Response	202 Accepted
Processing Type	Asynchronous
Input	List of supplier items dimensions to remove
Output	None
Max Response Limit	500

Input Data Definition

Attribute	Data Type	Required	Description
Dimensions	List of details	Yes	The dimensions to remove.

Supplier Items Dimensions Data Definition

Attribute	Data Type	Required	Description
supplierId	Long (10)	Yes	The supplier identifier.
itemId	String (25)	Yes	The item identifier.
countryCode	String (3)	Yes	The ISO country code of the supplier that produces the item
dimensionName	String (6)	Yes	Dimension name

Example Input

{

"dimensions":

[

{

"supplierId":"9002",

"itemId": "100637130",

"countryCode":"US",

"dimensionName":"Dimen1"

},

{

"supplierId":"9002",

"itemId": "100637148",

"countryCode":"US",

"dimensionName":"Dimen2"

}

]

}

API: Import Item Manufacturers

Import supplier item country manufacturer information.

If the import contains more than 500 supplier item manufacturers, an input too large error will be returned.

A "Forbidden" response will occur if application is integrated with MFCS or RMS.

API Basics

Endpoint URL	{base URL}/manufacturers
Method	POST
Successful Response	202 Accepted
Processing Type	Asynchronous
Input	List of supplier items manufacturer to import
Output	None
Max Response Limit	500

Input Data Definition

Attribute	Data Type	Required	Description
manufacturers	List of details	Yes	The manufacturers to import.

Detail Data Definition

Attribute	Data Type	Required	Description
supplierId	Long (10)	Yes	The supplier identifier.
itemId	String (25)	Yes	The item identifier.
countryCode	String (3)	Yes	The ISO country code of the supplier that produces the item.
primary	Boolean	-	True indicates it is primary country of manufacture.

Example Input

{

"manufacturers": [

{

"supplierId": 5100,

"itemId": "163715121",

"countryCode": "US",

"primary": true

},

{

"supplierId": 7100,

"itemId": "163715121",

"countryCode": "US",

"primary": true

}

]

}

API: Delete Item Manufacturers

Deletes supplier item country manufacturer information.

If the delete contains more than 500 supplier item manufacturers, an input too large error will be returned.

A "Forbidden" response will occur if application is integrated with MFCS or RMS.

API Basics

Endpoint URL	{base URL}/manufacturers/delete
Method	POST
Successful Response	202 Accepted
Processing Type	Asynchronous
Input	List of supplier items manufacturers to remove
Output	None
Max Response Limit	500

Input Data Definition

Attribute	Data Type	Required	Description
manufacturers	List of details	Yes	The manufacturers to remove.

Detail Data Definition

Attribute	Data Type	Required	Description
supplierId	Long (10)	Yes	The supplier identifier.
itemId	String (25)	Yes	The item identifier.
countryCode	String (3)	Yes	The ISO country code of the supplier that produces the item.

Example Input

{
"manufacturers":
[
{
"supplierId":"9002",
"itemId": "100637130",
"countryCode":"US"
},
{
"supplierId":"9002",
"itemId": "100637148",
"countryCode":"US"
}
]
}

REST Service: Ticket

Service Base URL

The Cloud service base URL follows the format:

https://<external_load_balancer>/<cust_env>/siocs-int-services/api/tickets

API Definitions

API	Description
readTicket	This API reads a current actively ticket by its identifier.
findTickets	API is used to find summarized ticket headers for a set of criteria.
readArchivedTicket	This API reads a previously printed and archived ticket by its identifier.
findArchivedTickets	API is used to find archived ticket summaries.
createTickets	Creates new tickets within the system.
updateTickets	Updates current tickets within the system.
printTickets	Prints the requested tickets.
findTicketFormats	Reads all the available ticket formats.
findTicketPrinters	Finds the printers available to print tickets.

API: readTicket

This API reads a current actively ticket by its identifier.

API Basics

Endpoint URL	{base URL}/{ticketId}
Method	GET
Success Response	200 OK
Processing Type	Synchronous
Input	The ticket identifier
Output	The ticket

Output Data Definition

Column	Type	Definition
ticketId	Long(12)	The unique ticket identifier.
itemId	String(25)	The item identifier.
storeId	Long(10)	The store identifier.
originType	Integer(2)	The origin type (See Index)
ticketFormatId	Long(12)	The identifier of the ticket format used for printing.
ticketFormatDescription	String(100)	The description of the ticket format associated to the ticket.
ticketFormatType	Integer(4)	The type of the ticket format associated to the ticket.
ticketFormatReference	String(255)	The format reference used to print the ticket.
ticketFormatZplId	Long(12)	The ZPL format file identifier.
ticketCount	Integer(3)	The number of instances of this ticket to print.
ticketSequence	Integer(3)	The sequence number of a ticket within a ticket grouping.
printQuantity	BigDecimal(12,4)	The quantity to be printed on the ticket.
printDate	Date	The date the ticket should be printed.
groupId	Long(12)	An internal EICS grouping identifier that groups the tickets.
groupIdExternal	String(128)	An external system grouping identifier that groups the tickets.
autoPrint	Boolean	True if the ticket should automatically print, false if the ticket requires manual printing.
autoRefeshQuantity	Boolean	True indicates the ticket count gets refreshed with SOH at the time of printing, false it prints as is.
countryOfManufacture	String(3)	A two letter country code denoting the country of manufacture for the item.
shortDescription	String(255)	The short description of the item.
longDescription	String(400)	The long description of the item.
shortDescriptionLanguage	String(255)	The short description of the item in the language of the store.

longDescriptionLanguage	String(400)	The long description of the item in the language of the store.
differentiatorType1	String(10)	The description of the differentiator type for differentiator 1.
differentiatorType2	String(10)	The description of the differentiator type for differentiator 2.
differentiatorType3	String(10)	The description of the differentiator type for differentiator 3.
differentiatorType4	String(10)	The description of the differentiator type for differentiator 4.
differentiatorDescription1	String(255)	The description of differentiator 1 of the item.
differentiatorDescription2	String(255)	The description of differentiator 2 of the item.
differentiatorDescription3	String(255)	The description of differentiator 3 of the item.
differentiatorDescription4	String(255)	The description of differentiator 4 of the item.
departmentId	Long(12)	The department identifier of the item.
departmentName	String(360)	The department name of the item.
classId	Long(12)	The class identifier of the item.
className	String(360)	The class name of the item.
subclassId	Long(12)	The subclass identifier of the item.
subclassName	String(360)	The subclass name of the item.
primaryUpc	String(25)	The primary Unique Product Code for the item.
priceCurrency	String(3)	The currency code of the ticket price.
priceAmount	BigDecimal(12,4)	The amount of the ticket price.
priceType	Integer(3)	The type of the ticket price. (See Index)
priceUom	String(4)	The unit of measure of the ticket price.
priceActiveDate	Date	The date the ticket price became active.
priceExpireDate	Date	The date the ticket price expires.
mutliUnitPriceCurrency	String(3)	The currency code of the ticket's multi-unit price.
multiUnitPriceAmount	BigDecimal(12,4)	The amount of the ticket's multi-unit price.
multiUnitPriceUom	String(4)	The unit of measure of the ticket's multi-unit price.
multiUnitQuantity	BigDecimal(12,4)	The multi-unit quantity associated to the price.
overridePriceCurrency	String(3)	The override price currency code.
overridePriceAmount	BigDecimal(20,4)	The override price amount.
previousPriceCurrency	String(3)	The currency code of the previous price.
previousPriceAmount	BigDecimal(12,4)	The amount of the previous price.
previousPriceType	Integer(3)	The price type of the previous price.
lowestMonthlyPriceCurrency	String(3)	The currency code of the lowest monthly price.
lowestMonthlyPriceAmount	BigDecimal(12,4)	The amount of the lowest monthly price.

lowestMonthlyPriceType	Integer(3)	The price type of the lowest monthly price.
printedDate	Date	The date that the ticket was printed.
printedUser	String(128)	The user that printed the ticket.
createDate	Date	The date the ticket was created.
createUser	String(128)	The user that created the ticket.
updateDate	Date	The date the ticket was last updated.
updateUser	String(128)	The user that last updated the ticket.
udas	Collection	A group of associated user defined attributes.

UDA

Column	Type	Definition
name	String	The name of the user defined attribute.
value	String	The value of the user defined attributes.

API: findTickets

API is used to find summarized ticket headers for a set of criteria.

If maximum results are exceeded, additional or more limiting input criteria will be required.

API Basics

Endpoint URL	{base URL}
Method	GET
Success Response	200 OK
Processing Type	Synchronous
Input	Query Parameters
Output	List of Ticket Headers
Maximum Results Allowed	5,000

Input Data Definition

Attribute	Type	Definition
itemId	String(25)	Include only records with this item identifier.
storeId	Long(10)	Include only records with this store identifier.
ticketFormatId	Long(12)	Include only records with this ticket format identifier.
groupId	Long(12)	Include only records with this EICS group identifier.
groupIdExternal	String(128)	Include only records with this external system group identifier.
printDateFrom	String	Include only records on or after this scheduled print date and time.
printDateTo	String	Include only records on or before this scheduled print date and time.

updateDateFrom	String	Include only records on or after this scheduled print date and time.
updateDateTo	String	Include only records on or before this scheduled print date and time.

Output Data Definition

Column	Type	Definition
ticketId	Long(12)	The unique ticket identifier.
itemId	String(25)	The item identifier.
storeId	Long(10)	The store identifier.
originType	Integer(2)	The origin type (See Index)
ticketFormatId	Long(12)	The identifier of the ticket format used for printing.
ticketSequence	Integer(3)	The sequence number of a ticket within a ticket grouping.
groupId	Long(12)	An internal EICS grouping identifier that groups the tickets.
groupIdExternal	String(128)	An external system grouping identifier that groups the tickets.
autoPrint	Boolean	True if the ticket should automatically print, false if the ticket requires manual printing.
printDate	Date	The date the ticket should be printed.
printQuantity	BigDecimal(12, 4)	The quantity to be printed on the ticket.
updateDate	Date	The date the ticket was last updated.

API: readArchivedTicket

This API reads a previously printed and archived ticket by its identifier.

API Basics

Endpoint URL	{base URL}/archives/{ticketId}
Method	GET
Success Response	200 OK
Processing Type	Synchronous
Input	The ticket identifier
Output	The ticket details

Output Data Definition

See readTicket() for the output data definition of this API.

API: findArchivedTickets

API is used to find archived ticket summaries.

If the number of tickets found exceeds the limit, additional or more limiting input criteria will be required.

API Basics

Endpoint URL	{base URL}/archives
Method	GET
Success Response	200 OK
Processing Type	Synchronous
Input	Query Parameters
Output	List of Ticket Headers
Maximum Results Allowed	5,000

Input Data Definition

Attribute	Type	Definition
itemId	String(25)	Include only records with this item identifier.
storeId	Long(10)	Include only records with this store identifier.
ticketFormatId	Long(12)	Include only records with this ticket format identifier.
groupId	Long(12)	Include only records with this EICS group identifier.
groupIdExternal	String(128)	Include only records with this external system group identifier.
printedDateFrom	String	Include only records that were printed on or after this date and time.
printedDateTo	String	Include only records that were printed on or before this date and time.

Output Data Definition

Column	Type	Definition
ticketId	Long(12)	The unique ticket identifier.
itemId	String(25)	The item identifier.
storeId	Long(10)	The store identifier.
originType	Integer(2)	The origin type (See Index)
ticketFormatId	Long(12)	The identifier of the ticket format used for printing.
ticketSequence	Integer(3)	The sequence number of a ticket within a ticket grouping.
groupId	Long(12)	An internal EICS grouping identifier that groups the tickets.
groupIdExternal	String(128)	An external system grouping identifier that groups the tickets.
printedDate	Date	The date the ticket was printed.
printQuantity	BigDecimal(12,4)	The quantity printed on the ticket.
priceCurrency	String(3)	The currency of the price of the ticket.
priceAmount	BigDecimal(12,4)	The amount of the price of the ticket.

API: createTickets

Creates new tickets within the system.

API Basics

Endpoint URL	{base URL}
Method	POST
Success Response	200 OK
Processing Type	Synchronous
Input	Ticket group
Output	List of ticket identifying information
Maximum Input Allowed	2,000 tickets within the group

Input Data Definition

Payload	Type	Req	Definition
storeId	Long(10)	X	The store identifier.
groupIdExternal	String(128)		An external system grouping identifier that groups the tickets.
printDate	Date	X	The date the ticket should be printed.
autoPrint	Boolean		True if the ticket should automatically print, false if the ticket requires manual printing. Defaults to false.
autoRefeshQuantity	Boolean		True indicates the ticket count gets refreshed with SOH at the time of printing, false it prints as is. Defaults to false.
tickets	Collection	X	The tickets to create.

Ticket

Payload	Type	Req	Definition
itemId	String(25)	X	The item identifier.
originType	Integer(2)	X	The origin type (See Index)
ticketFormatId	Long(12)	X	The identifier of the ticket format used for printing.
ticketCount	Integer(3)	X	The number of instances of this ticket to print.
ticketSequence	Integer(3)	X	The sequence number of a ticket within a ticket grouping.
printQuantity	BigDecimal(12, 4)	X	The quantity to be printed on the ticket.
overridePriceCurrency	String(3)		The override price currency code. Required if an amount is entered.
overridePriceAmount	BigDecimal(20, 4)		The override price amount.

countryOfManufacture	String(3)	A two letter country code denoting the country of manufacture for the item.

Output Data Definition

Payload	Type	Definition
ticketId	Long(12)	The unique ticket identifier.
storeId	Long(10)	The store identifier.
ticketFormatId	Long(12)	The identifier of the ticket format used for printing.

Example Input


```
{
   "storeId": 5000,
   "groupIdExternal": "123456",
   "printDate": "2023-01-10T23:59:59-05:00",
   "autoPrint": false,
   "autoRefreshQuantity": false,
   "tickets": [
      {
         "itemId": "100637121",
         "originType": 1,
         "ticketFormatId": 1,
         "ticketCount": 2,
         "ticketSequence": 3
      },
      {
         "itemId": "100637113",
         "originType": 2,
         "ticketFormatId": 2,
         "ticketCount": 4,
         "ticketSequence": 5
      }
   ]
}
```


API: Update Tickets

Updates current tickets within the system.

If a field that is not required contains an empty or null value, the ticket will be updated to that empty or null value.

API Basics

Endpoint URL	{base URL}/update
Method	POST
Success Response	204 No Content
Processing Type	Synchronous
Processing Type	Synchronous
Input	Tickets
Output	N/A
Maximum Input Allowed	2,000 tickets

Input Data Definition

Payload	Type	Req	Definition
ticketId	Long(12)	X	The unique ticket identifier.
originType	Integer(2)	X	The origin type (See Index)
ticketCount	Integer(3)	X	The number of instances of this ticket to print.
ticketSequence	Integer(3)	X	The sequence number of a ticket within a ticket grouping.
printQuantity	BigDecimal(12,4)	X	The quantity to be printed on the ticket.
printDate	Date	X	The date the ticket should be printed.
autoPrint	Boolean		True if the ticket should automatically print, false if the ticket requires manual printing.
autoRefeshQuantity	Boolean		True indicates the ticket count gets refreshed with SOH at the time of printing, false it prints as is.
overridePriceCurrency	String(3)		The override price currency code. Required if an amount is entered.
overridePriceAmount	BigDecimal(20,4)		The override price amount.
countryOfManufacture	String(3)		A two letter country code denoting the country of manufacture for the item.

Example Input

```
{
   "tickets": [
      {
         "ticketId": 8,
         "originType": 2,
         "printDate": "2023-01-10T23:59:59-05:00",
```

```
            "ticketCount": 22,
            "ticketSequence": 33
        },
        {
            "ticketId": 9,
            "originType": 1,
            "printDate": "2023-01-10T23:59:59-05:00",
            "ticketCount": 44,
            "ticketSequence": 55
        }
    ]
}
```


API: printTickets

Prints the requested tickets. This can print both current tickets and previously printed tickets.

It is assumed that the calling system will have verified or retrieved the ticket ids prior. The ticket ids and archived ticket ids will not be validated.

This service operation will simply print any tickets that are found that match identifiers passed in and ignore any identifiers for which tickets are not found.

Depending on printing configuration with the system, the printing may occur synchronous and real-time or the tickets may be staged and sent asynchronously to another system after a short delay.

API Basics

Endpoint URL	{base URL}/print
Method	POST
Success Response	204 No Content
Processing Type	Synchronous/Asynchronous
Input	Ticket identifiers
Output	N/A
Maximum Input Allowed	100 total ticket ids

Input Data Definition

Payload	Type	Req	Definition
printerId	Long	X	The identifier of the printer to print the tickets on.
refreshQuantity	Boolean		If true, the quantities of all the tickets will be refreshed prior to printing. If false, they will not.

ticketIds	List<Long>	A list of ticket identifiers of the tickets to print. If this is null or empty, then archived ticket ids must contain a value. Not validated.
archivedTicketIds	List<Long>	A list of archived ticket identifiers of tickets to print. If this is null or empty, then ticket ids must contain a value. Not validated.

API: findTicketFormats

Reads all the available ticket formats.

API Basics

Endpoint URL	{base URL}/formats
Method	GET
Success Response	200 OK
Processing Type	Synchronous
Input	Query parameters
Output	List of ticket formats

Input Data Definition

Column	Type	Definition
storeId	Long(10)	Include only ticket formats available at this store.

Output Data Definition

Column	Type	Definition
formatId	Long(12)	The unique format identifier.
storeId	Long(10)	The store identifier.
description	String(100)	A description of the ticket (possibly from ticket format).
type	Integer(4)	The type of ticket format. (See Index)
zplTemplateId	Long(12)	The unique identifier to the ZPL template to be used for printing.
formatReference	String(255)	A reference to the format content to use for this format.
defaultFormat	Boolean	True if this is the default form for the type, False otherwise.
defaultPrinterId	Long(6)	Teh default printer identifier for the format.
createDate	Date	The date the format was created.
createUser	String(128)	The user that created the format.
updateDate	Date	The date the format was last updated.
updateUser	String(128)	The user that last updated the format.

API: findTicketPrinters

Finds the printers available to print tickets.

API Basics

Endpoint URL	{base URL}/printers
Method	GET
Success Response	200 OK
Processing Type	Synchronous
Input	Query parameters
Output	List of printers

Query Params

Column	Type	Definition
storeId	Long(10)	Include only ticket printers available at this store.

Output Data Definition

Column	Type	Definition
printerId	Long(6)	The unique identifier of the printer.
storeId	Long(10)	The identifier of the store the printer is assigned to.
printerType	Integer(3)	The print type of the printer (see Index).
printerDescription	String(200)	A description of the printer.
printerUri	String(300)	The URI address of the printer.
printerName	String(128)	The logical name of the printer.
defaultManifest	Boolean	True if the printer is the default printer at the store for manifest printing.
defaultPreshipment	Boolean	True if the printer is the default printer at the store for preshipment printing.

Additional Data Definitions

Ticket Format Type

ID	Status
1	Item Basket
2	Shelf Label

Price Type

ID	Status
1	Permanent
2	Promotional
3	Clearance
4	Clearance Reset

Printer Type

ID	Status
1	Item Ticket
2	Shelf Label
3	Postscript

Rest Service: Transfer

This service allows the creation, modification, and cancellation of transfers from an external application. This does not include allocations.

Service Base URL

The Cloud service base URL follows the format:

https://<external_load_balancer>/<cust_env>/siocs-int-services/api/transfers

API Definitions

API	Description
readTransfer	Reads the full detail of a transfer.
findTransfers	Finds transfers headers based on a set of input criteria to search for.
findTransferContexts	Retrieves all available transfer contexts for creating a new transfer.
requestTransfer	Moves the status of a new transfer request to requested, notifying the sending location of the transfer.
cancelRequest	Moves the status of a new transfer request to canceled if the request is still new or a work in progress.
rejectRequest	Rejects the requested transfer as long as it is still only requested or request in progress.
approveRequest	Moves the transfer to approved status and does inventory shifts if the transfer is "Requested" or "Request In Progress" and goes into "Approved"
cancelTransfer	Moves the status a transfer to canceled, but only if it is in the "Transfer In Progress" status.
closeTransfer	Closes out a transfer as long as the transfer is open and not yet in shipping status.

API: Read Transfer

Read a Transfer.

Table 7-22 API Basics

Endpoint URL	/{transferId}
Method	GET
Successful Response	200 OK

Table 7-22 (Cont.) API Basics

Processing Type	Synchronous
Input	None
Output	Transfer

Table 7-23 Path Parameter Definitions

Parameter	Definition
transferId	The internal identifier of the transfer header.

Table 7-24 Output Data Definition

Attribute	Type	Definition
transferId	Long(15)	The unique identifier of the transfer.
externalTransferId	String(128)	An external identifier supplied from an external system.
distroNumber	String(128)	A distribution number (used for integration with other systems). It is another type of external identifier.
sourceLocationType	Integer	The location type of the source of the transfer. See Additional Data Definitions.
sourceLocationId	Long(10)	The identifier of the source location of the transfer.
destinationLocationType	Integer	The location type of the destination of the transfer. See Additional Data Definitions.
destinationLocationId	Long(10)	The identifier of the destination location of the transfer.
status	Integer	The current status of the transfer. See Additional Data Definitions
originType	Integer	The origin type of the transfer. See Additional Data Definitions
contextTypeId	Long(18)	Unique identifier of a context associated to the transfer.
contextTypeCode	Long(4)	An external code of the context.
contextValue	String(25)	A value or some information related to the context associated to the transfer.
customerOrderNumber	String(128)	External system identifier of the customer order.
fulfillmentOrderNumber	String(128)	External system identifier of the fulfillment order.
allowPartialDelivery	Boolean	True indicates that the partial delivery is allowed for the transfer, false indicates it is not.
useAvailable	Boolean	True indicates the transfer must use available stock, false indicates it uses unavailable stock.
authorizationCode	String(12)	An authorization code assigned to the transfer.
notAfterDate	Date	Date after which the transfer is no longer valid.
createDate	Date	The date this record was created.
createUser	String(128)	The user that created this record.

Table 7-24 (Cont.) Output Data Definition

updateDate	Date	The date this record was last updated.
updateUser	String(128)	The user that last updated this record.
requestDate	Date	The date the transfer was requested.
requestUser	String(128)	The user that requested the transfer.
approvalDate	Date	The date the transfer was approved.
approvalUser	String(12)	The user that approved the transfer.
importId	String(128)	The identifier from an original data seed import.
lineItems	Collection	A collection of transfer line items.

Table 7-25 Transfer Line Item Data Definition

Column	Type	Definition
lineId	Long(15)	The unique internal identifier of the transfer line item.
itemId	String(25)	The item identifier.
caseSize	BigDecimal(10,2)	The case size associated to this line item.
quantityRequested	BigDecimal(20,4)	The quantity that was requested.
quantityApproved	BigDecimal(20,4)	The quantity that was approved.
quantityShipping	BigDecimal(20,4)	The quantity that is currently in shipping.
quantityShipped	BigDecimal(20,4)	The quantity that has currently shipped.
quantityReceived	BigDecimal(20,4)	The quantity that has been received into stock.
quantityDamaged	BigDecimal(20,4)	The quantity that has been received as damaged.
preferredUom	String(4)	The preferred unit of measure of the transfer line item.

API: Find Transfer

This API is used to find transaction headers for transfers. If the number of transfer found exceeds 10,000, a maximum limit error will be returned. Additional or more limiting input criteria will be required.

Table 7-26 API Basics

Endpoint URL	
Method	GET
Successful Response	200 OK
Processing Type	Synchronous
Input	Query Parameters
Output	List of Transfer Headers

Table 7-27 Query Parameter Definitions

Attribute	Type	Definition

Table 7-27 (Cont.) Query Parameter Definitions

externalTransferId	String(128)	Include only records with this external transfer identifier.
sourceLocationId	Long(10)	Include only records with this source location identifier.
sourceLocationType	Integer	Include only records with this source location type. See Additional Data Definitions
destinationLocationId	Long(10)	Include only records with this destination location identifier.
destinationLocationType	Integer	Include only records with this destination location type. See Additional Data Definitions.
status	Integer	Include only records with this status. See Additional Data Definitions.
itemId	String(25)	Include only records with this item on them.
updateDateFrom	Date	Include records with a last update date on or after this date.
updateDateTo	Date	Include records with a first update date on or before this date.

Attribute	Type	Definition
transferId	Long(15)	The unique identifier of the transfer.
externalTransferId	String(128)	An external identifier supplied from an external system.
sourceLocationType	Integer	The type of the source location of the transfer. See Additional Data Definitions.
sourceLocationId	Long(10)	The identifier of the source location of the transfer.
destinationLocationType	Integer	The type of the destination location of the transfer. See Additional Data Definitions.
destinationLocationId	Long(10	The identifier of the destination location of the transfer.
status	Integer	The current status of the transfer. See Additional Data Definitions.
originType	Integer	The origin type of the transfer. See Additional Data Definitions.
customerOrderNumber	String(128)	The external customer order number.
fulfillmentOrderNumber	String(128)	The external fulfillment order number.
notAfterDate	Date	Date after which the transfer is no longer valid.
createDate	Date	The date this record was created.
updateDate	Date	The date this record was last updated.
requestDate	Date	The date the transfer was requested.
approvalDate	Date	The date the transfer was approved.

API: Request Transfer

This API finalizes the request of the transfer and begins the approval process.

Table 7-28 API Basics

Endpoint URL	/{transferId}/requests
Method	POST
Successful Response	204 No Content
Processing Type	Synchronous
Input	None
Output	None

Table 7-29 Path Parameter Definitions

Parameter	Definition
transferId	The internal identifier of the transfer header.

API: Approve Request

Approves a transfer request for shipment moving it to the shipping stage.

Table 7-30 API Basics

Endpoint URL	/{transferId}/requests/approve
Method	POST
Successful Response	204 No Content
Processing Type	Synchronous
Input	None
Output	None

Table 7-31 Path Parameter Definitions

Parameter	Definition
transferId	The internal identifier of the transfer header.

API: Reject Request

Rejects the transfer request closing out the transfer. This is done when the approver of a request decides not to fulfill it.

Table 7-32 API Basics

Endpoint URL	/{transferId}/requests/reject
Method	POST
Successful Response	204 No Content
Processing Type	Synchronous
Input	None

Table 7-32 (Cont.) API Basics

Output	None

Table 7-33 Path Parameter Definitions

Parameter	Definition
transferId	The internal identifier of the transfer header.

API: Cancel Request

Cancels a transfer request closing out the transfer. This is done when the requester decides the transfer is not needed prior to finishing their request.

Table 7-34 API Basics

Endpoint URL	/{transferId}/requests/cancel
Method	POST
Successful Response	204 No Content
Processing Type	Synchronous
Input	None
Output	None

Table 7-35 Path Parameter Definitions

Parameter	Definition
transferId	The internal identifier of the transfer header.

API: Cancel Transfer

Cancels a transfer that has been approved.

Table 7-36 API Basics

Endpoint URL	/{transferId}/cancel
Method	POST
Successful Response	204 No Content
Processing Type	Synchronous
Input	None
Output	None

Table 7-37 Path Parameter Definitions

Parameter	Definition
transferId	The internal identifier of the transfer header.

API: Close Transfer

Closes a transfer that has been approved or approved and partially shipped.

Table 7-38 API Basics

Endpoint URL	/{transferId}/close
Method	POST
Successful Response	204 No Content
Processing Type	Synchronous
Input	None
Output	None

Table 7-39 Path Parameter Definitions

Parameter	Definition
transferId	The internal identifier of the transfer header.

API: Find Transfer Contexts

Reads all the available contexts for a transfer.

Table 7-40 API Basics

Endpoint URL	/contexts
Method	GET
Successful Response	200 OK
Processing Type	Synchronous
Input	None
Output	List of Transfer Contexts

Table 7-41 Output Data Definition

Attribute	Type	Definition
contextId	Long(18)	The unique identifier of the context.
code	String(6)	A short code associated to the context
description	String(120)	A description of the context.
sequence	Long	A sorting sequence for the context.

Additional Data Definitions

Table 7-42 Transfer Status

Value	Definition

Table 7-42 (Cont.) Transfer Status

1	New Request
2	Requested
3	Request In Progress
4	Rejected Request
5	Canceled Request
6	Transfer In Progress
7	Approved
8	In Shipping
9	Completed Transfer
10	Canceled Transfer

Table 7-43 Transfer Location Type

Value	Definition
1	Store
2	Warehouse
3	Finisher

Table 7-44 Transfer Origin Type

Value	Definition
1	External
2	Internal
3	Adhoc

Table 7-45 Transfer Status For Query Parameters

Value	Definition
1	New Request
2	Requested
3	Request In Progress
4	Rejected Request
5	Canceled Request
6	Transfer In Progress
7	Approved
8	In Shipping
9	Completed Transfer
10	Canceled Transfer
99	Active

Rest Service: Transfer Shipment

This service allows the creation, modification, and cancellation of transfer shipments from an external application.

Service Base URL

The Cloud service base URL follows the format:

https://<external_load_balancer>/<cust_env>/siocs-int-services/api/tsfshipments

API Definitions

API	Definition
readShipment	Reads the transfer shipment. This does not include containers or items.
findShipments	Finds transfer shipment header information based on a set of criteria.
submitShipment	Submits the transfer shipment.
cancelSubmitShipment	Cancels the submission of a transfer shipment.
dispatchShipment	Dispatches a transfer shipment.
cancelShipment	Cancels a transfer shipment.
createShipment	Creates a transfer shipment
updateShipment	Updates transfer shipment header information
confirmCarton	Confirms a container for shipment.
cancelCarton	Cancels a container, removing it from the shipment.
openCarton	Opens a confirmed container so that it can be modified again.
createCarton	Adds a carton to the shipment.
updateCarton	Updates a carton on the shipment.

API: Read Shipment

Read a transfer shipment.

Table 7-46 API Basics

Endpoint URL	/{shipmentId}
Method	GET
Successful Response	200 OK
Processing Type	Synchronous
Input	None
Output	Transfer Shipment

Table 7-47 Path Parameter Definitions

Parameter	Definition

Table 7-47 (Cont.) Path Parameter Definitions

shipmentId	The internal identifier of the transfer shipment header.

Table 7-48 Output Data Definition

Column	Type	Definition
shipmentId	Long(15)	The unique internal identifier of the transfer shipment.
storeId	Long(10)	The unique store identifier that is the source of the shipment.
destinationType	Integer(2)	The location type of the destination. See Additional Data Definitions-Destination Type.
destinationId	Long(10)	The unique identifier of the destination.
asn	String(128)	The advance shipment notification number.
status	Integer(2)	The current status of the shipment. See Additional Data Definitions: Status
notAfterDate	Date	A date that the goods should not be shipped after.
authorizationCode	String(12)	An authorization code associated to the shipment.
billOfLadingId	Long(12)	The unique identifier of the bill of lading.
adhocDocumentId	Long(15)	An adhoc transfer document identifier associated to the shipment.
billOfLadingId	Long(15)	The bill of lading number.
alternateAddress	String(2000)	An alternate destination address.
carrierRole	Integer(2)	The type of carrier for the shipment. See Additional Data Definitions
carrierId	Long(10)	A unique identifier of a carrier for the shipment.
carrierServiceId	Long(10)	A unique identifier of a carrier service for the shipment.
alternateCarrierName	String(240)	The name of a third-party shipping company.
alternateCarrierAddress	String(2000)	The address of a third-party shipping company.
motive	String(120)	A motive for the shipment.
taxId	String(18)	The tax identifier of the supplier it is being shipped to.
trackingNumber	String(128)	A tracking number associated to the shipment.
dimensionId	Long(12)	The identifier of a dimension associated to the shipment.
weight	BigDecimal(12, 4)	The weight of the shipment.
weightUom	String(4)	The unit of measure of the weight of the shipment.
requestedPickupDate	Date	The requested pickup date.
fiscalDocumentRequestId	Long(20)	The identifier of the request for a fiscal document.
fiscalDocumentReferenceId	Long(20)	The unique identifier of the fiscal document.
fiscalDocumentNumber	String(255)	The fiscal document number.

Table 7-48 (Cont.) Output Data Definition

fiscalDocumentStatus	Integer(4)	The status of the fiscal document.
fiscalDocumentRejectReason	String(255)	A reason the fiscal document was rejected.
fiscalDocumentUrl	String(255)	A URL to the fiscal document.
createUser	String(128)	The user that created the shipment record.
createDate	Date	The date the shipment record was created.
updateUser	String(128)	The user that last updated the shipment.
updateDate	Date	The last date the shipment was updated.
submitUser	String(128)	The user that submitted the shipment record.
submitDate	Date	The date the shipment was submitted within EICS.
dispatchUser	String(128)	The user that dispatched the shipment.
dispatchDate	Date	The date the shipment was dispatched within EICS.
cartons	Collection	A collection of cartons on the shipment.

Table 7-49 Output Data Definition (Cartons)

Column	Type	Definition
cartonId	Long(10)	The unique identifier of the record.
externalCartonId	String(128)	A container identifier from an external system.
status	Integer(4)	The current status of the container.
dimensionId	Long(10)	The shipment container dimension identifier.
weight	BigDecimal(12,4)	The weight of the container.
weightUom	String(4)	The unit of measure of the weight of the container.
trackingNumber	String(128)	A tracking number for the container.
useAvailable	Boolean	True indicates use only available inventory, False indicates use non-available inventory.
restrictionLevel	Integer(4)	A hierarchy restriction level for items in the container.
customerOrderRelated	Integer(4)	The customer order related value (see Additional Data Definitions).
createUser	String(128)	The user that created the container in EICS.
createDate	Date	The date the container was created in EICS.
updateUser	String(128)	The user that last updated the container in EICS.
updateDate	date	The date the container was last updated in EICS.
approvalUser	String(128)	The user that approved the container in EICS.
approvalDate	Date	The date the container was approved in EICS.
lineItems	Collection	The line items in the container.

Table 7-50 Output Data Definition (Line Item)

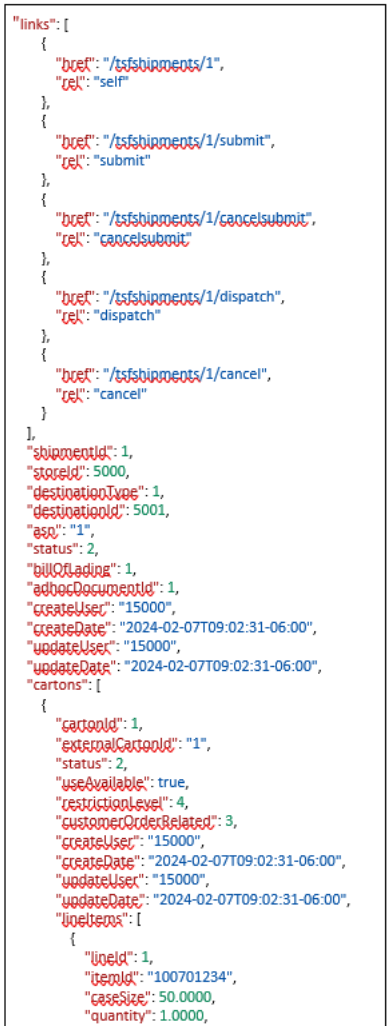
Column	Type	Definition

Table 7-50 (Cont.) Output Data Definition (Line Item)

lineId	Long(15)	The unique identifier of the record.
itemId	String(25)	The unique identifier of the item.
caseSize	BigDecimal(10,2)	The case size of this line item.
quantity	BigDecimal(20,4)	The quantity being shipped.
transferId	Long(12)	The unique identifier of the associated transfer.
externalTransferId	Long(15)	The external system identifier of the associated transfer.
customerOrderNumber	String(128)	The customer order number if line item is for a customer order.
fulfillmentOrderNumber	String(128)	The fulfillment order number if line item is for a customer order.
preferredUom	String(4)	The preferred unit of measure for the shipment quantity.
shipmentReasonId	Long(15)	A unique identifier of a shipment reason associated to this item.
uins	List<String>	A list of UINs that are shipped. The number of UINs must match the quantity attribute.

Example

Figure 7-9 Example




```
"links": [
    {
      "href": "/tsfshipments/1",
      "rel": "self"
    },
    {
      "href": "/tsfshipments/1/submit",
      "rel": "submit"
    },
    {
      "href": "/tsfshipments/1/cancelsubmit",
      "rel": "cancelsubmit"
    },
    {
      "href": "/tsfshipments/1/dispatch",
      "rel": "dispatch"
    },
    {
      "href": "/tsfshipments/1/cancel",
      "rel": "cancel"
    }
  ],
  "shipmentId": 1,
  "storeId": 5000,
  "destinationType": 1,
  "destinationId": 5001,
  "asn": "1",
  "status": 2,
  "billOfLading": 1,
  "adhocDocumentId": 1,
  "createUser": "15000",
  "createDate": "2024-02-07T09:02:31-06:00",
  "updateUser": "15000",
  "updateDate": "2024-02-07T09:02:31-06:00",
  "cartons": [
    {
      "cartonId": 1,
      "externalCartonId": "1",
      "status": 2,
      "useAvailable": true,
      "restrictionLevel": 4,
      "customerOrderRelated": 3,
      "createUser": "15000",
      "createDate": "2024-02-07T09:02:31-06:00",
      "updateUser": "15000",
      "updateDate": "2024-02-07T09:02:31-06:00",
      "lineItems": [
        {
          "lineId": 1,
          "itemId": "100701234",
          "caseSize": 50.0000,
          "quantity": 1.0000,
```

Figure 7-10 Example, continued

API Find Transfer Shipment

Search for shipments based on a set of criteria.

If more than 10,000 shipments are found a "Results Too Large" response is returned and search criteria will need to be more restrictive.

Table 7-51 API Basics

Endpoint URL	
Method	GET
Successful Response	200 OK
Processing Type	Synchronous
Input	Query Parameters
Output	List of Transfer Shipments Headers

Table 7-52 Query Parameter Definitions

Attribute	Type	Definition
storeId	Long(10)	Include only transfer shipments from this store.
destinationType	Integer(2)	Include only transfer shipments to this destination type.

Table 7-52 (Cont.) Query Parameter Definitions

destinationId	Long(10)	Include only transfer shipments to this destination identifier.
asn	String(128)	Include only transfer shipments with this ASN.
status	Integer(2)	Include only transfer shipments in this status.
updateDateFrom	String	Include only transfer shipments last updated on or after this date.
updateDateTo	String	Include only transfer shipments last updated on or before this date.

Table 7-53 Output Data Definition

Column	Type	Definition
shipmentId	Long(15)	The unique internal identifier of the transfer shipment.
storeId	Long(10)	The unique store identifier that is the source of the shipment.
destinationType	Integer(2)	The location type of the destination. See Additional Data Definitions: Destination Type.
destinationId	Long(10)	The unique identifier of the destination location.
asn	String(128)	The advance shipment notification number.
status	Integer(4)	The current status of the shipment. See Additional Data Definitions: Status
notAfterDate	Date	A date that the goods should not be shipped after.
createDate	Date	The date the shipment record was created.
updateDate	Date	The last date the shipment was updated.
submitDate	Date	The date the shipment was submitted within EICS.
dispatchDate	Date	The date the shipment was dispatched within EICS.

API: Submit Shipment

Submits the transfer shipment moving it to a noneditable state while it is waiting to be dispatched.

Table 7-54 API Basics

Endpoint URL	/{shipmentId}/submit
Method	POST
Successful Response	204 No Content
Processing Type	Synchronous
Input	None
Output	None

Table 7-55 Path Parameter Definitions

Parameter	Definition
shipmentId	The internal identifier of the transfer shipment header.

API: Cancel Submit Shipment

Cancels the submissions of the transfer shipment return it to an in progress and editable state.

Table 7-56 API Basics

Endpoint URL	/{shipmentId}/cancelsubmit
Method	POST
Successful Response	204 No Content
Processing Type	Synchronous
Input	None
Output	None

Table 7-57 Path Parameter Definitions

Parameter	Definition
shipmentId	The internal identifier of the transfer shipment header.

API: Dispatch Shipment

Dispatches the transfer shipment updating all the inventory positions and closing the shipment.

Table 7-58 API Basics

Endpoint URL	/{shipmentId}/dispatch
Method	POST
Successful Response	204 No Content
Processing Type	Synchronous
Input	None
Output	None

Table 7-59 Path Parameter Definitions

Parameter	Definition
shipmentId	The internal identifier of the transfer shipment header.

API: createShipment

This API is used to create a new transfer shipment whose status is "In Progress."

> **Note:**
>
> Container item/reason combination cannot be duplicated within the container.

Table 7-60 API Basics

Endpoint URL	{base URL}
Method	POST
Successful Response	200 OK
Processing Type	Synchronous
Input	Transfer Shipment
Output	Transfer Shipment Status
Maximum Input Limit	1,000 overall line items on shipment

Table 7-61 Input Data Definition

Attribute	Type	Req	Definition
storeId	Long(10	X	The identifier of the store shipping the goods.
destinationType	Integer(2)	X	The location type of the destination (see Index).
destinationId	Long(10)	X	The location identifier of the destination
asn	String(15)		The advance shipping number from an external system.
authorizationCode	String(12)		A vendor authorization code. It may be required for some suppliers.
displayCartons	Boolean		True if cartons should be displayed, otherwise false.
addressType	Integer		The type of return address. See Index.
carrierRole	Integer(2)	X	The type of carrier for the shipment. See Index
carrierId	Long(10)		A unique identifier of a carrier for the shipment.
carrierServiceId	Long(10)		A unique identifier of a carrier service for the shipment.
alternateAddress	String(2000)		An alternate destination address.
alternateCarrierName	String(240)		The name of a third-party shipping company.
alternateCarrierAddress	String(2000)		The address of a third-party shipping company.
motiveId	Long(18)		The unique identifier of a bill of lading motive.
taxId	taxId		The tax identifier of the supplier it is being shipped to.
trackingNumber	String(128)		A tracking number associated to the shipment.
dimensionId	Long(12)		The identifier of a dimension associated to the shipment.
weight	BigDecimal(12,4)		The weight of the shipment.
weightUom	String(4)		The unit of measure of the weight of the shipment.

Table 7-61 (Cont.) Input Data Definition

requestedPickupDate	Date		The requested pickup date.
cartons	Collection	X	A group of cartons to create along with the shipment.
notes	Collections of Strings		A collection of up to 100 notes.

Table 7-62 Shipment Carton Data Definition

Payload	Type	Req	Definition
externalCartonId	String(128)		A container identifier from an external system.
trackingNumber	String(128)		The tracking number of the container.
restrictionLevel	Integer(4)		A hierarchy restriction level for items in the container.
cartonSizeId	Long(10)		The shipment container dimension identifier.
weight	Long(12,4)		The weight of the container.
weightUom	String(4)		The unit of measure of the weight of the container.
useAvailable	Boolean		True indicates available inventory will be used, false indicates unavailable inventory should be used.
lineItems	Collection	X	A collection of line items to create along with the carton.

Table 7-63 Shipment Lien Item Data Definition

Payload	Type	Req	Definition
itemId	String(25)	X	The unique identifier of the SKU item.
transferId	Long(12)	X	The unique identifier of a transfer this item is being shipped for.
reasonId	Long(15)		The unique identifier of the shipment reason associated to this line item. It is not allowed for available inventory and required for unavailable inventory.
quantity	BigDecimal(12,4)	X	The quantity to ship.
caseSize	BigDecimal(10,2)		The case size of the item for this particular shipment.
uins	Collecton<String>		A collection of UINs to ship. This number of UINs must match the quantity.

Table 7-64 Output Data Definition

Attribute	Type	Definition
shipmentId	Long(15)	The unique identifier of the shipment.
storeId	Long(10)	The store identifier of the store shipping the goods.

Table 7-64 (Cont.) Output Data Definition

asn	String(15)	The Advancing Shipping Number.
status	Integer(2)	The shipment status. See Index.

Figure 7-11 Example Code


```
{
  "storeId": 5000,
  "destinationType": 1,
  "destinationId": 5001,
  "addressType": 1,
  "carrierRole": 1,
  "cartons": [
    {
      "externalCartonId": "100637113",
      "lineItems": [
        {
          "itemId": "100637113",
          "reasonId": 10,
          "transferId": 1,
          "quantity": 10
        },
        {
          "itemId": "100637121",
          "reasonId": 11,
          "transferId": 1,
          "quantity": 10
        }
      ]
    }
  ]
}
```


API: Cancel Shipment

Cancels the transfer shipment reversing any reserved inventory currently picked.

Table 7-65 API Basics

Endpoint URL	/{shipmentId}/cancel
Method	POST
Successful Response	204 No Content
Processing Type	Synchronous
Input	None
Output	None

Table 7-66 Path Parameter Definitions

Parameter	Definition
shipmentId	The internal identifier of the transfer shipment header.

API: updateShipment

This API is used to update to the header portion of a shipment as well as its bill of lading information.

The shipment header cannot be updated while containers/cartons are currently confirmed for shipping.

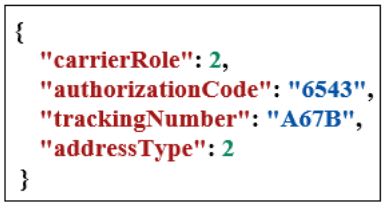
Table 7-67 API Basics

Endpoint URL	{base URL}/{shipmentId}
Method	POST
Successful Response	204 No Content
Processing Type	Synchronous
Path Parameter	The identifier of the transfer shipment
Input	Transfer Shipment

Table 7-68 Input Data Definition

Column	Type	Definition
authorizationCode	String(12)	An authorization code associated to the shipment.
displayCartons	Boolean	True if cartons should be displayed, otherwise false.
addressType	Integer(2)	The type of return address. See Index.
carrierRole	Integer(2)	The type of carrier for the shipment. See Index.
carrierId	Long(10)	A unique identifier of a carrier for the shipment.
carrierServiceId	Long(10)	A unique identifier of a carrier service for the shipment.
alternateAddress	String(2000)	An alternate destination address.
alternateCarrierName	String(240)	The name of a third-party shipping company.
alternateCarrierAddress	String(2000)	The address of a third-party shipping company
motiveId	Long(18)	The unique identifier of a bill of lading motive.
taxId	String(18)	The tax identifier of the supplier it is being shipped to.
trackingNumber	String(128)	A tracking number associated to the shipment.
dimensionId	Long(12)	The identifier of a dimension associated to the shipment.
weight	BigDecimal(12, 4)	The weight of the shipment.
weightUom	String(4)	The unit of measure of the weight of the shipment.
requestedPickupDate	Date	The requested pickup date.
notes	Collection(String)	A collection of up to 100 notes to add to the notes associated to the shipment.

Figure 7-12 Code Example



```
{
  "carrierRole": 2,
  "authorizationCode": "6543",
  "trackingNumber": "A67B",
  "addressType": 2
}
```

API: createCarton

This API is used to add a new carton/container to a "New" or "In Progress" shipment.

Container item/reason combination cannot be duplicated within the container.

Table 7-69 API Basics

Endpoint URL	{base URL}/{shipmentId}/cartons
Method	POST
Successful Response	200 OK
Processing Type	Synchronous
Path Parameter	The identifier of the transfer shipment
Input	Transfer Shipment Carton
Output	Transfer Shipment Carton Status
Maximum Input Limit	1,000 overall line items on carton

Table 7-70 Input Data Definition

Column	Type	Req	Definition
externalCartonId	String(128)		A carton identifier or barcode label from an external system.
cartonSizeId	Long(10)		The shipment container dimension identifier.
weight	BigDecimal(12,4)		The weight of the container.
weightUom	String(4)		The unit of measure of the weight of the container.
trackingNumber	String(128)		A tracking number for the container.
useAvailableInventory	Boolean		True indicates use only available inventory, False indicates use non-available inventory.
restrictionLevel	Integer(4)		A hierarchy restriction level for items in the container.
lineItems	Collection	X	The line items in the container.

Table 7-71 Shipment Line Item Data Definition

Column	Type	Req	Definition
itemId	String(25)	X	The unique identifier of the item.

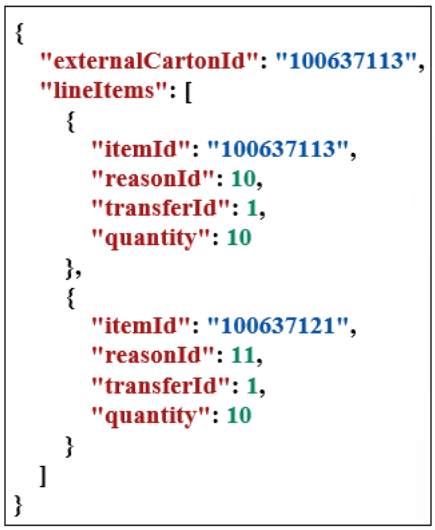
Table 7-71 (Cont.) Shipment Line Item Data Definition

caseSize	BigDecimal(10,2)		The case size of this line item.
quantity	BigDecimal(20,4)	X	The quantity to be shipped.
transferId	Long(12)	X	The unique identifier of the associated transfer.
reasonId	Long(15)		A unique identifier of a shipment reason associated to this item. It is not allowed for available inventory and required for unavailable inventory.
uins	List<String>		A list of UINs to be shipped. This must match the quantity.

Table 7-72 Output Data Definition

Attribute	Type	Definition
shipmentId	Long(15)	The unique identifier of the shipment.
cartonId	Long(15)	The unique identifier of the new carton.
externalId	String(128)	The external identifier or barcode label of the container.
status	Integer(2)	The carton status. See Index.

Figure 7-13 Code Example



```
{
  "externalCartonId": "100637113",
  "lineItems": [
    {
      "itemId": "100637113",
      "reasonId": 10,
      "transferId": 1,
      "quantity": 10
    },
    {
      "itemId": "100637121",
      "reasonId": 11,
      "transferId": 1,
      "quantity": 10
    }
  ]
}
```

API: Confirm Carton

Confirms a transfer shipment container, completing the container and making it non-editable and awaiting dispatch.

Table 7-73 API Basics

Endpoint URL	cartons/{cartonId}/confirm
Method	POST
Successful Response	204 No Content
Processing Type	Synchronous
Input	None
Output	None

Table 7-74 Path Parameter Definitions

Parameter	Definition
cartonId	The internal identifier of the transfer shipment carton.

API: Cancel Carton

Cancels a transfer shipment container.

Table 7-75 API Basics

Endpoint URL	cartons/{cartonId}/cancel
Method	POST
Successful Response	204 No Content
Processing Type	Synchronous
Input	None
Output	None

Table 7-76 Path Parameter Definitions

Parameter	Definition
cartonId	The internal identifier of the transfer shipment carton.

API: Open Carton

Opens a transfer shipment container.

Table 7-77 API Basics

Endpoint URL	cartons/{cartonId}/open
Method	POST
Successful Response	204 No Content
Processing Type	Synchronous
Input	None
Output	None

Table 7-78 Path Parameter Definitions

Parameter	Definition
cartonId	The internal identifier of the transfer shipment carton.

API: updateCarton

This API is used to update an existing carton that is in "New" or "In Progress" shipment.

Table 7-79 API Basics

Endpoint URL	{base URL}/cartons/{cartonId}
Method	POST
Successful Response	200 OK
Processing Type	Synchronous
Path Parameter	The identifier of the carton
Input	Transfer Shipment Carton
Output	Transfer Shipment Carton Status
Maximum Input Limit	1,000 overall line items on carton

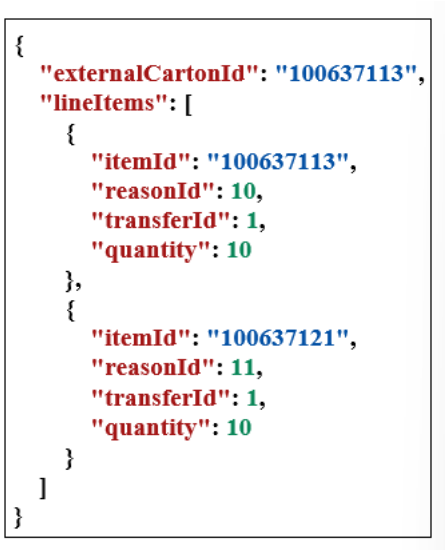
Table 7-80 Input Data Definition

Payload	Type	Req	Definition
externalCartonId	String(128)		A container identifier from an external system.
trackingNumber	String(128)		The tracking number of the container.
restrictionLevel	Integer(4)		A hierarchy restriction level for items in the container.
cartonSizeId	Long(10)		The shipment container dimension identifier.
weight	Long(12,4)		The weight of the container.
weightUom	String(4)		The unit of measure of the weight of the container.
lineItems	Collection		A collection of up to 1,000 line items to update or add within the carton. See TransferShipmentUpdateCartontemIdo.

Payload	Type	Req	Definition
itemId	String(25)	X	The unique identifier of the SKU item.
transferId	Long(12)	X	The unique identifier of the associated transfer.
reasonId	Long(15)		The unique identifier of the shipment reason associated to this line item. It is not allowed for available inventory and required for unavailable inventory.
quantity	BigDecimal(12,4)	X	The quantity shipped. Reducing this to 0 will remove the line item from the shipment.

caseSize	BigDecimal(10,2)	The case size of the item for this particular shipment.
uins	Collection<String>	The UINs associated to the item quantities. The number of UINS must match the quantity shipped.

Figure 7-14 Code Example



```
{
  "externalCartonId": "100637113",
  "lineItems": [
    {
      "itemId": "100637113",
      "reasonId": 10,
      "transferId": 1,
      "quantity": 10
    },
    {
      "itemId": "100637121",
      "reasonId": 11,
      "transferId": 1,
      "quantity": 10
    }
  ]
}
```

Additional Data Definitions

Table 7-81 Destination Type

Value	Description
1	Store
2	Warehouse
3	Finisher

Table 7-82 Status

Value	Description
1	New
2	In Progress
3	Submitted
4	Shipped
5	Canceled

Table 7-83 Customer Order Related

Value	Description
1	Yes
2	Mixed
3	No

REST Service Translations

This page captures the service APIs related to retrieving translations. It allows the translation of such things as labels and item descriptions.

Service Base URL

The Cloud service base URL follows the format:

https://<external_load_balancer>/<cust_env>/siocs-int-services/api/translations

API Definitions

API	Description
Find Locales	Finds all locales that can be used to translation a series of text keys.
Find Translations	Finds the translations for a series of text keys, translating into text for the locale if it is available.
Find Item Descriptions	Finds the translations for item descriptions, translating it to the text of the locale if it is available.

API: Find Locales

Finds all locales available for use in translation.

API Basics

Endpoint URL	{base URL}/locales
Method	GET
Successful Response	200 OK
Processing Type	Synchronous
Input	Criteria
Output	List of locales
Max Response Limit	N/A

Output Data Definition

Attribute	Data Type	Description

localeId	Long	The SIOCS internal unique identifier of the record.
language	String	A code representing a language (ISO 639 alpha-2 or alpha-3 language code).
country	String	A code representing a country of the language (ISO 3166 alpha-2 country code or UN M.49 numeric-3 area code.)
variant	String	A code representing a variant of the country of the language (an arbitrary value indication the variant).
description	String	A description of the locale.

Example Output

[

{

"localeId": 1,

"language": "en",

"description": "English"

},

{

"localeId": 2,

"language": "de",

"description": "German"

},

{

"localeId": 3,

"language": "fr",

"description": "French"

}

]

API: Find Translations

Searches for translations for text keys and a given locale.

API Basics

Endpoint URL	{base URL}/find
Method	POST
Successful Response	200 OK

Processing Type	Synchronous
Input	Criteria
Output	A map of translation key and its translation
Max Response Limit	N/A

Input Data Definition

Attribute	Data Type	Required	Description
localeId	Long(12)	Yes	Unique identifier of the Locale to translate keys for (see find Locales).
keys	List<String(600)>	Yes	A list of text keys to attempt to translate.

Example Input

{

"localeId": 1,

"keys": [

"invAdjReason.1",

"invAdjReason.2"

]

}

Output Data Definition

Attribute	Data Type	Description
values	Map<String, String>	A map where the key is the translation key and the value is the translation value.

Example Output

{

"invAdjReason.2": "Shrinkage",

"invAdjReason.1": "Wastage"

}

API: Find Items Descriptions

Finds the translations for item descriptions, translating it to the text of the locale if it is available.

API Basics

Endpoint URL	{base URL}/items
Method	POST

Successful Response	200 OK
Processing Type	Synchronous
Input	Criteria
Output	A list of translation items
Max Response Limit	N/A

Input Data Definition

Attribute	Data Type	Required	Description
LocaleId	Long (12)	Yes	Unique identifier of the Locale to translate descriptions for (see findLocales).
itemIds	List<String(25)>	Yes	A list of items to get descriptions for within the locale.

Example Input

{

"localeId": 1,

"itemIds": [

"100637121",

"100637113"

]

}

Output Data Definition

Attribute	Data Type	Required	Description
itemId	String		The item identifier.
shortDescription	String		The short description in the locale's text if available.
longDescription	String		The long description in the locale's text if available.

Example Output

[

{

"itemId": "100637121",

"shortDescription": "translation value for 100637121",

"longDescription": "translation value for 100637121"

},

{

"itemId": "100637113",

"shortDescription": "translation value for 100637113",

"longDescription": "translation value for 100637113"

}

]

Rest Service: Vendor Delivery

This service allows the import and handling of direct store deliveries from vendors/suppliers.

Service Base URL

The Cloud service base URL follows the format:

https://<external_load_balancer>/<cust_env>/siocs-int-services/api/dsds

API Definitions

API	Description
readDelivery	Read the full details of a vendor delivery.
findDeliveries	Find vendor delivery headers based on input search criteria.
receiveDelivery	Receives the expected quantities of a vendor delivery so that it is ready to be confirmed.
confirmDelivery	Confirm the receipt of a vendor delivery updating inventory positions with the receipt information.
rejectDelivery	Rejects the vendor delivery and do not allow it to be received.
cancelDelivery	Cancel a vendor delivery.
submitCarton	Moves the status of the carton to submitted and prevents further updates. The carton must still be confirmed. No inventory positions are updated via this operation.
cancelSubmitCarton	Opens a submitted carton for further updates, moving the status back to in-progress.
confirmCarton	Confirms the receipt a vendor delivery carton.
cancelCarton	Cancels a vendor delivery carton.
openCarton	Re-opens a completed carton after receipt allowing it to be received a second time (possibly adjusting quantities).

API: Read Delivery

Retrieves a vendor delivery.

Table 7-84 API Basics

Endpoint URL	/{deliveryId}
Method	GET

Table 7-84 (Cont.) API Basics

Successful Response	200 OK
Processing Type	Synchronous
Input	None
Output	Transfer Shipment

Table 7-85 Path Parameter Definitions

Parameter	Definition
deliveryId	The internal identifier of the vendor delivery header.

Table 7-86 Output Data Definition

Column	Type	Definition
deliveryId	Long(12)	The unique identifier of the delivery record.
storeId	Long(10)	The unique identifier of the store receiving the inventory.
supplierId	Long(10)	The unique identifier of the supplier shipping the inventory.
status	Integer(4)	The current status of the delivery. See Vendor Delivery Status.
originType	Integer(2)	The origin type of the delivery. See Vendor Delivery Origin Type.
purchaseOrderId	Long(12)	The purchase order that the delivery is associated to.
receiptNumber	Long	The global receipt number of the delivery used to identifier the receipt across applications through integration.
asn	String(128)	The advanced shipping notification of the delivery.
invoiceNumber	String(128)	A unique identifier of an invoice associated to this delivery.
invoiceCurrency	String	A currency code of the invoice cost.
invoiceAmount	BigDecimal	The value of the invoice cost.
customerOrderId	String(128)	A customer order identifier (from an external system) associated to the delivery.
fulfillmentOrderId	String(128)	A fulfillment order identifier (from an external system) associated to the delivery.
billOfLadingId	String(128)	An external identifier of a bill of lading record.
carrierName	String(128)	The name of the carrier.
carrierType	Integer(2)	The type of the carrier. See Carrier Type.
carrierCode	String(4)	Unique code that identifies the carrier.
countryCode	String(3)	A country code.
sourceAddress	String(1000)	The address of the source shipping location sending the delivery to the store.

Table 7-86 (Cont.) Output Data Definition

licensePlate	String(128)	The license plate of the delivery vehicle.
freightId	String(128)	A freight identifier associated to the delivery.
fiscalDocumentRequestId	Long(20)	The identifier of the request for a fiscal document.
fiscalDocumentReferenceId	Long(20)	The unique identifier of the fiscal document.
fiscalDocumentNumber	String(255)	The fiscal document number.
createDate	Date	The date the delivery record was created.
updateDate	Date	The date the delivery record was last updated.
expectedDate	Date	The expected date of the delivery.
invoiceDate	Date	The date of the delivery invoice.
receivedDate	Date	The date the delivery was received.
createUser	String(128)	The user that created the delivery .record.
updateUser	String(128)	The user who last updated the delivery record.
receivedUser	String(128)	The user who received the delivery record.
cartons	Collection	A list of cartons.

Table 7-87 Open Data Definition (Carton)

Column	Type	Definition
cartonId	Long(12)	The unique identifier of the carton record.
externalCartonId	String(128)	An external identifier of the carton.
referenceId	String(128)	A reference identifier to the carton.
status	Integer(4)	The current status of the carton (See Additional Data Definitions Vendor Delivery Carton Status)
damagedReason	String(128)	A reason for the carton damage that took place.
serialCode	Long(18)	A serial code for the carton.
trackingNumber	String(128)	The tracking number of the carton.
damageRemaining	Boolean	indicates all remaining quantities should be damaged on final receipt.
uinRequired	Boolean	True if a UIN item exists within the carton, otherwise false.
receiveAtShopFloor	Boolean	True if receive the inventory at shop floor, otherwise false.
qualityControl	Boolean	True indicates that the carton requires detailed receiving.
externalCreate	Boolean	True indicates the carton was externally created, false indicates it was created by EICS.
adjusted	Boolean	True indicates the carton is adjusted, otherwise false.
customerOrderRelated	Integer(4)	Customer Order Related Type (See Additional Data Definitions)

Table 7-87 (Cont.) Open Data Definition (Carton)

createUser	String(128)	The user who created the carton.
updateUser	String(128)	The user who last updated the carton.
receivedUser	String(128)	The user who received the carton.
createDate	Date	The date the carton was created.
updateDate	Date	The date the carton was last updated.
receivedDate	Date	The date the carton was received.
lineItems	Collection	The line items associated with the container.

Table 7-88 Output Data Definition (Line Item)

Column	Type	Definition
lineId	Long(12)	The unique identifier of the line item record.
itemId	String(25)	The unique identifier of the item.
caseSize	BigDecimal(10,2)	A number of units in the case that this item was shipped with.
quantityExpected	BigDecimal(20,4)	The total number of units expected on the delivery.
quantityReceived	BigDecimal(20,4)	The total number of units received on the delivery.
quantityDamaged	BigDecimal(20,4)	The total number of units received as damaged on the delivery.
quantityReceivedOverage	BigDecimal(20,4)	Amount of received inventory over expected quantities.
quantityDamagedOvarage	BigDecimal(20,4)	Amount of received damage inventory over expected quantities.
quantityPreviouslyReceived	BigDecimal(20,4)	Units previously received (captured at time container is re-opened after receipt)
quantityPreviouslyDamaged	BigDecimal(20,4)	Units previously received as damaged (captured at time container is re-opened after receipt)
unitCostCurrency	String(3)	The unit cost currency of this item delivery.
unitCostAmount	BigDecimal(12,4)	The unit cost value of this item delivery.
unitCostOverrideCurrency	String(3)	The override unit cost currency of this item delivery.
unitCostOverrideAmount	BigDecimal(12,4)	The override unit cost value of this item delivery.
purchaseOrderId	Long (12)	The internal unique identifier of the purchase order of this particular item delivery.
purchaseOrderNumber	String(128)	The external purchase order number of this particular item delivery.
customerOrderNumber	String(128)	The unique external customer order identifier of this particular item delivery.
fulfillmentOrderNumber	String(128)	The unique external fulfillment order identifier of this particular item delivery.
vendorProductNumber	String(256)	The vendor product number of the item.

Table 7-88 (Cont.) Output Data Definition (Line Item)

uins	Collection<String>	A list of UINs associated to the line item.

API: Find Vendor Delivery

API is used to find transaction headers for vendor deliveries.

If more than 10,000 deliveries are found, a "results too large" error will be returned. Limit the results with further search criteria.

Table 7-89 API Basics

Endpoint URL	
Method	GET
Successful Response	200 OK
Processing Type	Synchronous
Input	Query Parameters
Output	List of Vendor Shipments

Table 7-90 Query Parameter Definitions

Attribute	Type	Definition
storeId	Long(10)	Include only records where the delivery is to this store.
asn	String(128)	Include only records for this advanced shipping notification.
originType	Integer(2)	Include only records for this origin type. (See Additional Data Definitions: Vendor Delivery Origin Type).
status	Integer(4)	Include only records where the delivery is in this status. (See Additional Data Definitions: Vendor Delivery Criteria Status)
customerOrderNumber	String(128)	Include only records for this customer order number.
fulfillmentOrderNumber	String(128)	Include only records for this fulfilment order number.
invoiceNumber	String(128)	Include only records for this invoice number.
supplierId	Long(10)	Include only records for this supplier identifier.
purchaseOrderNumber	String(12)	Include only records where a line item is associated to this external purchase order number.
updateDateFrom	String	Include only records with a last update date on or after this date.
updateDateto	String	Include only records with a first update date on or before this date.

Table 7-91 Output Data Definition

Column	Type	Definition
deliveryId	Long(12)	The unique identifier of the delivery record.
storeId	Long(10)	The unique identifier of the store receiving the inventory.
supplierId	Long(10)	The unique identifier of the supplier shipping the inventory.
status	Integer(4)	The current status of the delivery. See Vendor Delivery Status.
originType	Integer(2)	The origin type of the delivery. See Vendor Delivery Origin Type.
purchaseOrderId	Long(12)	The purchase order that the delivery is associated to.
receiptNumber	Long	The global receipt number of the delivery used to identifier the receipt across applications through integration.
asn	String(128)	The advanced shipping notification of the delivery.
invoiceId	String(128)	A unique identifier of an invoice associated to this delivery.
customerOrderId	String(128)	A customer order identifier associated to the delivery.
fulfillmentOrderId	String(128)	A fulfillment order identifier (from an external system) associated to the delivery.
createDate	Date	The date the delivery record was created.
updateDate	Date	The date the delivery record was last updated.
expectedDate	Date	The expected date of the delivery.
invoiceDate	Date	The date of the delivery invoice.
receivedDate	Date	The date the delivery was received.

API: Receive Delivery

Updates the received quantities to quantities that were expected so that it is ready to be confirmed. It puts the delivery in the "In Progress" status.

Table 7-92 API Basics

Endpoint URL	/{deliveryId}/receive
Method	POST
Successful Response	204 No Content
Processing Type	Synchronous
Input	None
Output	None

Table 7-93 Path Parameter Definitions

Parameter	Definition
deliveryId	The internal identifier of the vendor delivery header.

API: Confirm Delivery

Confirms the delivery and receives the goods into inventory.

Table 7-94 API Basics

Endpoint URL	/{deliveryId}/confirm
Method	POST
Successful Response	204 No Content
Processing Type	Synchronous
Input	None
Output	None

Table 7-95 Path Parameter Definitions

Parameter	Definition
deliveryId	The internal identifier of the vendor delivery header.

API: Reject Delivery

Rejects the delivery without receiving goods, placing it in rejected status.

Table 7-96 API Basics

Endpoint URL	/{deliveryId}/reject
Method	POST
Successful Response	204 No Content
Processing Type	Synchronous
Input	None
Output	None

Table 7-97 Path Parameter Definitions

Parameter	Definition
deliveryId	The internal identifier of the vendor delivery header.

API: Cancel Delivery

Cancels the delivery without receiving the goods and places it in canceled status.

Table 7-98 API Basics

Endpoint URL	/{deliveryId}/cancel
Method	POST
Successful Response	204 No Content
Processing Type	Synchronous
Input	None
Output	None

Table 7-99 Path Parameter Definitions

Parameter	Definition
deliveryId	The internal identifier of the vendor delivery header.

API: Submit Carton

Moves the status of the carton to submitted and prevents further updates. The carton may still be confirmed. No inventory positions are updated via this operation.

Table 7-100 API Basics

Endpoint URL	cartons/{cartonId}/submit
Method	POST
Successful Response	204 No Content
Processing Type	Synchronous
Input	None
Output	None

Table 7-101 Path Parameter Definitions

Parameter	Definition
cartonId	The internal identifier of the vendor delivery carton.

API: Cancel Submit Carton

Opens a submitted carton for further updates, moving the status to in-progress.

Table 7-102 API Basics

Endpoint URL	cartons/{cartonId}/cancelsubmit
Method	POST
Successful Response	204 No Content
Processing Type	Synchronous
Input	None

Table 7-102 (Cont.) API Basics

Output	None

Table 7-103 Path Parameter Definitions

Parameter	Definition
cartonId	The internal identifier of the vendor delivery carton.

API: Confirm Carton

Confirms the final receipt of a vendor delivery carton.

Table 7-104 API Basics

Endpoint URL	cartons/{cartonId}/confirm
Method	POST
Successful Response	204 No Content
Processing Type	Synchronous
Input	None
Output	None

Table 7-105 Path Parameter Definitions

Parameter	Definition
cartonId	The internal identifier of the vendor delivery carton.

API: Cancel Carton

Cancels a vendor delivery carton.

Table 7-106 API Basics

Endpoint URL	cartons/{cartonId}/cancel
Method	POST
Successful Response	204 No Content
Processing Type	Synchronous
Input	None
Output	None

Table 7-107 Path Parameter Definitions

Parameter	Definition
cartonId	The internal identifier of the vendor delivery carton.

API: Open Carton

Re-open a completed carton after receipt allowing it to be received again.

Table 7-108 API Basics

Endpoint URL	cartons/{cartonId}/open
Method	POST
Successful Response	204 No Content
Processing Type	Synchronous
Input	None
Output	None

Table 7-109 Path Parameter Definitions

Parameter	Definition
cartonId	The internal identifier of the vendor delivery carton.

Additional Data Definitions

Table 7-110 Vendor Delivery Carton Status

Value	Status
1	New
2	In Progress
3	Submitted
4	Received
5	Damaged
6	Missing
7	Canceled

Table 7-111 Customer Order Related Type

Value	Status
1	Yes
2	Mix
3	No

Table 7-112 Vendor Delivery Status

Value	Status
1	New
2	In Progress

Table 7-112 (Cont.) Vendor Delivery Status

3	Received
4	Canceled
5	Rejected

Table 7-113 Vendor Delivery Criteria Status

Value	Status
1	New
2	In Progress
3	Received
4	Canceled
5	Rejected
99	Active

Table 7-114 Vendor Delivery Origin Type

Value	Status
1	Advanced Shipping Notification
2	Purchase Order
3	Dex-Nex
4	Manual/On the Fly

Table 7-115 Vendor Delivery Create Origin Type

Value	Status
2	Purchase Order
3	Dex-Nex
4	Manual/On the Fly

Table 7-116 Carrier Type

Value	Status
1	Corporate
2	Third Party

REST Service Warehouse

This service integrates warehouse and warehouse item foundation data as well as warehouse item inventory adjustments.

Asynchronous warehouse integration is processed through staged messages and is controlled by the MPS Work Type: DcsWarehouse.

Service Base URL

The Cloud service base URL follows the format:

https://<external_load_balancer>/<cust_env>/siocs-int-services/api/warehouses

API Definitions

API	Description
Import Warehouses	Imports a collection of warehouses into the system.
Delete Warehouse	Deletes a warehouse from the system.
Import Items	Imports a collection of warehouse items.
Delete Items	Deletes warehouse items from the system.
Import Adjustments	Imports a collection of warehouse items adjustments that took place.
Import Inventory	Imports a collection of warehouse item inventory to update the inventory positions.

API: Import Warehouses

This will import warehouses through foundation warehouse processing.

If more than 500 warehouses are sent in a single call, an input too large error will be returned.

API Basics

Endpoint URL	{base URL}/import
Method	POST
Successful Response	202 Accepted
Processing Type	Asynchronous
Input	List of warehouses to import
Output	None
Max Response Limit	500

Input Data Definition

Attribute	Data Type	Required	Description
warehouses	List of details	Yes	A list of warehouses to import.

Detail Data Definition

Attribute	Data Type	Required	Size	Description
warehouseId	Long (10)	Yes		The warehouse identifier.
name	String (150)	Yes	150	The name of the warehouse.

organizationUnit	String (15)	15	The organization the warehouse belongs to.
countryCode	String (3)	3	The ISO country code of the warehouse.
currencyCode	String (40)	3	The ISO currency code of the warehouse.

Example Output

{
"warehouses": [
{
"warehouseId": 64,
"name": "DownTownWarehouse-1",
"organizationUnit": "70001",
"countryCode": "IN",
"currencyCode": "INR"
},
{
"warehouseId": 65,
"name": "CitynWarehouse-1",
"organizationUnit": "70001",
"countryCode": "IN",
"currencyCode": "INR"
}
]
}
]

API: Delete Warehouse

Deletes a warehouse. The warehouse will not be deleted if any items remain ranged to the warehouse.

API Basics

Endpoint URL	{base URL}{warehouseId}/delete
Method	POST
Successful Response	202 Accepted

Processing Type	Synchronous
Input	None
Output	None
Max Response Limit	N/A

Path Parameter Definitions

Attribute	Definition
warehouseId	The internal identifier of the warehouse.

API: Import Items

Imports a collection of warehouse items.

If more than 5000 items are sent in a single call, an input too large error will be returned.

API Basics

Endpoint URL	{base URL}/{warehouseId}/items/import
Method	POST
Successful Response	202 Accepted
Processing Type	Asynchronous (High Volume)
Input	A list of warehouse items to import
Output	None
Max Response Limit	5000

Path Parameter Definitions

Attribute	Definition
warehouseId	The internal identifier of the warehouse.

Input Data Definition

Attribute	Data Type	Required	Description
items	List of details	Yes	A list of warehouse items to import.

Detail Import Data Definition

Attribute	Data Type	Required	Description
itemId	String (25)	Yes	The item identifier.
standardUom	String (4)	Yes	The standard unit of measure of the item.
status	Integer	Yes	The status (See Index: Warehouse Item Import Status)

clearInventory	Boolean	Yes	True indicates that the inventory positions should all be set to zero.

Example Input

{
"items": [
{
"itemId": "100000147",
"standardUom": "EA",
"status": 1,
"clearInventory": true
},
{
"itemId": "100000148",
"standardUom": "KG",
"status": 2,
"clearInventory": false
}
]
}

Additional Data Definitions

Warehouse Import Item Status

Value	Definition
1	ACTIVE
2	DISCONTINUED
3	INACTIVE

API: Delete Items

Marks warehouse items for later deletion.

If more than 5000 items are sent in a single call, an input too large error will be returned.

API Basics

Endpoint URL	{base URL}{warehouseId}/delete
Method	POST
Successful Response	202 Accepted

Processing Type	Asynchronous
Input	List of item ids to delete
Output	None
Max Response Limit	5000

Path Parameter Definitions

Attribute	Definition
warehouseId	The internal identifier of the warehouse.

Input Data Definition

Attribute	Data Type	Required	Description
items	List<String(25)>	Yes	A collection of up to 5000 items to remove.

Example Input

```
{
"itemIds": [ "100000301", "100000147" ]
}
```

API: Import Adjustments

API: Import Adjustments

A list of warehouse adjustments is processed, inventory is updated for the warehouse items, and then the adjustments are discarded.

They are not persisted anywhere and this process does not produce a transaction history record.

If more than 5000 items are sent in a single call, an input too large error will be returned.

API Basics

Endpoint URL	{base URL} {warehouseId}/adjustments/import
Method	POST
Successful Response	202 Accepted
Processing Type	Asynchronous (High Volume)
Input	A list of warehouse adjustments to import
Output	None
Max Response Limit	5000

Path Parameter Definitions

Attribute	Definition
warehouseId	The internal identifier of the warehouse.

Input Data Definition

Attribute	Data Type	Required	Description
adjustments	List of details	Yes	A list of adjustments that occurred for that warehouse.

Detail Import Data Definition

Attribute	Data Type	Required	Description
itemId	String (25)	Yes	The unique item identifier.
quantity	BigDecimal	Yes	The quantity to be adjusted.
reasonCode	Integer	Yes	The unique reason code of an inventory adjustment reason code.

Example Input

```
{
"adjustments": [
{
"itemId": "100000147",
"quantity": 100,
"reasonCode": 182
},
{
"itemId": "100000024",
"quantity": 50,
"reasonCode": 183
}
]
}
```

API: Import Inventory

This operation updates the inventory positions of a warehouse.

API Basics

Endpoint URL	{base URL}/{warehouseId}/inventory/import
Method	POST
Successful Response	200 Accepted

Processing Type	Asynchronous
Input	List of items with their inventory
Max Input	10,000 items
Output	N/A
Max Response Limit	N/A

Input Data Definition

Attribute	Type	Definition
Items	List of Warehouse Inventory	A list of items to overwrite inventory for at the warehouse.

Warehouse Inventory Ido

Attribute	Type	Definition
itemId	String(25)	The unique item identifier.
quantityTotal	BigDecimal(12,4)	The total quantity of the item in inventory.
quantityReserved	BigDecimal(12,4)	he quantity reserved for outgoing shipping.
quantityUnavailable	BigDecimal(12,4)	The quantity unavailable for usage.
quantityInTransit	BigDecimal(12,4)	The quantity in transit to the warehouse.

Example Input

```
{
"items
{
"itemId": "100000147",
"quantityTotal”: 100,
"quantityReserved”: 8
},
{
"itemId": "100000024",
"quantityTotal”: 50,
"quantityInTransit”: 183
}
]
}
```

Sales Integration

EICS integrates with POS systems and Sales Audit systems to ensure that the inventory positions are accurate. This is especially important where accurate up-to-date inventory positions are required to reduce customer disappointment when trying to locate items that appear in inventory or delays in filling customer orders.

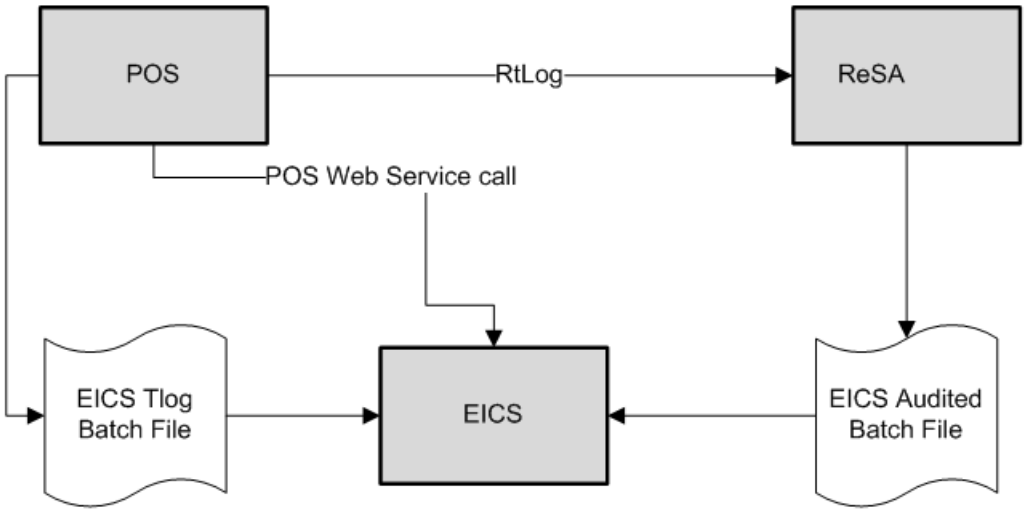
POS is the primary source of sales, returns, void, and some customer order transaction information to EICS.

ReSA sends only modified or new POS transaction records to EICS.

POS systems integrated with EICS can do the transaction notifications using a web service.

Sales Audit systems can only communicate through a file import process.

Figure 7-15 POS and Sales Audit Integration



The following features are part of this integration:

- Real-time web service integration
- Batch integration
- Audited sales data integration
- Automatic disposition processing for returns

Batch processing and ReSA processing are discussed elsewhere as are the store and system configurations that might determine how the sale is processes.

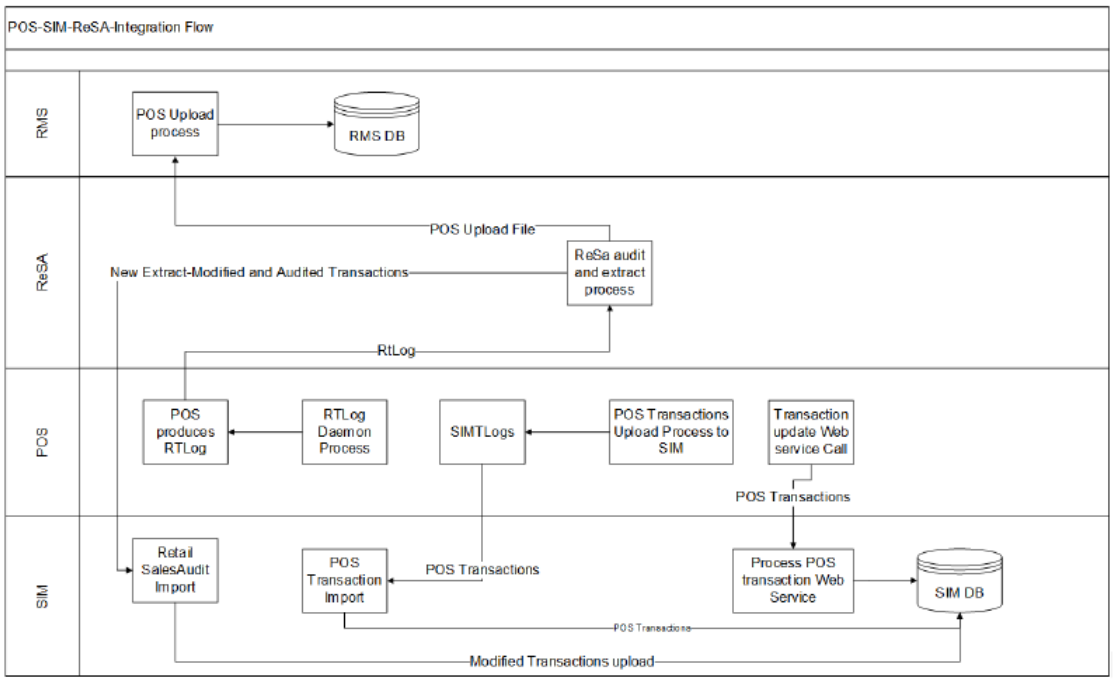
POS and Sales Audit Process Flow

The following figure shows how a POS, Retail Sales Audit, and EICS are integrated. A POS generates an RTLog containing all the POS transactions and sends it to the Oracle Retail Sales Audit system (ReSA). ReSA sends the audited modified or new transactions to EICS. ReSA also sends the POS transaction upload file to merchandising to update inventory.

Please note that Oracle Retail Xstore is interfaced with EICS to update the inventory transactions near real time only through web service. It does not use batch.

Non-Oracle POS systems can use a batch to import transactions directly into EICS. EICS also processes the POS transactions that have been changed or entered into the sales audit system and updates the inventory based on the delta.

Figure 7-16 POS and Sales Audit Process Flow



There are two reasons for POS to send sales data directly to EICS and not to the auditing system:

- Real-time inventory updates to support Commerce Anywhere are critical. A possible round trip from POS to ReSA to EICS takes too long in the dynamic inventory environment of today.
- POS is the application that owns sales data and ReSA owns audited data. Architecturally, it makes more sense to have data supplied by the owner of that data. POS sends sales data and ReSA sends audit changes to EICS.

Sales and Return Processing

As part of the sales processing, EICS updates the inventory depending on the nature of the transaction. The following are the supported transaction types for the sales processing: Sale, Return, and Post Void of these transactions. The audit system should not modify the post void transactions. A change to a void is not supported by EICS.

Customer Order Processing

In EICS, the Retail Sales Audit import process, POS Transaction import process, and POS Transaction web service process support the following types of customer orders.

- For layaway and on hold, EICS supports create, update, cancel, and pickup/delivery. For external web order type, only pickup transactions performed in POS are sent to EICS.
- Pickup transactions, both in-store and external, cannot be voided or modified by sales audit and if these transactions are modified by sales audit system, EICS just drops the transaction and does not process.

> **Note:**
>
> Current Xstore functionality is limited to only layaway and on hold orders. Web order processing is not supported in this release.

Item Disposition

POS can move inventory for return and post void transactions to 'unavailable' or 'out of stock'. This is especially useful in some environments where items returned must be disposed of or must be reprocessed.

The external sale transaction coming into EICS may include a reason code that is mapped to the inventory adjustment reason codes in EICS. Point of Service maps the EICS reason codes, and the reason codes are sent to EICS in the web service or file extract for the return and post void transactions. EICS first processes the return or post void and updates stock on hand. Next, if the reason code exists, EICS checks this reason code with the one in inventory adjustment reason code table. If a valid match is found, EICS generates an inventory adjustment to notify external systems and execute the disposition instructions tied to the inventory adjustment reason code. Based on the disposition mapped to the reason code, EICS moves the returned inventory to not for sale or out of stock and updates the history trail. If sub-buckets are used, they are also updated if the movement is to not for sale.

If the reason code received is invalid/not present/mapped incorrectly, the system writes an error log and continues to process the stock on hand part of the transaction.

Drop Ship

When the sales records indicate the record is a drop ship, EICS does not perform any processing of this record since the drop ship process implies the inventory is shipped from a third-party location and not from the store.

Item Types

EICS only processes SKU or UPC numbers. GS1 databars, or any other smart barcodes such as VPLUs or Type-E barcodes, should have been extracted to their SKU or UPC number by the POS system.

In addition, EICS only updates inventory for stock holding items. Non-inventory items do not update any stock on hand and are not processed.

Items with the store pack inventory indicator turned off are automatically broken down and the inventory of the component items is updated.

RFID

If the point-of-sale record for an item includes an RFID tag, the tag will be moved to a SOLD status indicating it should be out-of-store.

Integration with Customer Order System

CustomerOrderAddressService

When shipping to customer during the fulfillment order workflow, EICS retrieves the address for the order delivery from an external order managements system. When viewing delivery address information within the client application, it also retrieves it from an external system. The web service is defined to connect to an OrderManagementService.

Service Operation	Description
queryCustomerOrderAddress	Retrieves detailed address information for the order and customer information passed to it.

CustomerOrderService

This service connects to OrderManagementService to manage customer orders. It includes operations to create a customer order, query for customer orders, pickup/cancel items from a customer order and return items from customer orders.

Service Operation	Description
requestNewCustomerOrderId	Requests new customer order Id.
cancelNewCustomerOrderId	Cancels the new customer order id.
createCustomerOrder	Creates customer order.
queryCustomerOrder	Queries the customer order present in the system.
PickupCustomerOrderItems	Pickup items from the customer order.
ReturnCustomerOrderItems	Returns items from the customer order.
UpdateReceipt	Updates the receipt of customer order.

Integration with Manifesting Systems

In order for access to an external manifesting system to take place, the customer must first setup Carrier Type as "Third Party" and the Carrier Service (Manifest Type) must be Parcel (P). Configuration controls whether manifesting is done for a transfer to store, finisher, or warehouse. In addition, configuration controls manifesting for a return to vendor shipment or a customer order delivery.

Carrier services with manifest type of "O" (Other) and "H" (Home Fleet) do not go through the manifesting system. When Manifest Type is "O," EICS prompts the user to enter the carrier address where the shipment is to be sent for fulfillment. Manifest Type of "H" is within the company and therefore, does not prompt the user for an address.

Some carriers require weight, dimension, or both values to be sent in the manifest payload. If so, the carrier's service should have either the weight indicator or carton dimension indicate set to active (or both) during their carrier service setup.

EICS supplies an outbound and inbound Shipment Manifest SOAP web service. The following are supported service operations:

A web service is used to send all the shipment information to the external manifesting system and also to receive close shipment requests from external systems.

A web service accepts requests from external systems to close shipments. It is used to find those "Submitted" shipments for the provided tracking ID, carrier, service and date, and dispatch those shipments.

> **Note:**
>
> EICS supplies a WSDL and XSD that defines the web service, operation, and data content. This web service will need to be implemented either for the manifesting system or a plug-in set up.

ShipmentManifestService

This web service notifies an external manifesting system that a manifest needs to be created.

Service Operation	Description
createManifest	Requests the external manifesting system to create a new parcel manifest for an input transaction.

StoreShipmentManifestService

This web service receives a message from an external manifesting system that the items on the manifest have been picked up.

Service Operation	Description
closeManifest	Instructs EICS that submitted shipments have been picked up by the carrier.

Integration for Notifications

StoreExtNotificationService

When store order with external ID is approved, EICS sends notification to the external system.

This service is applicable only for externally created store orders.

Service Operation	Description
createNotification	Sends notification to external system on approving the externally created store orders with its items information.

Integration for Sales Forecast

SalesForecastService

EICS may retrieves item sales forecasting information from a third-party sales forecasting system.

Service Operation	Description
retrieveSalesForecast	Retrieves sales forecast data for the next 30 days for a particular item and store.

Integration for Store Order

OrderApproveNotificationService

When store order is approved, EICS sends notification to a third-party item management system.

This notification will be sent out for store orders that are created manually or system generated.

It is not applicable to store orders created by external system.

Service Operation	Description
orderRequestApproved	Sends notification to external item management system that the order request is approved.

StoreExtNotificationService

When store order with external ID is approved, EICS sends notification to the external system.

This service is applicable only for externally created store orders.

Service Operation	Description
createNotification	Sends notification to external system on approving the externally created store orders with its items information.

Integration for Ticket Printing

When printing tickets, EICS sends ticket information to an external system for printing. This web service needs to be implemented for printing tickets to a physical printer. In the JET administration screen for configuration external service, this endpoint can be configured to connect to either a SOAP or a REST service implemntation.

SOAP Ticket Printing

The details of the SOAP ticket printing endpoint is captured in the associated web service WSDL.

TicketPrintService

Service Operation	Description
printTickets	Sends item tickets to an external system to be printed. It must be implemented by the external system to receive the tickets.

Rest Ticket Printing

This web service defines an endpoint that can be developed by a third party in order to allow EICS to send item ticket printing information to an end system service that handles ticket printing.

The endpoint inputs and outputs must be adhered to by the provider.

API Publish Tickets

This API receives ticket printing information from EICS.

API Basics

Endpoint URL	{base URL}
Method	POST
Success Response	200 OK
Input	Input Print Request For Store and Printer
Output	None

Input Data Definition (Ticket Print Request)

Payload	Type	Definition
storeId	Long(10)	The identifier of the store.
printerName	String(200)	The name of the printer.
printerAddress	String(300)	The URI (or network address) of the printer.
printerId	String(5)	The identifier of the printer.
formatType	Integer(4)	The type of ticket format to print. See Index
formatReference	String(255)	A reference to the format content to use.
templateId	Long(12)	The identifier of a template to use to print the tickets.
zplContent	**String(3500)**	**The content of the ZPL print template.**
tickets	List<Tickets>	A collection of tickets to print.

Tickets

Payload	Type	Definition
ticketId	Long(12)	The identifier of the ticket.
itemId	String(25)	The identifier of the item/sku.
primaryUpc	Strin(25)	The primary Unique Produce Code for the item.
originType	Integer	The origin of the ticket.
sequenceNumber	Integer(3)	The sequence number of the ticket within its grouping.
ticketCount	Integer(3)	The number of instances of this ticket to print.
printQuantity	BigDecimal(12,4)	The quantity to be printed on each ticket.
shortDescription	String(255)	A short description for the item.
longDescription	String(400)	A long description for the item.
shortDescriptionLang	String(255)	A short descripton for the item at the store.
longDescriptionLang	String(400)	A long descripton for the item at the store.
diffType1	String(255)	The description of the first differentiator type.
diffType2	String(255)	The description of the second differentiator type.
diffType3	String(255)	The description of the third differentiator type.
diffType4	String(255)	The description of the fourth differentiator type.

diffDescription1	String(255)	The description of the first differentiator.
diffDescription2	String(255)	The description of the second differentiator.
diffDescription3	String(255)	The description of the third differentiator.
diffDescription4	String(255)	The description of the fourth differentiator.
departmentId	Long(12)	The department identifier of the item.
departmentName	String(360)	The department name.
classId	Long(12)	The class identifier of the item.
className	String(360)	The class name.
subclassId	Long(12)	The subclass identifier of the item.
subclassName	String(360)	The subclass name.
priceCurrency	String(3)	The currency code of the ticket price.
priceValue	BigDecimal(12,4)	The value of the ticket price.
priceType	Integer(3)	The type of the ticket price. See Index.
priceUom	String(4)	The unit of measure of the ticket price.
priceActiveDate	Date	The date the ticket price became active.
priceExpireDate	Date	The date the ticket price expired.
overridePriceCurrency	String(3)	An override price currency code.
overridePriceValue	BigDecimal(20,4)	The amount of an override price.
previousPriceCurrency	String(3)	A previous price currency code.
previousPriceValue	BigDecimal(20,4)	The amount of a previous price.
previousPriceType	Integer(3)	The price type of the previous price. See Index.
lowestMonthlyPriceCurrency	String(3)	The currency code of the lowest monthly price.
lowestMontlyPriceValue	BigDecimal(20,4)	The amount of the lowest monthly price.
lowestMonthlyPriceType	Integer(3)	The price type of the lowest montly price. See Index.
multiUnitPriceCurrency	String(3)	The currency code of the multi-unit price.
multiUnitValue	BigDecimal(20,4)	The amount of the multi-unit price.
multiUnitUom	String(4)	The unit of measure of the multi-unit price.
multiUnitQuantity	BigDecimal(12,4)	The multi-unit quantity associated to the price.
countryOfManufacture	String(3)	The country code of the country of manufacture of the item.
udas	List<TicketUdaExtIdo>	A list of user defined attributes associated to the ticket.

TicketUdaExtIdo

Payload	Type	Definition
name	String(120)	The name of the user defined attribute.

value	String(250)	The value of the user defined attribute.

Additional Data Definitions

Table Format Type

ID	Origin
1	Item Ticket
2	Shelf Label

Ticket Price Type

ID	Origin
1	Permanent
2	Promotional
3	Clearance
4	Clearance Reset

Ticket Origin Type

ID	Origin
1	External
2	Price Change
3	Foundation
4	Manual
5	Promotional Price Change
6	Clearance Price Change
7	Permanent Price Change

Retail Home Integration

EICS now supports following integration scenarios with Retail Home:

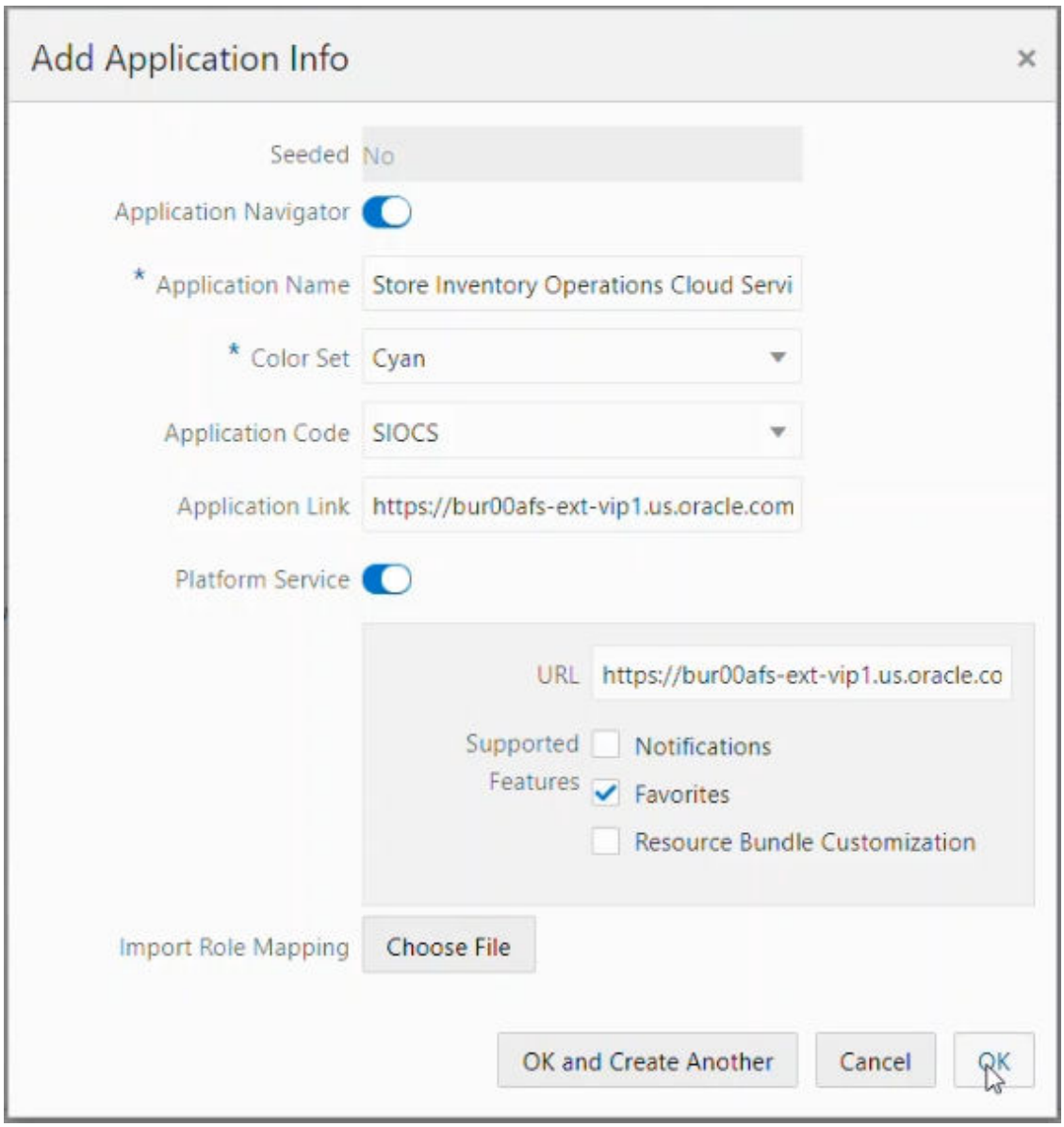
- Launch SIOCS web client from Retail Home
- Launch SIOCS favorites from Retail Home
- Display a tile report for items that are out of stock on shop floor
- Display a tile report for stock counts that are pending authorization
- Launch detailed operational views in SIOCS web client from related tile reports in Retail Home

Launch SIOCS from Retail Home

Launching SIOCS client requires an entry to be made under the application navigator section of Retail Home. It enables the user to launch SIOCS web client in a new browser tab from within Retail Home. Please refer to *Oracle Retail Home Administration Guide* for information on how to work with application navigator in Retail Home.

The SIOCS application configuration should look like this:

Figure 7-17 Add Application Info



- **Seeded**: Disabled and set to No.
- **Application Navigator**: Enable it to launch SIOCS client from Retail Home.
- **Application Name**: The name of the application that is, Store Inventory Operations Cloud.
- **Color Set**: Any color that you want to allocate to SIOCS.
- **Application Code**: Select SIOCS from the drop down.
- **Application Link**: The URL of SIOCS web client.
- **Platform Service**: Enable it to use Favorites feature.
  - **URL**: The base URL of the platform services. The URL would be of the form https://<SIOCS-HOST>/RetailAppsPlatformServices

<SIOCS-HOST> is the same host in Application Link.

- **Supported Features**: Check only the favorites feature.

The user needs to be part of RETAIL_HOME_ADMIN security group in order to access Application Navigator in Retail Home.

Tile Reports

EICS supports following two types of two metric reports:

- Shop Floor Out of Stock Items
- Stock Counts - Ready to Authorize

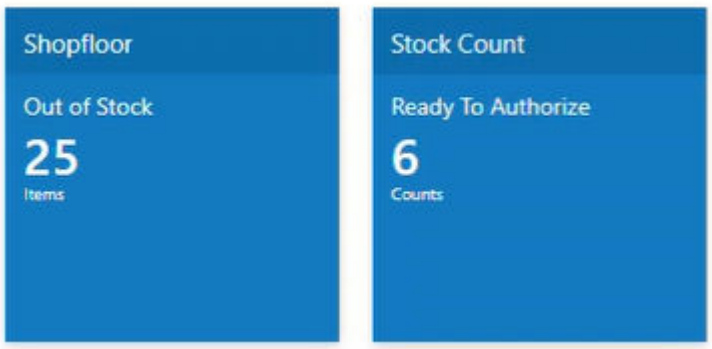
Adding an application navigator entry for SIOCS will automatically configure EICS tiles on Retail Home.

The data seed features do the following:

1. Creates a custom report for EICS tiles on Retail Home.
2. Creates two tiles from the custom report and maps them to retail_home_users IDCS or OCI IAM application role.
3. The data seed features will also configure tile states for the two tiles and hook them up with EICS end points.

After all the configuration, you should be able to see EICS tiles on the dashboard. They should look like the ones below:

Figure 7-18 Example EICS Tiles



EICS Endpoints

EICS exposes following two endpoints:

Shop Floor Out of Stock Items

This endpoint can be used as a data source for **Shop floor Out of Stock** tile state.

The response contains information on number of items that are out of stock across all the stores that are accessible to the user.

If the percentage of out of stock items to total items is greater than the **Shopfloor Out of Stock Items Critical Percentage** system configuration, EICS marks the response as important which displays a '!' mark next to the number on the tile report.

Table 7-117 Shop Floor Out of Stock

Endpoint	Operational View
https://<eics_external_load_balancer_address>/<CUST_ENV>/siocs-client-services/internal/rhreports/outofstock/shopfloor/tile	Shopfloor Out of Stock

Stock Counts - Ready to Authorize

This endpoint can be used as a data source for **Stock Count - Ready to Authorize** tile state.

The response contains information on number of stock counts that are pending authorization across all stores that are accessible to the user.

Table 7-118 Stock Counts - Ready to Authorize

Endpoint	Operational View
https://<eics_external_load_balancer_address>/<CUST_ENV>/siocs-client-services/internal/rhreports/readytoauthorize/tile	Stock Count - Ready To Authorize

The response payloads of both these endpoints confirm to the two metric payload specifications of Retail Home.

User should be a part of retail_home_users IDCS or OCI IAM application role to access these endpoints.

For convenience, EICS also provides a RETAIL HOME security role that captures security permissions required to access these operational views. The user still needs appropriate functional area permissions to navigate to transaction detail screens.

SIOCS Operational Views

EICS has added following operational views that can be hooked with related tiles:

- **Shopfloor Out of Stock Items**

  This view gives a store and item level breakdown of the information that is displayed on the tile. The user can look at item level records for each store and navigate to the item detail screen for any store/item combination provided he or she has the required permissions.

  This view is available under Operations / Operational Views / Shopfloor Out of Stock menu.

- **Stock Count - Ready to Authorize**

  This view gives a store and stock count level breakdown of the information that is displayed on the tile. The user can look at stock count level records for each store and navigate to the stock count detail for any store/count combination provided he or she has the required permissions.

  This view is available under Operations / Operational Views / Stock Count / Ready to Authorize menu.

Launch SIOCS Operational Views from Tile Report

Launching SIOCS operational views from related tile report requires the tile report to be configured with the URL of the related operational view. Once that is done, clicking on tile report header should open the related EICS operational view in a new browser tab.

Subscription Usage Batch

EICS has added a new batch to extract subscription usage for EICS and SOCS respectively during the subscription period. These extracted metrics are pushed to platform services from where Retail Home displays these on the Application Dashboard screen.

This is a restricted batch which by default is scheduled to run every month. The schedule can only be updated by Oracle.

It can also be run as an Adhoc batch from EICS / Admin / Technical Maintenance / Job Admin / Adhoc Job.

REST Web Service OAuth2 Requests

This section will describe how to call an EICS web service using the OAuth2 protocol. The target audience is developers who are looking to write code that calls the web service.

Using the OAuth Protocol

The OAuth protocol is relatively straightforward:

- Get an access token from the authentication provider
- Pass the access token along with the web service request

In this case, the authentication provider is Oracle Identity Cloud Service (IDCS). Every customer who purchases a subscription to EICS gets a subscription to IDCS as part of their purchase.

Obtaining a Token

REST APIs use OAuth2.0 for authorization.

To generate a token from IDCS, an IDCS application client will need to be created for you to use.

The Customer Administration users must create their own client credential IDCS application using the Oracle Retail Home Cloud Service. For additional details, refer to Oracle® Retail Home Administration Guide- Chapter: Oauth Application Configuration chapter – Section: Creating OAuth Client Applications.

The App name and scope that should be used for IDCS application creation should be environment specific using the format:

App Name - RGBU_SIOCS_<ENV>_EICS_INT

Scope - rgbu:siocs:integration-<ENV>

Example:

App Name- RGBU_SIOCS_STG1_ EICS_INT

Scope- rgbu:siocs:integration-STG1

You will need the following information about the IDCS application client to request a token:

- IDCS URL
- Client Id
- Client Secret
- Scope Name

> **Note:**
>
> the application client must be assigned the scope from the EICS IDCS cloud service in order to request the token. This assignment is performed when the application client is created.

The scope name will differ for each environment. Please ensure the correct value is used for your environment.

To generate a token, you will need to invoke the appropriate IDCS REST API. The curl command in Linux that describes the POST that will return a token is as follows:

```
curl -H 'Authorization: Basic <base64(clientId:clientSecret)>' -H 'Content-Type:
application/x-www-form-urlencoded;charset=UTF-8' --request POST <IDCS URL>/
oauth2/v1/token -d 'grant_type=client_credentials&scope=<EICS Scope>'
```

In Windows, use double-quotes, as follows:

```
curl -H "Authorization: Basic <base64(clientId:clientSecret)>" -H "Content-Type:
application/x-www-form-urlencoded;charset=UTF-8" --request POST <IDCS URL>/
oauth2/v1/token -d "grant_type=client_credentials&scope=<EICS Scope>"
```

This is a standard REST POST, with the following details:

- <IDCS URL> is the IDCS URL the retailer provided
- Include the Client Id and Client Secret as a Basic Authentication header
- Specify the Content Type as application/x-www-form-urlencoded;charset=UTF-8
- Specify the body as grant_type=client_credentials&scope=<EICS Scope>

The service will respond with the following JSON message:

{

"access_token": "<TOKEN>",

"token_type": "Bearer",

"expires_in": 3600

}

Note that the response will return how long the token is valid for. You should reuse the same token until it expires in order to minimize calls to IDCS to get a token.

If the token request fails, you will receive the following JSON response:

{

"error":"<error>",

"error_description":"<error description>",

"ecid":"u….."

}

The most common errors are:

- **Invalid Client**. This means that the client information you send in is not correct. The error description will expand on the reason:
  - **Client Authentication Failed** means that the client is valid, but the client secret is incorrect.
  - **Invalid OAuth Client <CLIENT>** means that the client id is not valid, and the invalid client will be listed in the error message.
- **Invalid Request**. Some part of the inbound request is not valid. The error description is usually descriptive about what the actual error condition is

Calling the EICS Web Service

To invoke the web service with an OAuth2 token, you must add an **Authorization** header to the request. The value of the Authorization header must be **Bearer <token>**, that is:

- The word **Bearer**
- A space
- A valid token

For a REST service call, the request might look something like this:

```
curl -X POST -H 'Content-Type: application/json' -H 'Authorization: Bearer
<TOKEN>' -i https://CloudServiceURL --data '{PAYLOAD}'
```

Remember that the token will expire after a specific amount time, and to be more efficient you should always use a token so long as it's valid. It is your responsibility to make sure that you are keeping track of whether the token is still valid. Your pattern should be:

- Check to see if you have a valid token that has not expired.
- If not, call to IDCS and get a new token. Store it and its expiration time.
- Send the request into the web service with the token in the **Authorization** header as a **Bearer** token.

8
Configuration

This chapter describes how you can configure functionality usage.

Configuration can be achieved by adjusting:

1. System Admin Parameters
2. Store Admin Parameters
3. Feature Permissions

System Admin Parameters

Table 8-1 System Admin Parameters

Option	Description	Default Value	Topic	Type
Allow Non-Range Item	Values: Yes/No This parameter is to determine to give stores the ability to add non ranged items to functional areas in the application.	Yes	Admin	Boolean
Allow Item Lookup for Non-Ranged Items	Values: Yes/No Yes:user can look up non-ranged items in item lookup. This is the case even if the system is configured to not allow for non-ranged items, Allow Non-Ranged items = 'No'	Yes	Admin	Boolean
Auto ranging of items for U&A Stock Counts	Values: Allow auto ranging items, Allow auto ranging UINs, Allow Auto ranging items & UINs and Not Allowed. Allow auto ranging items: This setting will allow auto ranging for items but not UINs. Allow auto ranging UINs:This setting will allow auto ranging for UINs but not Items. Allow Auto ranging items & UINs: This allows auto ranging for items and UINs. Only if item is previously ranged, UIN will be allowed to range. Not Allowed : With this setting, the system will allow neither.	Allow Auto ranging items & UINs	Admin	Integer
Barcode Scan/ Entry Log - Receiving	Values: Yes/No Yes: Captures, on the mobile, all container and item scans or manual entries by user, location, and time at the point of receiving deliveries. Containers are captured in the quick receiving dialog when item level is captured in the Item Receiving function of the Container Summary for DSD and Transfer Receiving. No: Does not capture any container or item information in the mobile receiving dialogs.	Yes	Admin	Boolean

Table 8-1 (Cont.) System Admin Parameters

Option	Description	Default Value	Topic	Type
Default UOM	Values: Standard UOM / Cases Standard UOM the system will default to the standard UOM until the UOM is changed. Cases the system will default to cases until the UOM is changed.	2	Admin	Integer
Disable Custom Flexible Attributes	Values: Yes/No Yes: CFAs are disabled on MAF No: CFAs will be available on MAF	Yes	Admin	Boolean
Disable Pack Size	Values: Yes/No Yes: Pack size is disabled and cannot be changed. No: Pack size is editable and can be updated.	No	Admin	Boolean
Enable Sub-buckets	Values: Yes/No Yes: Sub-buckets will be used throughout the application. No: Sub-buckets will not be used in the application.	Yes	Admin	Boolean
Filter Merchandise Hierarchy	Values: Yes/No Yes: Hierarchies / departments will be filtered to those that are for the user's permissions. No: Hierarchies / departments will not be filtered for the user's permissions, all will be available.	No	Admin	Boolean
File Transfer Service Bucket Name	The object storage bucket name for file transfer service.	-	Admin	String
File transfer storage archives prefix	Object storage archives prefix,	Archives	Admin	String
File transfer storage exports prefix	Object storage exports prefix.	Exports	Admin	String
File transfer storage imports prefix	Object storage imports prefix.	Imports	Admin	String
File transfer storage rejects prefix	Object storage rejects prefix.	Rejects	Admin	String
Initial Data Load Display Summary Count	Values: Yes/No Yes: The record count in the Module List table on the Initial Data Load screen is displayed. For large volume tables, loading the count summary might take longer time on loading the UI. No: The record count in the Module List table on the Initial Data Load screen will not be displayed.	No	Admin	Boolean

Table 8-1 (Cont.) System Admin Parameters

Option	Description	Default Value	Topic	Type
Initial Data Load Seed	Values: Yes / No Yes: It indicates that Initial Data Load screen will be used for initial data seeding from MFCS to SIOCS (where they reside in the same pluggable database) and also in case of Standalone SIOCS (through file imports). No: The Initial Data Load screen will not be used for initial data seeding.	No	Admin	Boolean
Initial Data Load Seed Foundation Data	Values: Yes/No Yes: Foundation Data Groups (Item, Miscellaneous, Supplier and Warehouse) will be available for seeding. No: Foundation Data Groups will not be available for seeding.	Yes	Admin	Boolean
Initial Data Load Seed Store Data	Values: Yes/No Yes: Store Data will be available for data seeding. No: Store Data will not be available for data seeding.	Yes	Admin	Boolean
Item Description Translation Preference	Values: System/User System: The Item description displayed in the application would be the item description from STORE_ITEM table (store description) or from ITEM table if the item is not ranged. This is existing functionality. User: The Item description will be displayed in the user's preferred language if translation is available. If the translation is not available in the user's preferred language, then the item description will be displayed in the store locale language if it is available. If the translation is not available in the store locale language, then the item description displayed will be item description from STORE_ITEM (store description). If the item is a non ranged item, then the item description displayed will be from the ITEM table.	System	Admin	Boolean
Maintain RFID History	Values: Yes/No Yes: will create history records in the history table for every transaction occurred for the RFID Tag. No: will not create the history records however the integration with the RFID solution and RFID tag tracking could still be on.	Yes	Admin	Boolean

Table 8-1 (Cont.) System Admin Parameters

Option	Description	Default Value	Topic	Type
Maximum number of Tickets to use synchronous call	This is to determine the integration method with the printing service for the ticket printing based on the number of tickets set. 0 indicates to use the MPS staging process only. Regardless of mobile or desktop, SIOCS will send the ticket to the MPS table for processing. This is needed to keep supporting live customers who have currently only deployed the staged method. >0: If the value set here is greater than zero, the system will do a direct synchronous call to the printer service when the number of tickets is equal or less than the number of tickets set in this parameter. Example: If the value set here is 5 and the number of tickets submitted to print is anything from 1 to 5, the system will do a direct synchronous call to the printer service bypassing the MPS staging process. If the number of tickets printed is above five, it will be MPS staged process. This behavior is regardless of mobile or desktop application.	0	Admin	Integer
Publish available inventory for customer order enabled stores	Parameter to publish the inventory updates (COINVAVAIL message) to external system (OB/OMS) for customer order enabled stores.	No	Admin	Boolean
Server Repave Pending Continue	The batch process checks if the server is about to be repaved for batch unit of work, if this configuration value is set to true, the batch will continue to process next unit of work; if the value is set to false, the batch will skip process the next unit of work, the remaining un-processed works will be marked as stopped and exit the batch process.	Yes	Admin	Boolean
Shopfloor Out of Stock Items Critical Percentage	Values: 0.01 - 100% If the percentage of out of stock items on shop floor to total items is greater than this percentage, then there will be an '!' alert with the number of items out of stock on the tile report.	0.01%	Admin	Integer
Stop Job On Over Max Duration	Internal batch scheduler configuration reserved for future use, currently not used.	No	Admin	Boolean
System Code	Code identifying the application for integration with an external system. This often is a company ID. This is only used for outbound integration.	INV	Admin	String
Audit Direct Store Delivery	Audit Records are log of activities and usage information in the system. This parameter is to determine whether activity records for actions (confirm/submit/update and so on) performed on vendor delivery and vendor delivery carton will be created.	Yes	Audit	Boolean

Table 8-1 (Cont.) System Admin Parameters

Option	Description	Default Value	Topic	Type
Audit Publish Message	Audit Records are log of activities and usage information in the system. This parameter is to determine whether activity records for outbound messages will be created.	Yes	Audit	Boolean
Audit Receive Message	Audit Records are log of activities and usage information in the system. This parameter is to determine whether activity records for inbound messages will be created.	Yes	Audit	Boolean
Audit RTV Update	Values: Yes/No Yes: Enables activity logging for rtv request and rtv shipments. No: Disables activity logging for rtv request and rtv shipments.	Yes	Audit	Boolean
Audit Security	Values: Yes/No Yes: Enables activity logging for security events. It includes login success/failure, security management changes (roles, user assignments, and so on). No: Disables activity logging for security events.	Yes	Audit	Boolean
Audit Stock Count Completed	Audit Records are log of activities and usage information in the system. This parameter is to determine whether activity records will be created for count or recount complete for stock count child.	Yes	Audit	Boolean
Audit Transfer Dispatch	Values: Yes/No Yes: Enables activity logging for transfer shipments. No: Disables activity logging for transfer shipments.	Yes	Audit	Boolean
Audit Transfer Receiving	Values: Yes/No Yes: Enables activity logging for transfer receiving. No: Disables activity logging for transfer receiving.	Yes	Audit	Boolean
Audit Transfer Update	Values: Yes/No Yes: Enables activity logging for transfer requests. No: Disables activity logging for transfer requests.	Yes	Audit	Boolean
Batch max files per job run	Batch max files per job run.	20	Batch	Integer
Batch Maximum Threads	Gives the maximum number of batch partitions for a batch to run.	5	Batch	Integer
Batch Scheduler Execution Interval	Gives the batch scheduler execution delay interval duration to throttle batch execution calls.	300	Batch	Integer

Table 8-1 (Cont.) System Admin Parameters

Option	Description	Default Value	Topic	Type
Compress inventory extract files into zip file	This system parameter indicate whether to compress the inventory extract files into zip. When the compress is true, then the export files will be placed into zip (each zip file will be limited to max 50 MB), multiple zip files maybe generated with the naming conversion as below: zipFileName = filePrefix + "_" + partNum + "_" + <datetime> + ".zip"	No	Batch	Boolean
Days to Hold Archived and Rejected Files	Gives the days to hold rejected and archived batch files before deleting them from the batch directory.	30	Batch	Integer
Initial Data Load Chunk Limit	The commit frequency in number of records during Initial Data Seeding process.	1000	Batch	Integer
Initial Data Load Chunk Log Limit	The number of errors allowed before terminating the Initial Data Seeding process.	15	Batch	Integer
Inventory Extract Omnichannel Store only	Values: Yes/No Yes: Among the SIOCS Managed Stores, the Inventory Extract Batch would consider only the Omnichannel stores to extract the inventory data of the items.	No	Batch	Boolean
Maximum Job Instances Per Scheduler Execution	Gives the maximum number of jobs allowed per run of the scheduler.	100	Batch	Integer
Maximum Processing Hours For Scheduled Job	Determines the maximum processing time for a batch job. After this time, the batch schedule record is updated to failed status.	5	Batch	Integer
Merge Data During Initial Data Load	Values: Yes/No Yes: Data from the Standalone IDLS staging tables will be merged into the SIOCS master tables. No: Data from the Standalone IDLS staging tables will be inserted into the SIOCS master tables.	Yes	Batch	Boolean
Pricing Max Events Per Job Run	Indicating the maximum pricing events to poll from pricing event Integration Change Log (ICL) table per Item Price ICL Import batch run.	5	Batch	Integer
Procedure Log Purging Info Enabled	Determines whether the procedure log purging info is enabled.	Yes	Batch	Boolean
Search Limit Default for Batch Job Days	Gives the default number of days in past for batch job records to be displayed on Batch Job Admin screen on EICS.	7	Batch	Integer
Days to Hold Areas	Values: 0-30 Purge all areas that are greater than or equal today's date minus the days to hold value.	30	Clean Up	Integer

Table 8-1 (Cont.) System Admin Parameters

Option	Description	Default Value	Topic	Type
Days to Hold Audit Records	Values 45-120 Audit Records are log of activities and usage information in the system. This parameter is to determine the number of days to hold the audit records. The batch will delete all records where the create date is less than or equal to current date minus the days to hold.	45	Clean Up	Integer
Days To Hold Batch Logs	Values: 0-30 Delete all logs where the log date is less than or equal to the current date minus the days to hold for any records.	30	Clean Up	Integer
Days to Hold Closed Warehouse Containers	Values : 0-999 This parameter holds the number of days after which the closed warehouse containers and associated deliveries will be deleted.	30	Clean Up	Integer
Days to Hold Completed Inventory Adjustments	Values: 0-120 Delete records in 'Complete' Status where the inventory complete date is less than or equal to the current date minus the days to hold.	120	Clean Up	Integer
Days to Hold Completed Purchase Orders	Values: 0-120 Purge all records in 'Closed' status after 'x' number of days defined by user, where the complete date (the date of when all items were received on the order) is less than or equal to the current date minus the days to hold.	120	Clean Up	Integer
Days to Hold Completed Staging Records	Values: 1-3 Delete all records that have been processed successfully or deleted where the update date is less than or equal to the current date minus the days to hold for any records.	3	Clean Up	Integer
Days to Hold Completed Stock Counts	Values: 0-90 Purges any records 'x' days after the last stock count event has occurred. In other words, when the schedule date is less than or equal to the current date the system will subtract the days to hold completed stock counts from the date and delete when this date is reached. The purging will occur when the stock count has a status of 'Complete'.	90	Clean Up	Integer

Table 8-1 (Cont.) System Admin Parameters

Option	Description	Default Value	Topic	Type
Days to Hold Completed UINs	Values: 0-30 Indicates how long completed UINs are kept in the system. Completed UINs are defined as any UIN that is in one of the following statuses: Sold Shipped to Warehouse Shipped to Vendor Shipped to Finisher Removed from Inventory Customer Fulfilled	30	Clean Up	Integer
Days to Hold Customer Orders	Values: 0-30 Indicates the number of days that Cancelled, and Fulfilled Customer Orders will be held in the system before being purged.	30	Clean Up	Integer
Days to Hold Expired item price	Values: 0-90 Indicates the number of days to hold the expired price changes in the system before being purged.	90	Clean Up	Integer
Days to Hold Expired User Roles	Values: 0-30 This will determine the number of days after which the expired roles will be purged from the system	30	Clean Up	Integer
Days to Hold In Progress Ad Hoc Stock Counts	Values: 0-7 Ad hoc stock counts that are In Progress will be deleted through the purge process. Any ad hoc count with a creation date/time stamp older than this parameter value will be deleted. For example, the default value of 1 would delete all in progress counts more than 24 hours old when the batch is run.	1	Clean Up	Integer
Days to Hold Item Basket	Values: 1-30 This will determine the number of days to hold 'Canceled' and 'Completed' Item Baskets.	30	Clean Up	Integer
Days to Hold Locking Records	Values: 0-3 Locking records will be purged through a batch process and the batch process will account for all locking activity across all functional areas. This is to determine the number of days to hold the locking records. The batch will delete all locking behavior around all functional areas where the lock date is less than or equal to the current business date minus the days to hold.	1	Clean Up	Integer
Days to Hold Notifications	Values: 0-14 This parameter is used to purge notifications which are greater than or equal to this value.	3	Clean Up	Integer

Table 8-1 (Cont.) System Admin Parameters

Option	Description	Default Value	Topic	Type
Days to Hold Price Change Worksheet Records	Values: 0-30 Records in the price change staging / worksheet table will be purged based upon this parameter.	30	Clean Up	Integer
Days to Hold Price History	Values: 0-90 The 'Days to Hold Price History' parameter allows the user to keep records beyond the 4 most recent historical prices for 'x' number of days if desired. Prices in the future will not be deleted and will not be included as part of the four historical prices that will remain on the database.	90	Clean Up	Integer
Days to Hold Received Shipment Records	Values: 0 - 30 Purge all PO and DSD Delivery records in 'Received' and 'Cancelled' status after 'x' number of days defined by user, where the inventory completed date is less than the current date minus the days to hold. There is a receipt record that will be deleted along with any record that is in 'Received' status.	30	Clean Up	Integer
Days to Hold Recently Edited Transactions	Values: 0-15 Purge all Recently Edited transactions on mobile where the post date is less than or equal to the current date minus the days to hold.	7	Clean Up	Integer
Days to Hold Related Items	Values: 0-10 To determine when a related item should be purged. Program will purge Related items that have an end date in the past. The system will purge the related items after 'x' number of days defined by user, where the related items End Date is less than the current date minus the days to hold.	0	Clean Up	Integer
Days to Hold Resolved UIN Exceptions	Values: 0-30 Indicates how long resolved UIN exceptions are kept in the system. The date the exception was resolved is the date the system uses to determine if the exception is ready to be purged.	30	Clean Up	Integer
Days to Hold RFID	Values: 0-7 Indicates how long the RFID data that are not present in the store is kept in the system.	3	Clean Up	Integer
Days to Hold RFID History	Values: 0-120 This parameter will purge RFID history that is greater than or equal today's date minus the days to hold value.	120	Clean Up	Integer

Table 8-1 (Cont.) System Admin Parameters

Option	Description	Default Value	Topic	Type
Days to Hold RTV	Values: 0-90 This parameter will decide that which RTV documents and Shipments need to be purged. The value in this parameter will decide the number of days after a RTV document or shipment gets into cancelled or completed status for document and cancelled or shipped for shipment.	90	Clean Up	Integer
Days to Hold Sales Posting	Values: 1-30 The audit trail for the sales posting will be purged on a periodic basis based on the specified parameter. The system will purge all records from the database after the configurable number of days, where the processed date is less than or equal to current date minus the days to hold	30	Clean Up	Integer
Days to Hold Scan Lists	Values: 0-60 Purge any records in 'Complete' or 'Cancelled' status where the post date is less than or equal to the current date minus the days to hold	60	Clean Up	Integer
Days To Hold SFTP Log File	Indicating number of days to keep the sFTP log files before the log files to be deleted.	7	Clean Up	Integer
Days to Hold Shelf Adjustment Lists	Values: 0-30 Purge any records in 'Complete' or 'Cancelled' status where the post date is less than or equal to the current date minus the days to hold	30	Clean Up	Integer
Days to Hold Shelf Replenishments	Values: 0-3 Purge any records in 'Complete' or 'Cancelled' status where the post date is less than or equal to the current date minus the days to hold.	1	Clean Up	Integer
Days to Hold Store Orders	Values: 0-60 Purge any records in 'Approved' or 'Canceled' status where the post date is less than or equal to the current date minus the days to hold.	60	Clean Up	Integer
Days to Hold Temporary UINs	Values: 0-10 To indicate how long the temporary UINs must stay in the system.	10	Clean Up	Integer
Days to Hold Ticket History	Values: 1-30 To indicate how long the tickets that printed and persisted in the history table must stay.	30	Clean Up	Integer
Days to Hold Transaction History	Values: 0-30 Determines the number of days after which store_item_stock_history records can be purged.	30	Clean Up	Integer
Days to Hold Transfer Documents	Values: 0 -120 This parameter would decide the number of days after which a Transfer document, shipments, and deliveries can be purged.	30	Clean Up	Integer

Table 8-1 (Cont.) System Admin Parameters

Option	Description	Default Value	Topic	Type
Days to Hold UIN Audit Information	Values: 0 -120 Indicates how long UIN audit information is kept in the system. Audit information can be purged for a UIN within the system. The date the audit transaction was captured is used to determine if the record needs to be purged.	120	Clean Up	Integer
Days To Hold Unexecuted Stock Counts	Range: 0 - 90 Gives the number of days after which an unexecuted stock count can be deleted via a batch	30	Cleanup	Integer
Display Item Image For Container Lookup - Operations	Values: Yes/No Yes: Indicates item image will be displayed in Container Lookup on EICS. No: Image will not be displayed in Container Lookup on EICS.	No	Container Lookup	Boolean
Customer Order Fulfillment Restriction	Values: Restricted/Transaction Controlled/Line ControlledTransaction Controlled: The Allow Partial Delivery indicator that comes in on the customer order will be used as it was sent. Restricted: The Allow Partial Delivery Indicator will be updated to 'No' on the Customer Order or Transfer Request upon coming into the system.- Customer Order Deliveries and Transfer Request/Shipment will validate the Allow Partial Delivery indicator as usual, however, it will be set to 'No' and force the user to have a full delivery (except for a user override in customer order deliveries).- Customer Order Picking: When creating a pick, the user will not be able to create the pick if there is not enough available to pick. When confirming a pick, everything must be picked on the customer order.- Reverse Picking: when creating a reverse pick, the user must reverse pick everything that was picked. Line Controlled: If an item is getting delivered, it must be delivered in its entirety- Customer Order Deliveries and Transfer Request/Shipment will validate the Allow Partial Delivery indicator as usual. When Allow Partial Delivery indicator is set to 'Yes', the system will force the user to approve/ship an item fully, if it is getting approved/shipped. When Allow Partial Delivery indicator is 'No', the system will force the user to ship the full order to the customer (except for a user override in customer order deliveries). In case the user has override permission, the customer order can be shipped partially however an item getting shipped should be shipped fully.- Customer Order Picking: When creating a pick, the user will not be able to create the pick if there is not enough available to pick at least one item. When confirming a pick, if an item is getting picked, it must be picked fully.	Transaction Controlled	Customer Order	Integer

Table 8-1 (Cont.) System Admin Parameters

Option	Description	Default Value	Topic	Type
Customer Order Pickup Notification	Values: Yes/No Yes: A new notification alert is generated that a customer is ready to pickup their order. No: A notification is not generated.	No	Customer Order	Boolean
Customer Order Receipt Notification	Values: Yes/No Yes: A receipt notification will be generated when a customer order related transfer receipt or Direct Store Delivery (DSD) has been confirmed. No: The notification will not be generated.	No	Customer Order	Boolean
Customer Order Reauthorization Notification	Values: Yes/No Yes: User will get a notification if a customer order has been reauthorized successfully, that is, when SIOCS successfully consumes and processes a RIB message from OBCS that a customer order has been released from on hold, a notification will be sent to the user. No: User will not get a notification if a customer order has been reauthorized successfully.	No	Customer Order	Boolean
Customer Order Tracking ID Required	Values: Yes/No Yes: If the store parameter 'Manifest Customer Order Deliveries' is set to No, the Tracking ID must be captured before dispatching the Customer Order Delivery. If 'Manifest Customer Order Deliveries' is set to Yes, then it is not required. No: Capturing Tracking ID becomes optional while dispatching the Customer Order Delivery.	No	Customer Order	Boolean
Display Item Image for Customer Order Delivery	Values: Yes/No Yes: This parameter indicates that item image will be displayed in Customer Order Deliveries. No: Images will not be displayed in Customer Order Deliveries	No	Customer Order	Boolean
Display Item Image for Customer Order Picking	Values: Yes/No Yes: This parameter indicates that item image will be displayed in Customer Order Picking. No: Images will not be displayed in Customer Order Picking.	No	Customer Order	Boolean
Display Item Image for Customer Order Reverse Picking	Values: Yes/No Yes: This parameter indicates that item image will be displayed in Customer Order Reverse Picking. No: Images will not be displayed in Customer Order Reverse Picking.	No	Customer Order	Boolean
Display Item Image for Customer Orders	Values: Yes/No Yes: Indicates item image will be displayed in Customer Orders. No: Image will not be displayed in Customer Orders.	No	Customer Order	Boolean

Table 8-1 (Cont.) System Admin Parameters

Option	Description	Default Value	Topic	Type
Minutes To Hold New Customer Order Before Sending Notification	Values: 0-999 This system parameter indicates the time interval in minutes to send a follow-up message to a store associated after a customer order (web order) has arrived, but no user has accessed the customer order.	5	Customer Order	Integer
Minutes To Hold Open Customer Order Pick Before Sending Notification	Values: 0-999 This system parameter dictates the time interval in minutes to send a follow-up message to a store associate after a pick list has been created but no one has started the pick list.	15	Customer Order	Integer
New Customer Order Notification	Values: Yes/No Yes: This parameter generates a notification for a new cross channel (web order) customer order. No: No notification is generated.	No	Customer Order	Boolean
New Customer Order Reverse Pick Notification	Values: Yes/No Yes: A notification alert is generated when a new system generated reverse pick comes into the system. No: A notification is not generated upon getting a new system generated reverse pick.	No	Customer Order	Boolean
Always Send DSD Receipt Cost	Values: Yes/No Yes: When the receipt is published, the unit cost will be sent if there is not an override cost. No: When the receipt is published, only the override cost will be sent if it exists.	No	DSD Receiving	Boolean
Display Unit Cost for Direct Deliveries	Values: Yes/No Yes: Display Unit Cost and allow editing when receiving. If On, the system displays the original cost and allows entering the new cost for the on-the-fly and Dex/Nex deliveries. For the delivery with PO and ASN, it displays the unit cost. Display the Unit Cost on the Direct Delivery Report when printing. No: Do not display this data to the user in the DSD Receiving Containers screen. Do not display the unit cost on the Direct Delivery Report. If No, the system does not display the unit cost and does not allow editing or entering new cost.	Yes	DSD Receiving	Boolean
Displays Item Image for DSD Receiving	Values: Yes/No Yes: This parameter indicates whether the item image will be displayed in Container Items and Item detail screens. No: Image will not be displayed in that functional area.	No	DSD Receiving	Boolean

Table 8-1 (Cont.) System Admin Parameters

Option	Description	Default Value	Topic	Type
Displays Item Image for Purchase Order	Values: Yes/No Yes: This parameter indicates if the item image will be displayed in Purchase Order Items screen. No: Image will not be displayed in that functional area.	No	DSD Receiving	Boolean
DSD Receiving Preferred Currency	Values: Store Currency/Supplier Currency This parameter will default the store or supplier currency to newly created POs depending on preference.	Store Currency	DSD Receiving	Integer
Ignore the Supplier DSD indicator to create a PO on the fly	Values: Yes/No Allows the system to ignore the supplier level indicator when creating a PO in the system. Yes: The system ignores the supplier level flag and will always allow stores to create purchase orders for any supplier based on the receipt. No: The system will verify creating a purchase order on the fly is allowed based on the supplier level flag.	Yes	DSD Receiving	Boolean
Number of days received direct deliveries can be adjusted	Values: 0-999 0: no adjustment 1: allowed to adjust until the end of today 2: allowed to adjust until the end of tomorrow X: allowed to adjust until X number of days starting with today as day 1 This parameter specifies the number of days received direct deliveries can be reopened and adjusted. If a direct delivery falls within the number of days, the system allows to adjust the received delivery. The user will be allowed to edit values and confirm the delivery.	0	DSD Receiving	Integer
Over Received Quantity Notification	Values: Yes/No This parameter will generate a notification when more than the expected quantity has been received and the store parameter Direct Delivery Auto Remove Over Received Quantity is set to yes.	Yes	DSD Receiving	Boolean

Table 8-1 (Cont.) System Admin Parameters

Option	Description	Default Value	Topic	Type
Days Beyond PO Not After Date	This parameter is used to determine the Purchase Orders returned in the deliveries on Item Detail as well as calculating the On Order Qty. 1. Planned Deliveries Ordered Qty needs to be taking the Not After Date into account. Purchase Order – only include those PO's where Not After Date >= Today + Days Beyond PO Not After Date Transfer – only include those transfers where Not After Date >= Today 2. Deliveries – Only display deliveries where: Purchase Order – only include those PO's where Not After Date >= Today + Days Beyond PO Not After Date Transfer – only include those transfers where Not After Date >= Today	50	DSD Receiving	Integer
Display Item Image for Inventory Adjustments - Execution	Values: Yes/No Yes: The item image is displayed within Inventory Adjustments in SOCS. No: The item image is not displayed in Inventory Adjustments.	No	Inventory Adjustment	Boolean
Display Item Image for Inventory Adjustments - Operations	Values: Yes/No Yes: The item image is displayed within Inventory Adjustments in EICS. No: The item image is not displayed in Inventory Adjustments.	No	Inventory Adjustment	Boolean
Display Item Image for Item Baskets - Execution	Values: Yes/No Yes: The item image is displayed within Item Basket on the mobile. No: The item image is not displayed in Item Basket on the mobile.	No	Item Basket	Boolean
Display Item Image for Item Baskets - Operations	Values: Yes/No Yes: The item image is displayed within Item Basket on the desktop. No: The item image is not displayed in Item Basket on the desktop.	No	Item Basket	Boolean
Display Item Image for Item Lookup - Execution	Values: Yes/No Yes: The item image is displayed within Item Lookup on SOCS. No: The item image is not displayed in Item Lookup.	No	Item Lookup	Boolean

Table 8-1 (Cont.) System Admin Parameters

Option	Description	Default Value	Topic	Type
Display Item Image for Item Lookup - Operations	Values: Yes/No Yes: This parameter indicates if the item image will be displayed in Item Lookup in EICS. It is in the list and the details. No: Image will not be displayed in Item Lookup.	No	Item Lookup	Boolean
Display Price in Search Result - Operations	Values: Yes/No This parameter decides whether Price and Pricing Type will be displayed in the search results in the Item Lookup screen in the desktop application. Yes: Price and Price Type will be displayed in the search results. No: Price and Price Type will not be displayed in the search results.	Yes	Item Lookup	Boolean
Display SOH/ Price in Search Result - Execution	Values: Yes/No This parameter decides whether Price and Pricing Type will be displayed in the search results in the Item Lookup screen in the mobile application. Yes: Price and Price Type will be displayed in the search results. No: Price and Price Type will not be displayed in the search results.	Yes	Item Lookup	Boolean
Related Items Group by for Item Lookup - Execution	This parameter is used to determine how to group related items together in item lookup on the mobile. This paramter is used on JET Mobile	Diff 1	Item Lookup	String
Background Thread Count	Gives the thread count for background tasks in SOCS.	1	Mobile	Integer
Barcode Attribute Refresh Rate Milliseconds	Determines the cache refresh rate for barcode attribute labels in milliseconds.	3600000	Mobile	Integer
Configuration Refresh Rate Milliseconds	Determines the cache refresh rate for system configurations, store configurations, batch scheduler and MPS work type in milliseconds.	3600000	Mobile	Integer
Date Output Chinese China	This is to determine the date format based on the locale.	yy-MM-dd	Mobile	String
Date Output Chinese Hong Kong	This is to determine the date format based on the locale.	yy-MM-dd	Mobile	String
Date Output Chinese Taiwan	This is to determine the date format based on the locale.	yy-MM-dd	Mobile	String
Date Output English Australia	This is to determine the date format based on the locale.	dd-MM-yy	Mobile	String
Date Output English Canada	This is to determine the date format based on the locale.	dd-MM-yy	Mobile	String

Table 8-1 (Cont.) System Admin Parameters

Option	Description	Default Value	Topic	Type
Date Output English India	This is to determine the date format based on the locale.	dd-MM-yy	Mobile	String
Date Output English Ireland	This is to determine the date format based on the locale.	dd-MM-yy	Mobile	String
Date Output English South Africa	This is to determine the date format based on the locale.	yy-MM-dd	Mobile	String
Date Output English United Kingdom	This is to determine the date format based on the locale.	dd-MM-yy	Mobile	String
Date Output English United States	This is to determine the date format based on the locale.	MM-dd-yy	Mobile	String
Date Output French Belgium	This is to determine the date format based on the locale.	dd-MM-yy	Mobile	String
Date Output French Canada	This is to determine the date format based on the locale.	yy-MM-dd	Mobile	String
Date Output French France	This is to determine the date format based on the locale.	dd-MM-yy	Mobile	String
Date Output French Luxembourg	This is to determine the date format based on the locale.	dd-MM-yy	Mobile	String
Date Output French Switzerland	This is to determine the date format based on the locale.	dd-MM-yy	Mobile	String
Date Output German Austria	This is to determine the date format based on the locale.	dd-MM-yy	Mobile	String
Date Output German Germany	This is to determine the date format based on the locale.	dd-MM-yy	Mobile	String
Date Output German Luxembourg	This is to determine the date format based on the locale.	dd-MM-yy	Mobile	String
Date Output German Switzerland	This is to determine the date format based on the locale.	dd-MM-yy	Mobile	String
Date Output Italian Italy	This is to determine the date format based on the locale.	dd-MM-yy	Mobile	String
Date Output Japanese Japan	This is to determine the date format based on the locale.	yy-MM-dd	Mobile	String
Date Output Korean South Korea	This is to determine the date format based on the locale.	dd-MM-yy	Mobile	String
Date Output New Zealand	This is to determine the date format based on the locale.	dd-MM-yy	Mobile	String

Table 8-1 (Cont.) System Admin Parameters

Option	Description	Default Value	Topic	Type
Date Output Portuguese Brazil	This is to determine the date format based on the locale.	dd-MM-yy	Mobile	String
Date Output Portuguese Portugal	This is to determine the date format based on the locale.	dd-MM-yy	Mobile	String
Date Output Russian Russia	This is to determine the date format based on the locale.	dd-MM-yy	Mobile	String
Date Output Spanish Argentina	This is to determine the date format based on the locale.	dd-MM-yy	Mobile	String
Date Output Spanish Bolivia	This is to determine the date format based on the locale.	dd-MM-yy	Mobile	String
Date Output Spanish Chile	This is to determine the date format based on the locale.	dd-MM-yy	Mobile	String
Date Output Spanish Columbia	This is to determine the date format based on the locale.	dd-MM-yy	Mobile	String
Date Output Spanish Costa Rica	This is to determine the date format based on the locale.	dd-MM-yy	Mobile	String
Date Output Spanish Dominican Republic	This is to determine the date format based on the locale.	MM-dd-yy	Mobile	String
Date Output Spanish Ecuador	This is to determine the date format based on the locale.	dd-MM-yy	Mobile	String
Date Output Spanish El Salvador	This is to determine the date format based on the locale.	MM-dd-yy	Mobile	String
Date Output Spanish Guatemala	This is to determine the date format based on the locale.	dd-MM-yy	Mobile	String
Date Output Spanish Honduras	This is to determine the date format based on the locale.	MM-dd-yy	Mobile	String
Date Output Spanish Mexico	This is to determine the date format based on the locale.	dd-MM-yy	Mobile	String
Date Output Spanish Nicaragua	This is to determine the date format based on the locale.	MM-dd-yy	Mobile	String
Date Output Spanish Panama	This is to determine the date format based on the locale.	MM-dd-yy	Mobile	String
Date Output Spanish Paraguay	This is to determine the date format based on the locale.	dd-MM-yy	Mobile	String

Table 8-1 (Cont.) System Admin Parameters

Option	Description	Default Value	Topic	Type
Date Output Spanish Peru	This is to determine the date format based on the locale.	dd-MM-yy	Mobile	String
Date Output Spanish Puerto Rico	This is to determine the date format based on the locale.	MM-dd-yy	Mobile	String
Date Output Spanish Spain	This is to determine the date format based on the locale.	dd-MM-yy	Mobile	String
Date Output Spanish Uruguay	This is to determine the date format based on the locale.	dd-MM-yy	Mobile	String
Date Output Spanish Venezuela	This is to determine the date format based on the locale.	dd-MM-yy	Mobile	String
Display Images	Values: Yes/No Yes - The Display Images user preference will be available No - The Display Images user preference will NOT be available.	Yes	Mobile	Boolean
Enable Device Camera Barcode Scan	Used for enabling device camera for scanning on SOCS.	No	Mobile	Boolean
External Scanner Refresh Rate Milliseconds	Determines the cache refresh rate for external scanner in milliseconds.	3600000	Mobile	Integer
Inventory Adjustment Reason Refresh Rate Milliseconds	Determines the cache refresh rate for inventory adjustment reason in milliseconds on SOCS.	3600000	Mobile	Integer
Item Image Refresh Rate Milliseconds	Determines the cache refresh rate for item image in milliseconds on SOCS.	3600000	Mobile	Integer
Manual Quantity Entry Default Mode	Values: Scan Mode/Override Scan Mode: The numeric entry popup on mobile will have its mode defaulted per the scan mode (as it has always done) Override: The numeric entry popup on mobile will have its mode always defaulted to override, and it will not look at the scan mode.	Scan Mode	Mobile	Integer
Non-Sellable Quantity Type Refresh Rate Milliseconds	Determines the cache refresh rate for non-sellable quantity types in milliseconds on SOCS.	3600000	Mobile	Integer
Notification Count Refresh Rate Milliseconds	Determines the cache refresh rate for notifications in milliseconds on SOCS.	300000	Mobile	Integer

Table 8-1 (Cont.) System Admin Parameters

Option	Description	Default Value	Topic	Type
RFID Zone Refresh Rate Milliseconds	Determines the cache refresh rate for RFID zones in milliseconds in the system.	3600000	Mobile	Integer
Scan Focus Item Detail	Determines if sticky focus is enabled on SOCS screens. Yes: Focus will automatically be in the Scan field when on a screen with the scan bar. The system will retain focus in the scan bar field until an error arises or until the user/system moves focus to somewhere else.The keyboard will display on the mobile device while scanning/ entering the scan bar field. This is because the system believes you are typing into the field.No: Focus will not stay in the scan bar.	No	Mobile	Boolean
Sound Error Enabled	Determines if severe error sound will be played in case of severe errors on SOCS.	Yes	Mobile	Boolean
Sound Information Enabled	Determines if information sound effect will be played on SOCS.	Yes	Mobile	Boolean
Sound Scan Enabled	Determines if beep sound will be played on scan on SOCS.	Yes	Mobile	Boolean
Sound Warning Enabled	Determines if a business error sound will be played on business errors on SOCS.	Yes	Mobile	Boolean
Store Printer Refresh Rate Milliseconds	Determines the cache refresh rate for store printer in milliseconds on SOCS.	3600000	Mobile	Integer
Store Refresh Rate Milliseconds	Determines the cache refresh rate for notifications in milliseconds on EICS and SOCS.	3600000	Mobile	Integer
Tablet Mode Screen Size	Determines the screen size for tablet mode for SOCS.	16,5	Mobile	.Double
UOM Conversion Refresh Rate Milliseconds	Determines the cache refresh rate for UOM conversion in milliseconds	3600000	Mobile	Integer
Vibration Enabled	Determines if vibration is enabled on errors on SOCS.	No	Mobile	Boolean
MPS Enabled	Determines if MPS is enabled which in turn determines if MPS work types can be enabled.	Yes	MPS	Boolean
MPS Increment Threads	Determines the allowed thread increment factor for MPS work types.	2	MPS	Integer
MPS Maximum Queue Age Seconds	Determines the maximum seconds before MPS work queue needs to be refreshed.	180	MPS	Integer
MPS Maximum Queue Size	Determines the maximum size limit for generating MPS work queues.	1000	MPS	Integer
MPS Maximum Threads	Determines the maximum thread count for MPS work types.	8	MPS	Integer

Table 8-1 (Cont.) System Admin Parameters

Option	Description	Default Value	Topic	Type
MPS Refresh Rate Seconds	Determines the MPS work queue refresh rate after checking for the system parameter MPS Maximum Queue Age Seconds. If the MPS Maximum Queue Age Seconds has not exceeded, then this parameter is checked to determine if MPS work queue needs to be refreshed.	15	MPS	Integer
Seconds to Check for Notifications	Defines how many seconds the system will check for new notifications. This applies to any notification inserted into the system.	300	Notification	Integer
Days to Hold Areas	Values: 1-999 Purge all areas that are greater than or equal today's date minus the days to hold value.	60	Clean Up	Integer
Days to Hold Audit Records	Audit Records are log of activities and usage information in the system. This parameter is to determine the number of days to hold the audit records. The batch will delete all records where the create date is less than or equal to current date minus the days to hold.	45	Clean Up	Integer
Display Item Image for RFID Locator	Values: Yes/No Yes: This parameter indicates if the item image will be displayed in the RFID Locator dialog in mobile application. No: The image will not be displayed.	No	RFID Locator	Boolean
Days to send Notification before not after date for return requests	Values: 0-999 RTV requests generated in an external system sometimes require the RTV to be dispatched to supplier before a certain date. This option prompts the recipient of the e-mail the specified number of days before the not after date is reached, if the RTV was not dispatched.	2	RTV	Integer
Displays Item Image for RTV Requests	Values: Yes/No Yes: This parameter indicates if the item image will be displayed in that transaction. It is in the item list and the details of the transaction. No: Image will not be displayed in that functional area.	No	RTV	Boolean

Table 8-1 (Cont.) System Admin Parameters

Option	Description	Default Value	Topic	Type
DSD delivery supplier for RTV	Values: Yes/No This indicator will check to see if the DSD allowed indicator needs to be set in addition to the return allowed values when creating a supplier return. Yes: If the “DSD delivery supplier for RTV” system option is set to ‘On’, then the system needs to check both the DSD indicator (Indicator on Supplier table which determines whether a supplier can create a new Purchase Order) and the return allowed indicator (also an indicator on the supplier table). No: If the “DSD delivery supplier for RTV” system option is set to ‘No’, then only the return allowed indicator needs to be validated for supplier returns. Note: Regardless of the indicator, the system should always be able to dispatch the RTV if it was created in an external system.	Yes	RTV	Boolean
RTV Unavailable Request Qty Notification	Values: Yes/No This system parameter will generate notification when “Auto Approve RTV request” parameter is set to On and the request has unavailable quantity greater than the stock.	Yes	RTV	Boolean
Displays Item Image for RTV Shipment	Values: Yes/No Yes: This parameter indicates if the item image will be displayed in that transaction. It is in the item list and the details of the transaction. No: Image will not be displayed in that functional area.	No	RTV Shipment	Boolean
Display Item Image for Replenishment Pick	Values: Yes/No Yes: This parameter indicates if the item image will be displayed in the replenishment pick. It is in the item list and the details of the transaction. No: Image will not be displayed in that functional area.	No	Shelf Replenishment	Boolean
Display Item Image for Scan List	Values: Yes/No Yes: This parameter indicates if the item image will be displayed in the scan list. It is in the item list and the details of the transaction. No: Image will not be displayed in that functional area.	No	Shelf Replenishment	Boolean
Display Item Image for Shelf Adjustment	Values: Yes/No Yes: This parameter indicates if the item image will be displayed in the shelf adjustment. It is in the item list and the details of the transaction. No: Image will not be displayed in that functional area.	No	Shelf Replenishment	Boolean

Table 8-1 (Cont.) System Admin Parameters

Option	Description	Default Value	Topic	Type
Auto Save number of items threshold	This parameter is to determine the number of items after which the system must auto save the items counted. When “Unguided Stock Counts Automatic Save “is set to Yes, SOCS will look at the number of items counted, when it hits the number configured, the system will auto save all the counted values of the user and refresh. If the user saves themselves, the counter of scanned items will be set to 0 again. If the configured value is 1, the system will auto save every item.	1	Stock Counts	Integer
Display Item Image for Stock Counts - Execution	Values: Yes/No Yes: This parameter indicates if the item image will be displayed in the stock counts. It is in the item list and the details of the transaction. No: The image will not be displayed.	No	Stock Counts	Boolean
Display Item Image for Stock Counts - Operations	Values: Yes/No Yes: This parameter indicates if the item image will be displayed in the stock counts functionality in EICS. No: The image will not be displayed.	No	Stock Counts	Boolean
Enable Adhoc Stock Count Locking	This parameter will determine whether a stock count can be locked or not from adding additional item. Yes - User will have the ability to lock an adhoc stock count. i.e., the user will not be able to add additional items after the initial import. No - User will be able to add any number of items until the adhoc stock is completed. Values: Yes/No	No	Stock Counts	Boolean
Stock Count Display Default Timeframe	This parameter is to determine whether the system must prompt the user to select to whether it is performed before store open or after store close.	No	Stock Counts	Boolean
Stock Count Lockout Days	Stock Count Lockout Days is used to determine when a Unit and Amount Stock Count can be generated. The system will take this value plus the system date and enforce a start date of the schedule to be greater than or equal to that date. Note: If the system is integrated with the merchandising system, the values in the two systems must be the same.	1	Stock Counts	Integer

Table 8-1 (Cont.) System Admin Parameters

Option	Description	Default Value	Topic	Type
Stock Count Null Count Quantity = 0	Values: Yes/No Yes: The system changes the quantity to zero for items not counted (null quantity to zero), which makes the items appear as though they were counted. No: The system does not change the quantity to zero for items not counted, but rather leaves the value as null. These items will still appear as though they were not counted. Note This parameter does not apply to Unit and Amount stock counts.	No	Stock Counts	Boolean
Unguided Stock Count Allow Multiple Users	Values: Yes/No This parameter controls whether more than one user can scan simultaneously again the same child stock count for an unguided count. Yes: The system will allow more than one user to access the same stock count, child count. No: The system will allow more than one user to access the same stock count, but only one user may access a child stock count at a time.	No	Stock Counts	Boolean
Unguided Stock Counts Automatic Save	Values: Yes/No Yes: The physical count timestamp and item count quantity are automatically saved when the next item on the count is scanned. No: The physical count timestamp and item count quantity are saved when the user manually saves the count. It is assumed with this option: the user frequently saves. Note: The physical count timestamp is taken when the user scans the item for the first time.	No	Stock Counts	Boolean

Table 8-1 (Cont.) System Admin Parameters

Option	Description	Default Value	Topic	Type
Unit and Amount Stock Count Sales Processing	Values: Timestamp Processing, Daily Sales Processing Timestamp Processing: This option is used when sales data is available near real-time with a date and time available on the transaction. The user is not prompted to select Before Store Open or After Store Close when starting the stock count since the sales timestamp will be used to compare with the timestamps taken during the stock count. Daily Sales Processing: This option is used when sales data is only available with a date and no time is provided and/or when integrated with RMFCS. The user is either prompted or the store parameter determines when the stock count is performed, (before store opens or after store close). The date is used to determine if a sale is late or not. Note: Unit and Amount stock counts require some dual processing in RMFCS for capturing the financial value. RMS is only capable of processing sales data daily and disregards the time value if included.	Timestamp	Stock Counts	Integer
Unit Stock Count Sales Processing	Values: Timestamp Processing, Daily Sales Processing Timestamp Processing: This option is used when sales data is available near real-time with a date and time available on the transaction. The user is not prompted to select Before Store Open or After Store Close when starting the stock count since the sales timestamp will be used to compare with the timestamps taken during the stock count. Daily Sales Processing: This option is used when sales data is only available with a date and no time is provided. The date is used to determine if a sale is late or not.	Timestamp	Stock Counts	Integer
Update Stock On Hand	Values: Yes/No Yes: Will update SOH. No: Will not update SOH.	No	Stock Counts	Boolean

Table 8-1 (Cont.) System Admin Parameters

Option	Description	Default Value	Topic	Type
Update Stock On Hand	Values: All/Discrepant Discrepant Items only: The system will update only items identified as discrepant when the Update Auth Qty button is selected and when the stock count has been authorized, only the SOH is updated for the discrepant items only. All Items: The system will update all items regardless of if they are discrepant or not when the Update Auth Qty button is selected and when the stock count has been authorized, the SOH is updated for all items, including the non-discrepant. Note: Discrepant items are defined as items having a counted to actual variance greater than the pre-configured allowed variance. Non-discrepant items have a difference between the counted and actual qty, but they are within a tolerated variance. Unit and Amount stock counts will disregard this setting since all items will always be updated for that type of stock count.	1	Stock Counts	Integer
Auto Approve Store Orders	Values: Yes/No Yes: external store orders will be auto approved based upon the Days before auto approving Store Orders parameter No: external store orders will not be auto approved.	Yes	Store Order	Boolean
Days to hold before Auto Canceling Store Orders	Values: 0-999 The number of days before setting store orders to canceled status.	0	Store Order	Integer
Default Minimum Store Order Search Quantity	Values: 0–999 Defines the default that will be set for the Minimum ROQ quantity on the search screen. User can override this quantity.	0	Store Order	Integer
Display Item Image Store Orders - Execution	Values: Yes/No Yes: This parameter indicates that item image will be displayed in Store Orders. No: Images will not be displayed in Store Orders.	No	Store Order	Boolean
Default Minimum Store Order Search Quantity	Values: 0–999 Defines the default that will be set for the Minimum ROQ quantity on the search screen. User can override this quantity.	0	Store Order	Integer
Minimum Store Order Quantity	Values: 0–999 Defines the minimum quantity that must be ordered by the user in store orders.	0	Store Order	Integer

Table 8-1 (Cont.) System Admin Parameters

Option	Description	Default Value	Topic	Type
Number of hours after create date in SIOCS to approve store orders	Values: 0-999 This parameter will be used to set external store orders to auto approve store orders on an hourly basis.	0	Store Order	Integer
Store Order Default UOM	Values: SUOM/Cases/System Cases - it will always display cases in store orders. UOM in preferences will be ignored. SUOM - will always display SUOM in store orders. UOM in preferences will be ignored. System - will use the system / preferences UOM in store orders and user can change UOM in preferences and the UI will be updated. This applies to Quick Orders on mobile.	System	Store Order	String
Carrier Service Refresh Rate Milliseconds	Determines the cache refresh rate for carrier service in milliseconds	3600000	System Settings	Integer
Carton Type Refresh Rate Milliseconds	Determines the cache refresh rate for carton type in milliseconds.	3600000	System Settings	Integer
Context Type Refresh Rate Milliseconds	Determines the cache refresh rate for context type in milliseconds	3600000	System Settings	Integer
Database Clock Refresh Rate Milliseconds	Determines the cache refresh rate for database clock in milliseconds.	10800000	System Settings	Integer
Default POS transaction max size	Determines the max size of pos transaction records that can be processed in one pos transaction web service call.	1000	System Settings	Integer
Enable Inbound Integration for Non SIOCS Managed Stores	Values: Yes/No Yes: Inbound integration for stores marked as Non SIOCS Managed will occur when subscribing as well as in batch processing. No: Inbound integration for stores marked as Non SIOCS Managed will NOT occur when subscribing as well as in batch processing. Web services and foundation data are not impacted.	No	System Settings	Integer
Item Image Request Timeout Milliseconds	When loading an item image from an external server, the amount of time before we time out the request.	25000	System Settings	Integer
Merchandise Hierarchy Refresh Rate Milliseconds	Determines the cache refresh rate for merchandise hierarchy in milliseconds.	3600000	System Settings	Integer

Table 8-1 (Cont.) System Admin Parameters

Option	Description	Default Value	Topic	Type
OAuth2 Authorization Cache Refresh Rate Milliseconds	OAuth2 Authorization Cache Refresh Rate Milliseconds.	900000	System Settings	Integer
Price History Refresh Rate Milliseconds	Determines the cache refresh rate for item price history in milliseconds.	3600000	System Settings	Integer
Price default extract size	Determines the default extract size for price change import file for Regular, Promotion and Clearance price change batch jobs.	1000	System Settings	Integer
Print Format Refresh Rate Milliseconds	Determines the cache refresh rate for print format types in milliseconds.	3600000	System Settings	Integer
Publish 3 Character Country Code	Values: Yes, No Yes: The system will publish all the outgoing messages that involves country code with the 3 characters ISO country code. No: The system will publish all the outgoing messages that involves country code with the 2 characters ISO country code.	No	System Settings	Boolean
Publish Non Inventory Items	Values: Yes/No This parameter indicates whether the non-inventory items will be published in the outgoing messages or not.	Yes	System Settings	Boolean
Serialization Label Refresh Rate Milliseconds	Indicates serialization label refresh rate in milliseconds.	3600000	System Settings	Integer
Server Repave Check Refresh Rate Milliseconds	Server Repave Check Refresh Rate Milliseconds.	300000	System Settings	Integer
Server Repave Pending Minimum Minutes	Server Repave Pending Minimum Minutes.	60	System Settings	Integer
Shipment Reason Refresh Rate Milliseconds	Determines the cache refresh rate for finisher shipment reason in milliseconds.	3600000	System Settings	Integer
Store Shipment Reason Refresh Rate Milliseconds	Determines the cache refresh rate for store shipment reason in milliseconds.	3600000	System Settings	Integer
Supplier Refresh Rate Milliseconds	Determines the cache refresh rate for supplier in milliseconds.	3600000	System Settings	Integer

Table 8-1 (Cont.) System Admin Parameters

Option	Description	Default Value	Topic	Type
Supplier Shipment Reason Refresh Rate Milliseconds	Determines the cache refresh rate for supplier shipment reason in milliseconds.	3600000	System Settings	Integer
Translation Refresh Rate Milliseconds	Determines the cache refresh rate for locale and translations in milliseconds.	3600000	System Settings	Integer
Uda Details Refresh Rate Milliseconds	Determines the cache refresh rate for UDA details in milliseconds.	3600000	System Settings	Integer
User Authorization Cache Refresh Rate Milliseconds	Determines the cache refresh rate for user authorization cache in milliseconds.	600000	System Settings	Integer
Warehouse Refresh Rate Milliseconds	Determines the cache refresh rate for warehouse in milliseconds.	3600000	System Settings	Integer
Warehouse Shipment Reason Refresh Rate Milliseconds	Determines the cache refresh rate for warehouse shipment reason in milliseconds.	3600000	System Settings	Integer
Display Item Image for Ticket - Execution	This indicates whether the item image must be displayed in the ticketing dialog in mobile application.	No	Ticketing	Integer
Display Item Image for Ticket - Operations	This indicates whether the item image must be displayed in the ticketing dialog in desktop application.	No	Ticketing	Integer
Daily GMT Batch Run	Values: Yes/No Yes: Program will only be run once a day. The server is running on GMT and the dates for processing are still stored in GMT. The batch will process everything for the GMT day the server is in, including future records for the same day as the server GMT day. This only applies for the Transfer Delivery Auto Receive batch.	Yes	Time Zone	Boolean
Enable GMT for Customer Orders	Values: Yes/No This is to determine whether the customer orders uploaded in the system are in GMT.	No	Time Zone	Boolean
Enable GMT for Dex/Nex	Values: Yes/No This parameter will dictate whether the DEX/NEX data being loaded into the system is in GMT.	No	Time Zone	Boolean

Table 8-1 (Cont.) System Admin Parameters

Option	Description	Default Value	Topic	Type
Enable GMT for Direct Deliveries	Values: Yes/No This parameter will dictate whether the Direct Delivery messages published by an external system should have dates in GMT or not. Yes: When publishing the Direct Delivery messages, it means that the dates in the message should be written in GMT. When subscribing to the purchase order messages, it means that the dates are coming in, in GMT time and no conversion needs to occur. When publishing a purchase order message, it means that the dates in the message should be written in GMT. No: When publishing the direct delivery messages, it means that the dates in the message should be converted from GMT and written in the store's local date/time. When subscribing to the purchase order message, it means that the dates are coming in, in are in the store's local date/time and must be converted to GMT prior to persisting the date in the database. When publishing the purchase order message, it means that the dates in the message should be converted from GMT and written in the store's local date/time.	No	Time Zone	Boolean
Enable GMT for Foundation Data	Values: Yes/No This parameter will dictate whether any foundation data messages being loaded into the system are in GMT.	No	Time Zone	Boolean
Enable GMT for Inventory Adjustments	Values: Yes/No This new system parameter will determine which date/time stamp is used in the inventory adjustment message when it is being published. Yes: When publishing the inventory adjustment message, it means that the dates in the message should be written in GMT. No: When publishing the inventory adjustment message, it means that the dates in the message should be converted from GMT and written in the store's local date/time.	No	Time Zone	Boolean
Enable GMT for POS sale import process	Values: Yes/No This parameter will dictate whether or not the POS data being loaded into the system are in GMT.	No	Time Zone	Boolean

Table 8-1 (Cont.) System Admin Parameters

Option	Description	Default Value	Topic	Type
Enable GMT for Price Changes	Values: Yes/No This parameter will dictate whether the price changes being subscribed to by the system are time zone sensitive. Yes: When subscribing to a price change, it means that the effective date is coming in GMT time and no conversion needs to occur. No: The effective date must be converted prior to storing the price change in the system.	No	Time Zone	Boolean
Enable GMT for RTVs	Values: Yes/No This system parameter will dictate whether the RTV message being loaded into the system is in GMT. Likewise, if the system publishes any RTV message this will determine which date/time stamp is used on the message as well Yes: When subscribing to the RTV message, it means that the dates are coming in, in GMT time and no conversion needs to occur. When publishing the RTV message, it means that the dates in the message should be written in GMT. No: When subscribing to the RTV message, it means that the dates are in the store's local date/time and must be converted to GMT prior to persisting the date in the database. When publishing the RTV message, it means that the dates in the message should be converted from GMT and written in the store's local date/time	No	Time Zone	Boolean
Enable GMT for ReSA sale import process	Values: Yes/No This parameter will dictate whether the ReSA data being loaded into the system are in GMT.	No	Time Zone	Boolean
Enable GMT for Receiving	Values: Yes/No This parameter will dictate whether receiving messages need to be published in GMT or not.	No	Time Zone	Boolean
Enable GMT for Stock Counts	Values: Yes/No This parameter will determine which date/time stamp is used in the stock count message when it is being published. Yes: When publishing the stock count message, it means that the dates in the message should be written in GMT. No: When publishing the stock count message, it means that the dates in the message should be converted from GMT and written in the store's local date/time.	No	Time Zone	Boolean

Table 8-1 (Cont.) System Admin Parameters

Option	Description	Default Value	Topic	Type
Enable GMT for Store Orders	Values: Yes/No This parameter will determine which date/time stamp is used in the store order message when it is being published. Yes: When publishing the store order message, it means that the dates in the message should be written in GMT. No: When publishing the store order message, it means that the dates in the message should be converted from GMT and written in the store's local date/time.	No	Time Zone	Boolean
Enable GMT for Store Transfers	Values: Yes/No This new system parameter will dictate whether the Transfer messages being loaded into the system from an external system has dates in GMT or not. Likewise, if the system publishes any Transfer messages to an external system this will determine which date/time stamp is used on the message as well. Yes: When subscribing to the Transfer messages, it means that the dates are coming in, in GMT time and no conversion needs to occur. When publishing a transfer message, it means that the dates in the message should be written in GMT. No: When subscribing to the transfer message, it means that the dates are coming in, in are in the store's local date/time and must be converted to GMT prior to persisting the date in the database. When publishing the transfer message, it means that the dates in the message should be converted from GMT and written in the store's local date/time.	No	Time Zone	Boolean
Enable GMT for Third Party Stock Counts	Values: Yes/No his parameter will determine whether the date/time stamp in the Third party stock count file (DSLDAT) is in GMT or not.	No	Time Zone	Boolean
Enable GMT for Initial Inventory Import	Values: Yes/No This parameter will determine whether the date/time stamp in the Initial Inventory Import file (.DAT file) is in GMT or not.	No	Time Zone	Boolean
Enable GMT for Vendor ASN	Values: Yes/No This parameter will dictate whether the Vendor ASN messages being loaded into the system have dates in GMT or not.	No	Time Zone	Boolean

Table 8-1 (Cont.) System Admin Parameters

Option	Description	Default Value	Topic	Type
Enable GMT for Warehouse Transfers	Values: Yes/No This new system parameter will dictate whether the transfer messages being loaded into the system have GMT dates or not. Likewise, if the system publishes any transfer message to an external system this will determine which date/time stamp is used on the message as well. Yes: When subscribing to the transfer messages, it means that the dates are coming in, in GMT time and no conversion needs to occur. When retrieving transfer data, it means that the dates are in GMT time and no conversions needs to occur. This includes both reading and writing data. When publishing a transfer message, it means that the dates in the message should be written in GMT. No: When subscribing to the transfer message, it means that the dates are coming in, in are in the store's local date/time and must be converted to GMT prior to persisting the date in the database. When retrieving transfer data, it means that the dates are not in GMT time and must be converted to GMT. This includes both reading and writing data. When publishing the transfer message, it means that the dates in the message should be converted from GMT and written in the store's local date/time.	No	Time Zone	Boolean
Damaged Delivery Notification	Values: Yes/No Yes: Sends a notification to the receiving store when damaged items are received on the delivery. No: No alert is sent. This parameter generates a notification for transfers with items marked as damaged (Warehouse, Store, Finisher).	Yes	Transfer Receiving	Boolean
Days Shipped Delivery Overdue Notification	Values: 1-999 This parameter generates a notification when the delivery is overdue. The delivery will be considered overdue when the create date of the delivery plus the days from this parameter have passed. This will include only deliveries from source type store.	7	Transfer Receiving	Integer
Display Item Image for Transfer Receiving	Values: Yes/No Yes: This parameter indicates if the item image will be displayed in that transaction. It is in the item list and the details of the transaction. No: Image will not be displayed in that functional area.	No	Transfer Receiving	Boolean

Table 8-1 (Cont.) System Admin Parameters

Option	Description	Default Value	Topic	Type
External Finisher UIN Qty Discrep Notification	Values: Yes/No This system parameter will generate notification when there is a discrepancy with the number of UINs on the ASN and the UINs received when auto receiving with a Source Type of 'Finisher'. Yes: Whenever the transaction cannot be auto received, the system generates a notification when there is a discrepancy with the number of UINs on the Finisher Return and the UINs received. Auto Received by batch Auto Received thru RIB Injector. No: No alert is generated.	Yes	Transfer Receiving	Boolean
Misdirected Container Notification	Values: On / No Yes: Sends a notification when a location receives a container belonging to another location. No: No notification is sent. This system parameter will generate a notification when there is a misdirected container that has been received in another location.	Yes	Transfer Receiving	Boolean
Number of Days Received Transfer Deliveries can be Adjusted	Values: 0-999 This parameter controls the number of days a container can be adjusted within a receipt after (Warehouse, Store, Finisher) are received. 0: no adjustment allowed 1: allowed to adjust until the end of today 2: allowed to adjust until the end of tomorrow X: allowed to adjust until x days starting from today	0	Transfer Receiving	Integer
Quick Receiving - Receive misdirected containers	Values: Not Allowed, Automatic, Prompted Not Allowed: Misdirected container cannot be received, no messaging. Automatic: Receives the misdirected container without prompting the user. Prompted: User is prompted to receive the misdirected container.	Not Allowed	Transfer Receiving	Integer
Quick Receiving - Receive missing containers	Values: Yes/No Yes: Enables the ability to receive missing containers. No: Disables the ability to receive missing containers.	Yes	Transfer Receiving	Boolean

Table 8-1 (Cont.) System Admin Parameters

Option	Description	Default Value	Topic	Type
Receive Entire Transfer	Values: Yes/No Yes: User is ONLY allowed to receive the entire delivery. It is not allowed to add any items, only confirmation of the receipt is allowed (Warehouse, Store, Finisher). No: The user is not limited to only receiving the entire delivery.	No	Transfer Receiving	Boolean
Store Receiving Force Close Indicator	Values: RL / SL / NL This parameter applies to deliveries with a Source Type of 'Store'. RL: (Receiver Loss) Any shipped quantity that was not received is a loss at the receiving store. SL: (Sending Loss) Any shipped quantity that was not received is a loss at the sending store. NL: (No Loss) Any shipped quantity that was not received does not affect the receiving or the sending store.	Receiving Loss	Transfer Receiving	Integer
Store Receiving Over/Under Notification	Values: Yes / No This parameter generates a notification to the receiving store when items on a transfer receipt with a Source Type of 'Store' is greater than or less than the expected quantity or if the expected quantity is null or zero. A notification should not be sent in the case of a manually created container or a copied misdirected container. On: Sends a notification when the receiving store over or under receives goods. No: No alert is sent.	Yes	Transfer Receiving	Boolean
Warehouse/ Store UIN Qty Discrep Notification	Values: Yes/No This system parameter will generate notification when there is a discrepancy with the number of UINs on the ASN and the UINs received when auto receiving a warehouse or store delivery with a Source Type of 'Warehouse' or 'Store'. Yes: Sends a notification when there is a discrepancy with UINs on the ASN. The number of UINs on the ASN and the Qty received do not match and cannot be auto received. Auto Received by batch (Store), (Warehouse) Auto Received thru RIB Injector No: No notification is generated.	Yes	Transfer Receiving	Boolean
Display Item Image for Transfer Shipment	Values: Yes/No Yes: This parameter indicates if the item image will be displayed in that transaction. It is in the item list and the details of the transaction. No; Image will not be displayed in that functional area.	No	Transfer Shipment	Boolean

Table 8-1 (Cont.) System Admin Parameters

Option	Description	Default Value	Topic	Type
Days to send Notification before not after date for transfer requests	Values: 0-999 For transfer requests generated in an external system (warehouse, store or finisher), his option sends a notification the specified number of days before the not after date is reached, if the transfer was not dispatched.	2	Transfers	Integer
Display Item Image for Transfer	Values: Yes/No Yes: This parameter indicates if the item image will be displayed in that transaction. It is in the item list and the details of the transaction. No: Image will not be displayed in that functional area.	No	Transfers	Boolean
Transfer Request Approve Notification	Values: Yes/No Yes: A notification will be generated when a requested transfer is approved. No: No notification will be generated. Note: The notification will only be generated for SIOCS initiated store to store requests.	No	Transfers	Boolean
Transfer Request Notification	Values: Yes/No Yes: A notification will be generated when a transfer is requested. No: No notification will be generated. Note: The notification will only be generated for SIOCS initiated store to store requests.	No	Transfers	Boolean
Transfer Request Reject Notification	Values: Yes/No Yes: A notification will be generated when a transfer is rejected. No: No notification will be generated. Note: The notification will only be generated for SIOCS initiated store to store requests.	No	Transfers	Boolean
Unavailable Qty Discrepancy Notification	Values: Yes/No Yes: A notification will be generated when a transfer request fails auto-approval. No: No notification will be generated. Note: Auto Accept External Generated Request is set to On and Unavailable quantity requested is more than the sending store has.	No	Transfers	Boolean
Currency Default Type	Gives the default currency for EICS and SOCS.	USD	UI	String

Table 8-1 (Cont.) System Admin Parameters

Option	Description	Default Value	Topic	Type
Display Item Description	Values: Short Description / Long Description On Mobile the description will be short or long based upon this configuration. Short Item Description: The description displayed everywhere will be the short item description. Note that when integrated with Oracle Retail's merchandising system, the short description of an item is a product of the first 20 characters of the long description. Long Item Description: The description displayed everywhere will be the long item description. Note that when integrated with Oracle Retail's merchandising system, the Merchandising system concatenates the diff descriptions with the long item description, so the user is able to view all diff information. The diffs display in order from Diff 1 to Diff 4. On the desktop the system will always use the long description.	Short Description	UI	Integer
Display Item Description Diffs	Values: Yes/No Yes: A second item description line for diffs will be used wherever an item description is displayed where applicable. No: The second line will not be displayed for the item description and the diffs will not be displayed separately.	Yes	UI	Boolean
Maximum Manual Quantity Entry	Values: 1-100,000,000 The value set here will be the maximum value a user can enter for a quantity via the Numeric Entry on the mobile.	999	UI	Integer
Problem Line UI Limit	Gives the recommended item count in product group component screen for problem line stock count product group.	1500	UI	Integer
Search Date Range Default for Container Lookup	Values: 0-99 This holds the default number of days for which the Container records need to be listed in the Container Lookup screen.	0	UI	Integer
Search Date Range Default for Transaction History	Values: 0-99 This holds the default number of days for which the transaction history records need to be listed in the Transaction History List screen.	0	UI	Integer
Search Limit Default for Area	Values: 1-999 This parameter indicates the default search limit for the Area List screen on EICS.	50	UI	Integer

Table 8-1 (Cont.) System Admin Parameters

Option	Description	Default Value	Topic	Type
Search Limit Default for Container Lookup - Execution	Values: 1-999 Indicates the default search limit for Container Lookup on SOCS.	50	UI	Integer
Search Limit Default for Container Lookup - Operations	Values: 1-999 This parameter will determine the number of records to be displayed on container lookup list screen. The default value on container lookup list screen should be set to the value for the system parameter.	50	UI	Integer
Search Limit Default for Customer Order Picking	Values: 1-999 Indicates the default search limit for Customer Order Picking.	50	UI	Integer
Search Limit Default for Customer Orders	Values: 1-999 Indicates the default search limit for Customer Orders.	50	UI	Integer
Search Limit Default for DSD Receiving	This is to determine the default search limit for DSD receiving list.	50	UI	Integer
Search Limit Default for Finisher Lookup - Operations	Values: 1-999 Indicates the default search limit for Finisher Lookup.	50	UI	Integer
Search Limit Default for Format Assignment - Operations	Values: 1-999 This parameter will determine the number of records to be displayed on the Format Assignment List screen in desktop application. The default value on the search screen should be set to the value from this parameter.	50	UI	Integer
Search Limit Default for Inventory Adjustments - Execution	Values: 1-999 Indicates the default search limit for Inventory Adjustments on SOCS.	50	UI	Integer
Search Limit Default for Inventory Adjustments - Operations	Values: 1-999 Indicates the default search limit for Inventory Adjustments on EICS.	50	UI	Integer
Search Limit Default for Item Baskets - Execution	Values: 0-999 Indicates the default search limit for Item Baskets on SOCS.	50	UI	Integer

Table 8-1 (Cont.) System Admin Parameters

Option	Description	Default Value	Topic	Type
Search Limit Default for Item Baskets - Operations	Values: 0-999 Indicates the default search limit for Item Baskets on EICS.	50	UI	Integer
Search Limit Default for Item Lookup - Execution	Values: 1-999 Indicates the default search limit for Item Lookup on SOCS.	50	UI	Integer
Search Limit Default for Item Lookup - Operations	Values: 1-999 Indicates the default search limit for Item Lookup on EICS.	50	UI	Integer
Search Limit Default for Item Scan Number Lookup	Values: 1-999 This parameter will determine the number of records to be displayed on the Item Scan Number Lookup screen. The default value on the ISN Lookup search screen should be set to the value from this parameter.	500	UI	Integer
Search Limit Default for MPS Staged Messages	Values: 1-999 Indicates the default search limit for MPS staged messages on MPS staged message screen in EICS.	50	UI	Integer
Search Limit Default for Notifications	Values: 1-999 Indicates the default search limit for Notifications.	50	UI	Integer
Search Limit Default for Open Transaction	Values: 1-999 Indicates the default search limit for Open Transactions.	50	UI	Integer
Search Limit Default for Operational Views - Operations	Values: 1-999 This parameter will determine the number of records to be displayed on various operational view screens. The default value on various operational views screens should be set to the value from the system parameter.	50	UI	Integer
Search Limit Default for POS Transaction Resolution	Values: 1-999 This parameter will determine the number of records to be displayed on the POS Transaction resolution dialog in desktop application. The default value on the search screen should be set to the value from this parameter.	50	UI	Integer
Search Limit Default for Purchase Order	This parameter will determine the default number of records to be displayed on PO list screen.	50	UI	Integer

Table 8-1 (Cont.) System Admin Parameters

Option	Description	Default Value	Topic	Type
Search Limit Default for Replenishment Pick	Values: 1-999 Indicates the default search limit for shelf replenishment.	50	UI	Integer
Search Limit Default for RTV	Values: 1-999 Indicates the default search limit for Returns.	50	UI	Integer
Search Limit Default for Scan List	Values: 1-999 Indicates the default search limit for scan lists.	50	UI	Integer
Search Limit Default for Shelf Adjustment	Values: 1-999 Indicates the default search limit for shelf adjustments.	50	UI	Integer
Search Limit Default for Store Order	Values: 1-999 Indicates the default search limit for Store Orders.	50	UI	Integer
Search Limit Default for Stock Count	Values: 1-999 Indicates the default search limit for Stock Counts.	50	UI	Integer
Search Limit Default for Supplier Lookup	Values: 1-999 This parameter will determine the number of records to be displayed on supplier lookup list screen. The default value on supplier lookup list screen should be set to the value for the system parameter.	50	UI	Integer
Search Limit Default for Supplier Lookup - Operations	Values: 1-999 This parameter will determine the number of records to be displayed on supplier lookup list screen. The default value on supplier lookup list screen should be set to the value for the system parameter.	50	UI	Integer
Search Limit Default for Ticket - Operations	Values: 1-999 This parameter will determine the number of records to be displayed on the ticketing dialog in desktop application. The default value on the search screen should be set to the value from this parameter.	50	UI	Integer
Search Limit Default for Transaction History	Values: 1-999 Indicates the default search limit for Transaction History.	50	UI	Integer
Search Limit Default for Transfer Receipts	Values: 1-999 Indicates the default search limit for Transfer receipts.	50	UI	Integer

Table 8-1 (Cont.) System Admin Parameters

Option	Description	Default Value	Topic	Type
Search Limit Default for Transfer Shipment	Values: 1-999 Indicates the default search limit for Transfer shipments.	50	UI	Integer
Search Limit Default for Transfers	Values: 1-999 Indicates the default search limit for Transfer documents.	50	UI	Integer
Search Limit Default for Troubled Transaction List	Values: 1-999 Indicates the default search limit for Troubled Transactions	50	UI	Integer
Search Limit Default for UIN Lookup	Values: 1-999 Indicates the default search limit for UIN Lookup.	50	UI	Integer
Shelf Replenishment UI Limit	Values: 1-99999 Gives recommended item count in product group component screen for shelf replenishment pick product groups.	1500	UI	Integer
Store Order UI Limit	Values: 1-99999 This parameter indicates the UI limit for store orders, used in generation of the store orders. It is also used in the Recommended # of Items in Product Group Components.	1500	UI	Integer
Ticketing UI Limit	Values: 1-99999 This parameter indicates the UI limit for ticketing, used in generation of the auto ticket print. It is also used in the Recommended # of Items in Product Group Components.	1500	UI	Integer
Auto Inventory Adjustment UI Limit	Values: 1-99999 This parameter indicates the UI limit for the auto inventory adjustment, used in the generation of auto inventory adjustment records. It is also used in the recommended # of items in the Product Group components.	1500	UI	Integer
Unit and Amount Count UI Limit	Values: 1-99999 Gives the recommended item count in product group component screen for unit and amount stock count product group.	1500	UI	Integer
Unit Count UI Limit	Values: 1-99999 Gives the recommended item count in product group component screen for unit stock count product group.	5000	UI	Integer
Allow Store UIN Relocation	Values: Yes/No Indicates whether UIN can be relocated from one store to another.	Yes	UIN	Boolean

Store Admin Parameters

Table 8-2 Store Admin Parameters

Options	Description	Default Value	Topic	Type
Display Shopfloor/ Backroom Quantity in Header	Values: Yes/No Yes: This parameter indicates if the shop floor and back room SOH should be displayed in various areas of the system including item lookup as well as transactions. No: Shop floor and back room SOH will not be displayed in various areas of the system.	No	Admin	Boolean
Manifest Weight UOM	Values: List of UOMs from the Weight UOM table The UOM selected for this store admin will be used as the Weight UOM for the weight on the BOL in store to store transfer shipments, customer order deliveries and returns.	LBS	Admin	String
SSCC Shipping Label ID Generation	Values: Yes/No Yes: The system will generate an identifier for printing on the shipping label. No: The user will need to enter an identifier for printing on the shipping label. This store parameter will be used for RTV Shipping and Transfer Shipping.	Yes	Admin	Boolean
UIN Processing Enabled	Values: Yes/No Yes: Enables UIN processing for the store. No: UIN functionality is disabled for the store.	No	Admin	Boolean
Use Extended Attribute Entry	This will turn on and off the feature for editing transaction item level attributes, the capturing of Extended Attributes will be skipped. Values: Yes/No Yes: If set to ‘Yes’, the Extended Attributes can manually be entered, scanned and viewed. This is for mobile and desktop.. No:If set to ‘No’, the Extended Attributes capture screen is not available. User cannot manually enter, view or scan extended attributes on mobile or desktop.	No	Admin	Boolean
Allow Picking By Area	Values: Yes/No Yes: Picking by Area is allowed. The user will be able to select an Area when creating a pick if the Customer Order Fulfillment Restriction is set to 'Transaction Controlled'. The system will narrow down the customer order to those items within an area when creating the pick. No: Picking by Area is not allowed, and the system will always look at all items on the customer order when creating a pick.	Yes	Customer Order	Boolean

Table 8-2 (Cont.) Store Admin Parameters

Options	Description	Default Value	Topic	Type
Auto Pick Mixed Containers	Values: Yes/No Yes: If a container has items in it that are for both customer orders and non-customer orders, the system will auto pick the container. It will mark those items that exist on the customer order as picked. No: If there is a mixed container of customer order and non-customer order items, it will not get auto picked and the picked quantities will not be updated.	No	Customer Order	Boolean
Auto Pick On Receive - Direct Delivery	Values : Yes/No Yes: The system will automatically fill in the pick quantities on the customer order when receiving. This can only happen if the customer order record has already come into the system. If there is no customer order, the auto picking will not happen at the time of receiving, rather it will occur when the customer order comes in. No: The system will not pick when receiving goods.	No	Customer Order	Boolean
Auto Pick On Receive - Transfer Receiving	Values: Yes/No Yes: The system will automatically fill in the pick quantities on the customer order when receiving. This can only happen if the customer order record has already come into the system. If there is no customer order, the auto picking will not happen at the time of receiving, rather it will occur when the customer order comes in. No: The system will not pick when receiving goods in transfer receiving.	No	Customer Order	Boolean
Capture Vehicle Details on Submit	Values: Yes/No Yes: The details regarding the vehicle/driver who is handling the shipment, should be captured before submitting a customer order delivery No: The shipment can be submitted without the details regarding the vehicle/driver	No	Customer Order	Boolean
Default Customer Order Picking Method	Values: Bin / Store Customer Order This parameter is used to define the default picking method when creating a customer order pick, bin or store customer order. Note this is just a default and the user can still switch the picking method.	Store Customer Order	Customer Order	Integer
Default Number of Bins	Values: 1-999 This parameter will determine the number of bins to default into the 'Bin Qty' field if the user selects 'Bin' as the pick type when creating the pick.	1	Customer Order	Integer

Table 8-2 (Cont.) Store Admin Parameters

Options	Description	Default Value	Topic	Type
Dispatch Validate	Values: Ship Direct, Ship Submit Ship Direct: System will control all processes. The user will be able to go from create/edit directly to dispatch. There will not be a submit option. Ship Submit: This option will require the user to press the Submit option and require a specific press of the dispatch button.	Ship Direct	Customer Order	Integer
Holding Location Required	Values: Yes/No Yes - In the Customer Order Quick flow on mobile, the holding location will be required in the Customer Order Pick. No - In the Customer Order Quick flow on mobile, the holding location will not be required in the Customer Order Pick.	No	Customer Order	Boolean
Generate Bins	Values: System / Manual System: The system will automatically generate the bin IDs when the pick is created. Manual: The system will require the user to enter the bin IDs upon acting on the created pick. This option removes the need for printing out labels since the labels already exist on the bin. For both System and Manual, the user will still have the option to print labels for the bins.	Manual	Customer Order	Integer
Item Substitution - Store Discretion	Values: Yes/No This store parameter is used to determine if the user can use their own discretion when doing substitute items in the picking process. Yes: The user can choose any item to be used as a substitute item. No: Only those items that are defined as substitutes can be substituted.	No	Customer Order	Boolean
Navigate to Shipment	Values: Yes/No Yes - In the Customer Order Quick Flow on mobile, for those orders that are shipment type, the user will navigate to the shipment upon completing the pick. No - In the Customer Order Quick Flow on mobile, for those orders that are shipment type, the user will navigate to the Open Transactions screen.	No	Customer Order	Boolean
Override Bin Quantity	Values: Yes/No This store parameter determines whether the user is allowed to override the default bin quantity when creating a pick by bin. The Bin Quantity is defaulted based upon the store parameter for Default Number of Bins.	No	Customer Order	Boolean

Table 8-2 (Cont.) Store Admin Parameters

Options	Description	Default Value	Topic	Type
Picking Required for Customer Orders	Values: Yes/No Yes: Requires that manual picking be performed on the customer order prior to being able to create a delivery for it. No: Picking is not necessary to create a delivery.	Yes	Customer Order	Boolean
Pre-shipment Notification	Values: Yes / No This parameter will drive the following functionality: Yes: The system will publish a pre-shipment message No: The system will not publish a pre-shipment message	No	Customer Order	Boolean
Reserve Customer Order Inventory Upon Receiving	Values: Yes/No This store parameter will dictate when inventory for a web order customer order should be reserved. Yes: Inventory will be reserved upon receiving a delivery in the store which contains the customer order (Store to store transfer, DSD, Warehouse to Store transfer). If the retailer wishes to fulfill the customer orders from deliveries, for example getting the goods from the warehouse, then this option would be set to Yes to reserve upon receiving. No: Inventory will be reserved upon getting the customer order into the store. If the retailer chooses to mainly fulfill customer orders from within the stock in the store, this parameter would be set to No, thus reserving inventory right away when the customer order is received.	No	Customer Order	Boolean
Restrict Shipment Dispatch After Submit	Values: Yes/No . Yes: This restriction will not allow the user to move an Customer Order Deliveries from Submitted to Dispatched status unless the Fiscal Doc ID/E-way Bill ID has been filed in No: The shipment can be dispatched without Fiscal Doc ID/E-way Bill ID	No	Customer Order	Boolean
Allow Multiple Deliveries against PO with No ASN	Values: Yes/No Yes: The user is able to create more than one delivery for the same PO when the PO does not have an associated ASN. No: The user can only create a single delivery against a PO when the PO does not have an associated ASN. The PO will be closed when the delivery is confirmed.	Yes	DSD Receiving	Boolean
Auto close days after expected date	Values: 0-999 Number of days after the expected delivery date the ASN will be closed.	5	DSD Receiving	Integer

Table 8-2 (Cont.) Store Admin Parameters

Options	Description	Default Value	Topic	Type
Direct Delivery Auto Remove Over Received Quantity	Values: Yes/No Yes: If set to Yes, the user is allowed to add any quantity for the DSD, but any quantity above the expected quantity will be removed from the transaction. After the user confirms the transaction, they are prompted that any over received quantities will be removed. The user can create more than one delivery for the same PO when the PO does not have an associated ASN. No: The over received quantities will stay in the delivery transaction.	No	DSD Receiving	Boolean
Direct Delivery Default to ShopFloor Receiving	Values: Yes/No This parameter determines whether the DSD receiving dialog will default to receive inventory into the shop floor instead of automatically receiving into the back room or delivery bay. Yes: The DSD Receiving Container workflow will default the option to receive inventory into the shop floor. The shop floor inventory bucket will be incremented instead of the backroom or delivery bay bucket. If a capacity is defined for the item and Direct Delivery Receive Item Capacity is set to yes, the maximum shop floor quantity will equal the capacity; otherwise, the shop floor will be updated to the entire receipt amount. If the capacity is used and if the receiving quantity is excess, the balance is incremented to delivery bay or back room depending on the Replenishment - Delivery Bay Inventory parameter. No: The option will be defaulted to back room or delivery bay depending on the Replenishment - Delivery Bay Inventory parameter. If this parameter is on, the system will increment the delivery bay bucket instead of back room.	No	DSD Receiving	Boolean
Direct Delivery Invoice Entry	Values: Enabled/Disabled/Unique Enabled: This option allows the user to enter any value for the invoice number, including duplicates. Disabled: The Invoice Number and date fields are disabled. Unique: The Invoice Number field will allow entry; however, the user will not be able to enter a duplicate invoice number based upon the supplier. Upon entering an Invoice Number, the system will validate if the invoice already exists for the supplier defined on the DSD. If a duplicate exists, there will be an error. If no duplicate exists, the invoice number will be accepted.	Enabled	DSD Receiving	Integer

Table 8-2 (Cont.) Store Admin Parameters

Options	Description	Default Value	Topic	Type
Direct Delivery Receive Item Capacity	Values: Yes/No This parameter determines whether the capacity will be considered while receiving the deliveries. Yes: While receiving, the capacity will be considered. For example: If capacity is 50, delivery is for 100, and Available SOH is 10 on the shop floor, if this parameter is on and receive in shop floor is checked, 40 (un-damaged) will be moved to shop floor and the rest to the backroom. No: While receiving, the capacity will not be considered. Damaged inventory will not move to the shop floor.	No	DSD Receiving	Boolean
DSD Receiving Auto Remove Damaged Quantity	Values: Yes/No Yes: All damaged items on the delivery are removed automatically when confirming the transaction. No: All damaged items remain on the delivery when confirming the transaction.	No	DSD Receiving	Boolean
Vehicle Number Required for Transfer Shipment	Values: Yes/No Yes: Capturing of vehicle number of the vehicle transporting the shipment is mandatory before Submitting a transfer shipment No: Capturing of vehicle number of the vehicle transporting the shipment is not mandatory before Submitting a transfer shipment	No	Fiscal Document	Boolean
Vehicle Number Required for RTV Shipment	Values: Yes/No Yes: Capturing of vehicle number of the vehicle transporting the shipment is mandatory before Submitting a RTV shipment No: Capturing of vehicle number of the vehicle transporting the shipment is not mandatory before Submitting a RTV shipment	No	Fiscal Document	Boolean
Vehicle Number Required for Customer Order Delivery	Values: Yes/No Yes: Capturing of vehicle number of the vehicle transporting the shipment is mandatory or not before Submitting a customer order delivery No: Capturing of vehicle number of the vehicle transporting the shipment is mandatory or not before Submitting a customer order delivery	No	Fiscal Document	Boolean
Vehicle State or Country Required for Transfer Shipment	Values: Yes/No Yes: Capturing of vehicle country/state/county of the vehicle transporting the shipment is mandatory before Submitting a transfer shipment No: Capturing of vehicle country/state/county of the vehicle transporting the shipment is not mandatory before Submitting a transfer shipment	No	Fiscal Document	Boolean

Table 8-2 (Cont.) Store Admin Parameters

Options	Description	Default Value	Topic	Type
Vehicle State or Country Required for RTV Shipment	Values: Yes/No Yes: Capturing of vehicle country/state/county of the vehicle transporting the shipment is mandatory before Submitting a RTV shipment No: Capturing of vehicle country/state/county of the vehicle transporting the shipment is not mandatory before Submitting a RTV shipment	No	Fiscal Document	Boolean
Vehicle State orCountry Required for Customer Order Delivery	Values: Yes/No Yes: Capturing of vehicle country/state/county of the vehicle transporting the shipment is mandatory or not before Submitting a customer order delivery No: Capturing of vehicle country/state/county of the vehicle transporting the shipment is mandatory or not before Submitting a customer order delivery	No	Fiscal Document	Boolean
Driver Name Required for Transfer Shipment	Values: Yes/No Yes: Capturing of driver name of the vehicle transporting the shipment is mandatory before Submitting a transfer shipment No: Capturing of driver name of the vehicle transporting the shipment is not mandatory before Submitting a transfer shipment	No	Fiscal Document	Boolean
Driver Name Required for RTV Shipment	Values: Yes/No Yes: Capturing of driver name of the driver transporting the shipment is mandatory before Submitting a RTV shipment No: Capturing of driver name of the driver transporting the shipment is not mandatory before Submitting a RTV shipment	No	Fiscal Document	Boolean
Driver Name Required for Customer Order Delivery	Values: Yes/No Yes Capturing of driver name of the driver transporting the shipment is mandatory or not before Submitting a customer order delivery No: Capturing of driver name of the driver transporting the shipment is mandatory or not before Submitting a customer order delivery	No	Fiscal Document	Boolean
Driver License Number Required for Transfer Shipment	Values: Yes/No Yes: Capturing of driver license number of the driver transporting the shipment is mandatory or not before Submitting a customer order delivery No: Capturing of driver license number of the driver transporting the shipment is mandatory or not before Submitting a customer order delivery	No	Fiscal Document	Boolean

Table 8-2 (Cont.) Store Admin Parameters

Options	Description	Default Value	Topic	Type
Driver License Number Required for RTV Shipment	Values: Yes/No Yes: Capturing of driver license number of the driver transporting the shipment is mandatory before Submitting a RTV shipment No: Capturing of driver license number of the driver transporting the shipment is not mandatory before Submitting a RTV shipment	No	Fiscal Document	Boolean
Driver License Number Required for Customer Order Delivery	Values: Yes/No Yes Capturing of driver license number of the driver transporting the shipment is mandatory or not before Submitting a customer order delivery No: Capturing of driver license number of the driver transporting the shipment is mandatory or not before Submitting a customer order delivery	No	Fiscal Document	Boolean
Context Type required for Inventory Adjustments	Values: Yes/No This configuration decides whether the capturing of context type is mandatory or not for Inventory Adjustments.	No	Inventory Adjustments	Boolean
Days Before Item Basket Expiration	Values: 0-999 This parameter will be used to add to the system date when defaulting the expiration date on an item basket. A value of 0 would set the expiration date to today. A value of 1 will set the expiration date to tomorrow (today + 1).	1	Item Basket	Integer
Auto Accept External Generated RTV Request	Values: Yes/No This parameter determines whether system automatically approves the return request and defaults the requested quantity to the accepted quantity for externally generated RTV requests.	No	RTV	Boolean
Not After Date Default days	Values: 0-999 This parameter would decide the number of days after which a RTV document can be closed After not after date is passed. All status documents would be marked cancelled once this criterion is met.	30	RTV	Integer
Capture Vehicle Details on Submit	Values: Yes/No Yes: The details regarding the vehicle/driver who is handling the shipment, should be captured before submitting an RTV shipment No: The shipment can be submitted without the details regarding the vehicle/driver	No	RTV Shipment	Boolean

Table 8-2 (Cont.) Store Admin Parameters

Options	Description	Default Value	Topic	Type
Dispatch Validate	Values: Ship Direct, Ship Submit Ship Direct: SIOCS will control all processes. The user will be able to go from create/edit directly to dispatch. Ship Submit: This option will require the user to press the Submit button and require a specific press of the dispatch button. An additional option is that an external system will generate a dispatch message through a standard web service.	Ship Direct	RTV Shipment	Integer
Pre-shipment Notification	Values: Yes/No Yes: The system will publish a pre-shipment message. No: The system will not publish a pre-shipment message.	No	RTV Shipment	Boolean
Restrict Shipment Dispatch After Submit	Values: Yes/No Yes: This restriction will not allow the user to move an RTV Shipment from Submitted to Dispatched status unless the Fiscal Doc ID/E-way Bill ID has been filled in. No: The shipment can be dispatched without Fiscal Doc ID/E-way Bill ID	No	RTV Shipment	Boolean
RTV Shipment Carrier Default	Values: Sender / Receiver / Third Party When creating a return, the Carrier Type on the BOL will default initially based upon this parameter. The user can still change this value and if so, that will be the value used on the return. Sender: Sender will be selected for Carrier Type on BOL Receiver: Receiver will be selected for the Carrier type on BOL. Third Party: Third Party will be selected for the Carrier type on the BOL. The type (drop down) will be defaulted to "Other".	Third Party	RTV Shipment	Integer
Display Sequence Fields	Values: Yes/No Yes: Will display sequencing information throughout the application including guided stock count option, capacity, and an item's locations including primary location. No: Sequence information will not be displayed in the system.	No	Sequencing	Boolean
Allow Delivery Bay Quantity to Move to Shop Floor	This is to determine whether the user is allowed to move the delivery bay quantities to shop floor. If allowed, the system will provide an option for the user to select shop floor or back room to move the inventory. If not allowed, it moves the quantity entered to back room. The system will show a pop up to select the option in the Manage Delivery Bay dialog. Values : Yes/No	No	Shelf Replenishment	Boolean

Table 8-2 (Cont.) Store Admin Parameters

Options	Description	Default Value	Topic	Type
Replenishment - At Case Level	Yes: The standard UOM will default to Cases on the shelf replenishment screens. No: The standard UOM will default to Units on the shelf replenishment screens. Note: This parameter will determine whether to replenish shelves at the Case or standard unit of measure. This will override the 'Default UOM' system parameter.	Yes	Shelf Replenishment	Boolean
Replenishment - Delivery Bay Inventory	Values: Yes/No Yes: The delivery bay will be used for replenishment. No: The delivery bay will not be used.	Yes	Shelf Replenishment	Boolean
Replenishment - End of Day max. fill %	This parameter will determine the percentage the stock can fall to before creating the end of day replenishment list.	100	Shelf Replenishment	Double
Replenishment - Item Substitution Store Discretion	Values: Yes/No Yes: The user is allowed to choose any item to substitute. An item lookup feature will allow the user to search for an item to select. No: The user is restricted to scanning/entering an item that exists on the list of approved substitute items defined by the merchandising system. Note: If there are no items defined for item substitution, the dialogue will be displayed with the original item in the header with no substitutes.	No	Shelf Replenishment	Boolean
Replenishment - Within Day Max. fill %	This parameter will determine the percentage the stock can fall to before creating the within day replenishment list.	75	Shelf Replenishment	Double

Table 8-2 (Cont.) Store Admin Parameters

Options	Description	Default Value	Topic	Type
Display Late Inventory Adjustment Message	Yes: When the user is confirming a Stock Count in the Authorization phase and there are items on the stock count with In Progress Inventory Adjustments, the user may return to the stock count to complete the inventory adjustments or to continue and ignore the adjustments. When the user is approving an inventory adjustment with items on an open stock count, a message is displayed allowing the user to determine if further processing should be undertaken. Similar logic to how late sales is processed will be used for these adjustments. No: When the user is confirming a Stock Count in the Authorization phase and there are items on the stock count with In Progress Inventory Adjustments the system ignores the inventory adjustments and allows the user to complete the count. When the user is approving an inventory adjustment, there is no additional processing. Note: The system will process the inventory adjustment like how late sales are processed and determine if the stock count should be adjusted or not. The update to the stock count is not immediate but rather is at the time of authorization if there are reversing entries created.	No	Stock Counts	Boolean
Stock Count Default Timeframe	Values: Before Store Open/After Store Close This parameter defines when the stock count is performed in relation to the store opening hours for Daily Sales Processing. This value may be overridden at the time of the stock count if the system is configured to allow the override. If an override is allowed, this setting will determine the default value displayed. Before Store Open: The stock count is performed before the opening of the store. All sales on the day of the stock count will only update SOH. It will not update any counted quantities. After Store Close: The stock count is performed after the closing of the store. All sales on the day of the stock count will update both SOH and any counted quantities. If using RMS, After Store Close must be selected. Note: Timestamp processing does not use this parameter.	Before Store Open	Stock Counts	Integer
Display Delivery Timeslot	Values: Yes/No Yes: The Delivery Timeslot fields will display throughout Store Orders as well as the Admin screen Delivery Timeslots. No: The Delivery Timeslot fields will NOT display throughout Store Orders as well as the Admin screen Delivery Timeslots.	Yes	Store Order	Boolean

Table 8-2 (Cont.) Store Admin Parameters

Options	Description	Default Value	Topic	Type
DSD Delivery Supplier for Store Order	Values: Yes/No This indicator will check to see if the DSD allowed indicator needs to be yes when adding a supplier restriction when creating a store order. Yes: The system needs to check the DSD indicator (Indicator on Supplier table which determines whether a supplier can create a new Purchase Order) when creating a new Store Order. If the indicator is set to 'Yes' the supplier can be added. If it is set to 'No', the supplier cannot be added. No: The DSD indicator on the supplier does not need to be checked.	Yes	Store Order	Integer
Enable Area for Store Order	Values: Yes/No Yes: The Area will display throughout Store Orders. No: The Area field will NOT display throughout Store Orders.	Yes	Store Order	Boolean
Supplier Restriction for Store Order	Values: Enabled / Required / Disabled Enabled: Supplier will be available as a restriction when creating and searching for a store order. Required: Supplier will be available as a restriction when creating and searching for a store order. When creating it will also be required. Disabled: Supplier will not be available as a restriction when creating a store order.	Enabled	Store Order	Integer
Warehouse Restriction for Store Order	Values: Enabled / Required / Disabled Enabled: Warehouse will be available as a restriction when creating and searching for a store order. Required: Warehouse will be available as a restriction when creating and searching for a store order. When creating it will also be required. Disabled: Warehouse will not be available as a restriction when creating a store order.	Enabled	Store Order	Integer
Auto Generate Item Tickets for Clearance Price Changes	Values: Yes/No Yes: When a clearance price event comes from the pricing system, a new item ticket is sent to the ticketing dialogue. No: When a clearance pricing event comes from the pricing system, the system does not generate an item ticket. This determines whether the system must auto generate item tickets in the system when there is a clearance price event coming in from the pricing system.	No	Ticketing	Boolean

Table 8-2 (Cont.) Store Admin Parameters

Options	Description	Default Value	Topic	Type
Auto Generate Item Tickets for Description Changes	Values: Yes/No Yes: When a new description comes from the merchandising system, a new item ticket is sent to the ticketing dialogue. No: When a new description comes from the merchandising system, the system does not generate an item ticket. This configuration will be used to auto send item tickets to ticketing when an item description is updated and sent to EICS.	No	Ticketing	Boolean
Auto Generate Item Tickets for Promotion Price Changes	Values: Yes/No Yes: When a promotion price event comes from the pricing system, a new item ticket is sent to the ticketing dialogue. No: When a promotion pricing event comes from the pricing system, the system does not generate an item ticket. This determines whether the system must auto generate item tickets in EICS when there is a promotion price event coming in from the pricing system.	No	Ticketing	Boolean
Auto Generate Item Tickets for Regular Price Changes	Values: Yes/No Yes: When a regular price change comes from the pricing system, a new item ticket is sent to the ticketing dialogue. No: When a regular price change comes from the pricing system, the system does not generate an item ticket. This determines whether the system must auto generate item tickets in EICS when there is a regular price change event coming in from the pricing system.	No	Ticketing	Boolean
Auto Generate Shelf Edge Labels for Clearance Price Changes	Values: Yes/No Yes: When a clearance price event comes from the pricing system, a shelf edge label is sent to the ticketing dialogue. No: When a clearance pricing event comes from the pricing system, the system does not generate a label. This determines whether the system must auto generate item tickets in the system when there is a clearance price event coming in from the pricing system.	No	Ticketing	Boolean

Table 8-2 (Cont.) Store Admin Parameters

Options	Description	Default Value	Topic	Type
Auto Generate Shelf Edge Labels for Description Changes	Values: Yes/No Yes: When a new description comes from the merchandising system, a shelf edge label is sent to the ticketing dialogue. No: When a new description comes from the merchandising system, the system does not generate a label. This configuration will be used to auto generate labels when an item description is updated and to send to EICS.	No	Ticketing	Boolean
Auto Generate Shelf Edge Labels for Promotion Price Changes	Values: Yes/No Yes: When a promotion price event comes from the pricing system, a new shelf edge label is sent to the ticketing dialogue. No: When a promotion pricing event comes from the pricing system, the system does not generate a label. This determines whether the system must auto generate labels in EICS when there is a promotion price event coming in from the pricing system.	No	Ticketing	Boolean
Auto Generate Shelf Edge Labels for Regular Price Changes	Values: Yes/No Yes: When a regular price change comes from the pricing system, a new shelf edge label is sent to the ticketing dialogue. No: When a regular price change comes from the pricing system, the system does not generate a label. This determines whether the system must auto generate labels in EICS when there is a regular price change event coming in from the pricing system.	No	Ticketing	Boolean
Auto Ticket Generate Future Days	Values: 0 – 99 This parameter indicates the number of days the system must consider for future day events for generating tickets when the batch is run. If it is set to 0, it means the system will not consider the future events. If the value is above zero, the system will consider the price events that are falling in the range of current date plus the number of days set in this parameter to generate the tickets.	0	Ticketing	Integer

Table 8-2 (Cont.) Store Admin Parameters

Options	Description	Default Value	Topic	Type
Item Print Events	This is to determine the default item price events for the ticket printing. Values: Always, Clearance, Promotion, Permanent, Clearance or Permanent and Any Price Event Always: This option will always print a ticket regardless of if there is a price change. Clearance: Only print a ticket if on the specific date any clearance event is effective. So, for Clearance 2, that means today's date + 2 days, if the item on that day has a clearance going on, print the clearance ticket. Promotion: Only print a ticket if on the specific date any promotion event is effective. Regular or Clearance: Only print a ticket if on the specific date any Regular or Clearance event is getting effective. Permanent: Only print a ticket if on the specific date any Regular event of getting effective. Any Price Event (Promotion, Clearance or Permanent (Regular)): Based on the date selected, if any price event goes into effect (clearance, promotion or regular price a ticket. If no price event goes into effect, print nothing).	Always	Ticketing	String
Maximum Ticket Quantity to Print	This is to determine the maximum ticket size to print in one command. This is used in auto ticket printing batch and ticketing dialogue.	500	Ticketing	Boolean
Auto Close Receipt	Values: 1-99 0: close the receipt immediately 1: close the receipt the end of day today 2: close the receipt end of day tomorrow X: close end of day x days starting from today the batch program will auto close any transfer receipts and marks all non-received containers to missing. Partially received containers will be marked as damaged.	1	Transfer Receiving	Integer
External Finisher Auto Receive	Values: Not Allowed, External Message, Date Driven Not allowed will make the system work as today. External message will receive the full external finisher delivery with a Source Type of 'Finisher', the moment an ASN transaction arrives that indicates that the delivery needs to be auto received. Date Driven will look at a secondary store option (External Finisher Auto Receive number of Days) to determine how many days the transaction stays open before it is fully received. If it is set to 0, it will auto-receive on the ETA date.	Not Allowed	Transfer Receiving	Integer

Table 8-2 (Cont.) Store Admin Parameters

Options	Description	Default Value	Topic	Type
External Finisher Auto Receive Number of Days	Values: 0-999 A batch program will auto receive any external finisher deliveries with a Source Type of 'Finisher', that have not been closed x-days after the ETA date or the create date depending on if the ETA date is set or not and if the auto receive external delivery parameter is set 0 means immediate receiving 1 means today (EOD) 2 means EOD tomorrow x means EOD x days starting from today	0	Transfer Receiving	Integer
Store Auto Receive	Values: Not allowed / External message / Date Driven Not allowed: Auto receiving is not allowed for the store. External message: Receives the full store delivery the moment an ASN transaction arrives when the indicator on the ASN identifies this as an auto receive delivery and the Source Type is 'Store'. This parameter works with the Store Auto Receive screen. Date Driven: Receives the delivery automatically when the date is reached. A second options, 'Store Auto Receive Number of Days' is used to determine how many days the transaction stays open before it is fully received. If it is set to 0, it will receive immediately when the transfer is shipped. This parameter works with the Store Auto Receive screen.	Not Allowed	Transfer Receiving	Integer
Store Auto Receive Number of Days	Values: 0-99 0: immediate receiving 1: end of day today 2: end of day tomorrow X: end of day x days starting from today The batch program will auto receive any transfers with a Source Type of 'Store' not previously closed x-days after they have been shipped. This parameter is only used when the Store Auto Receive parameter is enabled.	0	Transfer Receiving	Integer

Table 8-2 (Cont.) Store Admin Parameters

Options	Description	Default Value	Topic	Type
Store Transfer Default to ShopFloor Receive	Values: Yes/No This parameter determines whether the receiving will default to receive inventory into the shop floor instead of automatically receiving into the back room or delivery bay when the source type is 'Store'. Yes: The Transfer workflow will default to receive inventory into the shop floor when the Source Type is 'Store'. The shop floor inventory bucket will be incremented instead of the backroom or delivery bay bucket. If a capacity is defined for the item, the maximum shop floor quantity will equal the capacity; otherwise, the shop floor will be updated to the entire receipt amount. If the capacity is used and if the receiving quantity is excess, the balance is incremented to delivery bay or back room depending on the Replenishment. No: This will default to back room or delivery bay depending on the Replenishment -Delivery Bay Inventory parameter and all inventory will be automatically received into the back room or delivery bay when Source Type is 'Store'. Note: With the proper permissions, the user will still have the option to receive onto the shop floor while receiving the delivery by changing the default to Shop Floor. Damaged inventory will not move to shop floor.	No	Transfer Receiving	Boolean
Store Transfer Receive Item Capacity	Values: Yes/No This parameter will determine whether the capacity will be considered while receiving the deliveries. Yes: If the value is set to YES, then while receiving and Source Type of 'Store', the capacity will be considered. For example: If capacity is 50, receipt is for 100 and Available SOH is 10 on shopfloor, then if this parameter is on and receive in shop floor is checked then 40 (un-damaged) will be moved to shop floor and rest to back room. No: If the value is set to NO, then while receiving the capacity will not be considered. Damaged inventory will not move to shop floor.	No	Transfer Receiving	Boolean

Table 8-2 (Cont.) Store Admin Parameters

Options	Description	Default Value	Topic	Type
Warehouse Auto Receive	Values: Not Allowed, External Message, Date Driven This parameter will drive the following functionality. Not Allowed will make the system work as today. External message will receive the full warehouse delivery the moment an ASN transaction arrives that indicates that the delivery needs to be auto received and the Source Type is 'Warehouse'. Date Driven will look at a secondary store option (Warehouse Auto Receive number of Days) to determine how many days the transaction stays open before it is fully received. If it is set to 0, it will auto-receive on the ETA date.	Not Allowed	Transfer Receiving	Integer
Warehouse Auto Receive Number of Days	Values: 0-99 0: immediate receiving 1: end of day today 2: end of day tomorrow X: end of day x days starting from today The batch program will auto receive any transfers with a Source Type of 'Warehouse' not previously closed x-days after they have been shipped. This parameter is only used when the Warehouse Auto Receive parameter is enabled.	0	Transfer Receiving	Integer

Table 8-2 (Cont.) Store Admin Parameters

Options	Description	Default Value	Topic	Type
Warehouse Default to ShopFloor Receive	Values: Yes/No This parameter determines whether the receiving will default to receive inventory into the shop floor when source type is 'Warehouse' instead of automatically receiving into the back room or delivery bay when the source type is 'Warehouse'. Yes: The Transfer receiving workflow will default to receive inventory into the shop floor when the Source Type is 'Warehouse'. The shop floor inventory bucket will be incremented instead of the backroom or delivery bay bucket. If a capacity is defined for the item, the maximum shop floor quantity will equal the capacity; otherwise, the shop floor will be updated to the entire receipt amount. If the capacity is used and if the receiving quantity is excess, the balance is incremented to delivery bay or back room depending on the Replenishment - Delivery Bay Inventory parameter. No: This will default to back room or delivery bay depending on the Replenishment -Delivery Bay Inventory parameter and all inventory will be automatically received into the back room or delivery bay when Source Type is 'Warehouse'. Note: With the proper permissions, the user will still have the option to receive onto the shop floor while receiving the delivery by changing the default to Shop Floor. Damaged inventory will not move to shop floor.	No	Transfer Receiving	Boolean
Warehouse Receive Item Capacity	Values: Yes/No This parameter will determine whether the capacity will be considered while receiving the deliveries. Yes: If the value is set to YES, then while receiving and Source Type of 'Warehouse', the capacity will be considered. For example: If capacity is 50, receipt is for 100 and Available SOH is 10 on shopfloor, then if this parameter is on and receive in shop floor is checked then 40 (un-damaged) will be moved to shop floor and rest to back room. No: If the value is set to NO, then while receiving the capacity will not be considered. Damaged inventory will not move to shop floor	No	Transfer Receiving	Boolean

Table 8-2 (Cont.) Store Admin Parameters

Options	Description	Default Value	Topic	Type
Dispatch Validate	Values: Ship Direct, Ship Submit Ship Direct: SIOCS will control all processes. The user will be able to go from create/edit directly to dispatch. Ship Submit: This option will require the user to press the Submit button and require a specific press of the dispatch button. An additional option is that an external system will generate a dispatch message through a standard web service.	Ship Direct	Transfer Receiving	Integer
Capture Vehicle Details on Submit	Values: Yes/No Yes: The details regarding the vehicle/driver who is handling the shipment, should be captured before submitting a transfer shipment No: The shipment can be submitted without the details regarding the vehicle/driver	No	Transfer Shipment	Boolean
Pre-shipment Notification	Values: Yes/No This parameter will drive the following functionality: Yes: The system will publish a pre-shipment message. No: The system will not publish a pre-shipment message.	No	Transfer Shipment	Boolean
Restrict Shipment Dispatch After Submit	Values: Yes/No Yes: This restriction will not allow the user to move a transfer Shipment from Submitted to Dispatched status unless the Fiscal Doc ID/E-way Bill ID has been filed in. No: The shipment can be dispatched without Fiscal Doc ID/E-way Bill ID	No	Transfer Shipment	Boolean
Ship to Finisher Carrier Default	Values: Sender / Receiver / Third Party Sender: Sender will be selected for Carrier Type on BOL Receiver: Receiver will be selected for the Carrier type on BOL. Third Party: Third Party will be selected for the Carrier type on the BOL. The type (drop down) will be defaulted to "Other". When creating a store to Finisher transfer the Carrier Type on the BOL will default initially based upon this parameter. The user can still change this value and if so, that will be the value used on the transfer.	Third Party	Transfer Shipment	Integer

Table 8-2 (Cont.) Store Admin Parameters

Options	Description	Default Value	Topic	Type
Ship to Store Carrier Default	Values: Sender / Receiver / Third Party Sender: Sender will be selected for Carrier Type on BOL Receiver: Receiver will be selected for the Carrier type on BOL. Third Party: Third Party will be selected for the Carrier type on the BOL. The type (drop down) will be defaulted to "Other". When creating a store to store transfer the Carrier Type on the BOL will default initially based upon this parameter. The user can still change this value and if so, that will be the value used on the transfer.	Third Party	Transfer Shipment	Integer
Ship to Warehouse Carrier Default	Values: Sender / Receiver / Third Party Sender: Sender will be selected for Carrier Type on BOL Receiver: Receiver will be selected for the Carrier type on BOL. Third Party: Third Party will be selected for the Carrier type on the BOL. The type (drop down) will be defaulted to "Other". When creating a store to WH transfer the Carrier Type on the BOL will default initially based upon this parameter. The user can still change this value and if so, that will be the value used on the transfer.	Third Party	Transfer Shipment	Integer
Auto Accept External Generated Request	Values: Yes/No This parameter automatically approves the requested transfer and defaults the requested quantity to the accepted quantity for externally generated requests.	No	Transfers	Boolean
Auto Accept Store Transfer Request	Values: Yes/No This parameter automatically approves the requested transfer and defaults the requested quantity to the accepted quantity for store to store requests.	No	Transfers	Boolean
Context Type/ Value required for Transfer	Values: Yes/No Yes: Capturing of context type and value is mandatory before requesting a transfer No: Capturing of context type and value is not mandatory before requesting a transfer	No	Transfers	Boolean
Not After Date Default Days	Values: 0-999 This parameter adds a value to the current date and uses the value to default the 'Not After Date' when creating a transfer or requesting a transfer. Documents will be closed via batch when this date is reached.	30	Transfers	Integer

Table 8-2 (Cont.) Store Admin Parameters

Options	Description	Default Value	Topic	Type
Manifest Customer Order Deliveries	Values: Yes/No Yes: The Manifesting system will be called. No: The Manifesting system will not be called.	No	Web Service Enablement	Boolean
Manifest RTV to Supplier	Values: Yes/No Yes: The Manifesting system will be called for return to supplier. No: The Manifesting system will not be called. Note: The interface will still need to be implemented; this just is to determine if it will be called.	No	Web Service Enablement	Boolean
Manifest Transfer to Finisher	Values: Yes/No Yes: The Manifesting system will be called for transfer to Finisher. No: The Manifesting system will not be called. Note: The interface will still need to be implemented; this just is to determine if it will be called.	No	Web Service Enablement	Boolean
Manifest Transfer to Store	Values: Yes/No Yes: The Manifesting system will be called for transfer to store. No: The Manifesting system will not be called. Note: The interface will still need to be implemented; this just is to determine if it will be called.	No	Web Service Enablement	Boolean
Manifest Transfer to Warehouse	Values: Yes/No Yes: The Manifesting system will be called for transfer to warehouse No: The Manifesting system will not be called. Note: The interface will still need to be implemented; this just is to determine if it will be called.	No	Web Service Enablement	Boolean
OBCS Customer Order Delivery Validation	Values: Yes/No Yes: SIOCS-OBCS Integration will be enabled No: SIOCS-OBCS Integration will not be enabled	No	Web Service Enablement	Boolean
OBCS Customer Order Delivery Validation	Values: Yes/No Yes: SIOCS-OBCS Integration will be enabled No: SIOCS-OBCS Integration will not be enabled	No	Web Service Enablement	Boolean

Table 8-2 (Cont.) Store Admin Parameters

Options	Description	Default Value	Topic	Type
OMS Customer Order Delivery Query Address	Values: Yes/No Yes: query the address from an external service as part of processing in: submitting an order, delivering an order, or reading the address for usage in UI. No: address will not be queried from external system.	No	Web Service Enablement	Boolean
OMS Customer Order Delivery Validation	Values : Yes/No Yes: When confirming the delivery, the system will make a call out to an external system (such as an OMS) to validate the delivery status and delivery quantities before completing the dispatch. No: external system will not be called.	No	Web Service Enablement	Boolean
Sales Forecast Data	Yes: The web service for Sales Forecast Data will be called. No: The web service for Sales Forecast Data will NOT be called.	No	Web Service Enablement	Boolean
Send Event Alert External	Values: Yes / No Yes: Notification event alerts (non ad hoc notifications) will be sent externally via web service. No: Notification event alerts (non ad hoc notifications) will not be sent externally. Web service is not called.	No	Web Service Enablement	Boolean

Permissions

Table 8-3 Security Permissions

Permission	Topic	Usage
Access Ad Hoc Stock Count Tolerances	Admin	With this permission, the user will have access to the Ad hoc Stock Count Tolerance dialog.
Access Admin	Admin	With this permission, the user will have access to the Admin menu.
Access Auto-Receive Stores	Admin	With this permission, the user will have access to the Auto Receive Stores admin dialog.
Access Barcode Processor	Admin	With this permission, the user will have access to the Barcode Processor dialog.
Access Buddy Stores	Admin	With this permission, the user will have access to the Buddy Store dialog.
Access Carrier Services	Admin	With this permission, the user can access the Carrier Service dialog to add and edit the carrier service data.
Access Carriers	Admin	With this permission, the user can access the Carrier dialog to add or edit the carrier data.

Table 8-3 (Cont.) Security Permissions

Permission	Topic	Usage
Access Code Info	Admin	With this permission, the user can access the Code Info dialog to add, edit and delete code information.
Access Container Lookup	Admin	With this permission, the user will have access to the Container Lookups dialog.
Access Credential Administration	Admin	With this permission, the user will have access to the Credential Administration Screen. Without this permission, the user will not have access to the Credential Administration Screen.
Access Customer Order Picking Tolerances	Admin	With this permission, the user will have access to the Customer Order Picking Tolerance dialog.
Access Data Seed	Admin	With this permission the user will have the ability to start the data seeding job via the batch job admin.
Access Delivery Timeslot	Admin	User must have this permission in order for the Delivery Timeslot menu option to be available within the Data Setup menu. With this permission the user will be able to do all operations on this screen.
Access DCS Work Types	Admin	With this permission, a sysop user will have access to the DCS Work Type screen in the desktop application.
Access Extended Attribute	Admin	With this permission, the Extended Attributes Menu option is displayed under Admin/Configuration and the user gets the ability to setup and assign extended attributes.
Access Extended Attribute Dept Assign	Admin	With this permission, the Assign Extended Attributes Menu option is displayed under Admin/Configuration and the user gets the ability to assign new extended attributes and also remove the existing assignments.
Access Extended Attribute Setup	Admin	With this permission, the Setup Extended Attributes Menu option is displayed under Admin/Configuration and the user gets the ability to view and edit extended attributes.
Access External Service Administration	Admin	With this permission, the user will have access to the External Service Admin screen.
Access File Transfer Service	Admin	With this permission, the user will be able to access File Transfer Service screen
Access Finisher Lookup	Admin	With this permission, the user will have access to the Finisher Lookup functionality.
Access Fiscal Document External Reference Link URL	Admin	With this permission, the user will be able to view the Fiscal Doc URL
Access Flexible Attributes	Admin	With this permission, the user will have access to the Custom Flexible Attributes dialog and can setup (Create) and Delete flexible attributes. Note that CFAs are not editable once created due to data integrity.

Table 8-3 (Cont.) Security Permissions

Permission	Topic	Usage
Access Future Price Events	Admin	Desktop: With this permission, 'Future Price Events' will also be listed in the Price Events screen in Item Lookup. Without this permission, 'Future Price Events' will not be listed in the Price Events screen. Mobile: With this permission, the user will be able to view future price events in Pricing of Item Lookup. Without this permission, the user will only be able to see current and past events.
Access Initial Data Load	Admin	With this permission the user will have access to the Initial Data Load dialog. Without this permission the user will not have access to the Initial Data Load dialog.
Access Integration Dashboard	Admin	On desktop application, with this permission, the user can access the Integration dashboard.
Access Inventory Adjustment Reasons	Admin	With this permission, the user will have access to the Inventory Adjustment Reason admin dialog and the ability to setup and maintain inventory adjustment reason codes.
Access Fiscal Doc URL	Admin	With this permission, the user will be able to view the Fiscal Doc URL
Access Inventory Management	Admin	With this permission, the user will have access to the Inventory Management.
Access ISN Types	Admin	With this permission, user will be able to access the ISN Types dialog in the desktop application.
Access Item CFAs	Admin	With this permission, the CFAs will be available on Item Detail in Item Lookup
Access Item Lookup	Admin	With this permission, the user will have access to the Item Lookup functionality.
Access Item Maintenance	Admin	With this permission the user will be able to access the Item Maintenance screen in JET Mobile
Access Item Scan Number Lookup	Admin	With this permission, the user can access the Item Scan Number Lookup.
Access Job Admin	Admin	With this permission, the user can access the Job admin dialog.
Access Job Scheduler	Admin	With this permission, the user can access the Job scheduler dialog.
Access Lookup	Admin	With this permission, the user will have access to functionality within Lookups.
Access MPS Staged Messages	Admin	User must have this permission in order to access the MPS Staged Messages screen to view or edit the inbound and outbound messages.
Access MPS Work Types	Admin	User must have this permission in order to Access or Edit the Worker Type settings.
Access Operational Issues	Admin	With this permission, the user can access the Operational Issues dialog.
Access Operational Views menu	Admin	On EICS, with this permission, the operational views main menu is displayed, and user is allowed to view the different operational views.

Table 8-3 (Cont.) Security Permissions

Permission	Topic	Usage
Access Package Size	Admin	With this permission the user will have access to the Package Size admin dialog.
Access POS Transaction Resolution List	Admin	On desktop application, with this permission, the user can access the Transaction Resolution dialogue.
Access Price Events	Admin	With this permission, 'View Price Events' button will be available in the Item Detail screen hence the user will be able to access the Price Events screen.
Access Printer Setup	Admin	With this permission, the user can access the printer setup dialog.
Access Product Group Schedules	Admin	With this permission the user will have access to the Product Group Schedule dialog.
Access Product Groups	Admin	With this permission, the user will have access to the admin Product Group and Product Group Component functionality
Access Reports	Admin	With this permission, the user can access the Reports dialog.
Access RFID Locator	Admin	With this permission, the user can access the RFID Locator dialog in mobile.
Access Sequence Admin	Admin	With this permission the user will have the Sequence Admin menu option under Technical Maintenance menu.
Access SIOCS Managed Stores	Admin	With this permission, the user will have access to the SIOCS Managed Stores screen in EICS.
Access Shipment Reasons	Admin	With this permission, the user will have access to the Shipment Reason admin dialog and the ability to setup and maintain shipment reason codes.
Access Shipping Receiving	Admin	With this permission, the user will have access to functionality within Shipping/Receiving.
Access Store Administration	Admin	With this permission, the user will have access to Store Administration dialog.
Access Store Administration Default	Admin	With this permission, the user will have access to Store Administration default dialog.
Access Store Shipping Network	Admin	With this permission, the user will have access to the Store Shipping Network screen in the desktop application.
Access Sub Buckets	Admin	With this permission, the user can access the Sub bucket dialog.
Access Supplier CFAs	Admin	With this permission, the CFAs will be available on Supplier Detail in Supplier Lookup.
Access Supplier Lookup	Admin	With this permission the user will have access to Supplier Lookup and the Supplier Lookup menu option will appear in the drawer (main menu).
Access System Administration	Admin	With this permission, the user will have access to the System Administration dialog.
Access Technical Maintenance	Admin	With this permission, the user will have access to Technical Maintenance dialog.

Table 8-3 (Cont.) Security Permissions

Permission	Topic	Usage
Access Transaction History	Admin	With this permission the user will have access to the Transaction History dialog.
Access Translation Setup	Admin	With this permission, the user can access the Translation Setup dialog.
Access Troubled Transaction	Admin	With this permission, the user can select the Troubled Transaction List from the Inventory Management menu.
Access UDAs	Admin	With this permission, in Item Lookup the user will be able to search for an item by a search type of UDA. Also, the UDAs will be available on Item Detail in Item Lookup.
Access UDA Print Setup	Admin	With this permission, the user can access the UDA Print Setup dialog in the desktop application.
Access UIN Label Setup	Admin	With this permission, the user can access the UIN Label Setup dialog in the desktop application.
Access Unit of Measure	Admin	User must have this permission in order for the Unit of Measure menu option to be available within the Configuration menu. With this permission the user will be able to do all operations on this screen. (Add ""+"", Remove ""-"" and Edit).
All Stores Product Groups	Admin	With this permission, Product Groups for All Stores can be created as well as edited. Without this permission, Product Groups cannot be created for all stores, and they can only be viewed for all stores.
Allow Bulk Scan	Admin	With this permission, the user will have access to the Bulk Scan dialog. The Bulk Scan menu option will be in the footer menu of applicable transaction item list screens. The Bulk scan type will be available as a scan mode in the mode bar.
Allow Operational Issue Batch Execution Delete	Admin	With this permission, the user will be able to delete the issues in Bulk Import, Transaction Execution and Data Purge tasks via Operational Issues Dialog.
Create Item Scan Number	Admin	With this permission, the user is allowed to create a new item scan number in the system.
Create Notes	Admin	With this permission, the user will be able to add notes within the notes dialog.
Create Product Group Schedules	Admin	With this permission, the user can create new Product Group Schedules.
Create Product Groups	Admin	With this permission, the user will be able to create a new Product Group.
Create Translations	Admin	With this permission, the user can create new translations.

Table 8-3 (Cont.) Security Permissions

Permission	Topic	Usage
Delete Initial Data Load	Admin	With this permission the user will have the Delete Data button. Without this permission the user will not have the Delete Data button.
Delete Item Scan Number	Admin	With this permission, the user is allowed to delete an existing Item Scan Number.
Delete MPS Staged Messages	Admin	User must have this permission in order to delete the inbound and outbound messages.
Delete Product Group Schedules	Admin	With this permission, the user can delete Product Group Schedules.
Delete Product Groups	Admin	With this permission, the user can delete a Product Group.
Display Stock Locator	Admin	With this permission, the user will have access to Stock Locator within Item Lookup.
Edit Item Scan Number	Admin	With this permission, the user is allowed to edit an existing Item Scan Number. Applicable for webservice operation.
Edit Item Scan Number CDA	Admin	This is required for web service action to edit the Item Scan number CDAs.
Edit Job Schedules	Admin	With this permission the user can edit Job Schedules.
Edit POS Transaction	Admin	On desktop application, with this permission, the user can edit the troubled pos transaction message.
Edit Product Group Schedules	Admin	With this permission, the Product Group Schedule will be editable.
Edit Product Groups	Admin	With this permission, the user can Edit an existing Product Group which also includes adding and removing Product Group Components.
Edit UDA Print Setup	Admin	With this permission, the user can edit the UDA Print Setup values in the desktop application.
Submit Initial Data Load	Admin	With this permission the user will have the Submit Seed button. Without this permission the user will not have the Submit Seed button.
Update Resolution Status	Admin	With this privilege the Resolve/Reset button on the Troubled Transactions List screen will be displayed and enabled.
Update UIN Status	Admin	With this permission, the user can update the status of the UIN from the history screen.
View UIN History	Admin	With this permission, the user will be able to access the UIN Lookup feature and view the history in EICS.
Warehouse Inventory Access	Admin	With this permission, the warehouse inventory details will be included in the Stock Locator Item Lookup section.
Access Area	Area	With this permission, the user will have access to the Area dialog.
Confirm Area	Area	With this permission, the user will be able to Confirm an Area.

Table 8-3 (Cont.) Security Permissions

Permission	Topic	Usage
Create Area	Area	With this permission, the user will be able to create Areas.
Delete Area	Area	With this permission, the user will be able to delete an Area.
Edit Area	Area	With this permission, the user will be able to edit active Areas.
Access Customer Details	Customer Order	With this permission the user will have access to the Customer Details (name, address, and so on) associated with the customer order.
Access Customer Order	Customer Order	With this permission, the user will have access to Customer Orders dialog.
Access Customer Order Delivery	Customer Order	With this permission, a user can access the Customer Order Delivery dialog.
Access Customer Order Delivery Attribute	Customer Order	With this permission, the user is allowed to view the extended attributes in the functional dialog..
Access Customer Order Management	Customer Order	With this permission, the user will have access to the Customer Order Management operations within the drawer/menu.
Access Customer Order Pick	Customer Order	With this permission, the user will have access to the Customer Order Picking dialog.
Access Customer Order Quick Flow	Customer Order	With this permission, Customer Orders will display on the Open Transaction. This is for JET Mobile Quick Flow. Go to Transaction option will exist in Notifications.
Access Customer Order Reverse Pick	Customer Order	With this permission, the user will have access to the Customer Order Reverse Picking dialog.
Allow dispatch without Fiscal Document IDL	Customer Order	With this permission, the user will be able to dispatch the shipment without the Fiscal Doc ID/E-way bill ID being available.
Cancel Submit Customer Order Delivery	Customer Order	With this permission, the user will be able to Cancel Submit a “Submitted” status Customer Order Delivery that is a “web order.
Confirm Customer Order Pick	Customer Order	With this permission, the user can Confirm a customer order pick.
Confirm Customer Order Reverse Pick	Customer Order	With this permission, the user will be able to Confirm a customer order reverse pick.
Create Customer Order Delivery	Customer Order	With this permission, the user can create a new delivery for a Customer Order that is a “web order”. Used in conjunction with Create Customer Order Delivery for Shipment or Create Customer Order Delivery for Pickup permission. This permission must also exist in order to create a delivery that is not a web order from an external system.
Create Customer Order Delivery for Pickup	Customer Order	With this permission, the user will be able to create Customer Order Deliveries which are of type store pickup. Used in conjunction with the Create Customer Order Delivery permission.

Table 8-3 (Cont.) Security Permissions

Permission	Topic	Usage
Create Customer Order Delivery for Shipment	Customer Order	With this permission, the user will be able to create Customer Order Deliveries which are of type store shipment. Used in conjunction with the Create Customer Order Delivery permission.
Create Customer Order Pick	Customer Order	With this permission, the user will be able to create customer order picks. Picks can be created from within a Customer Order (for a single customer order pick). Picks can be created from within Customer Order Picking.
Create Customer Order Reverse Pick	Customer Order	With this permission, the user will be able to create customer order reverse picks.
Delete Customer Order Delivery	Customer Order	With this permission, the user will be able to delete a customer order delivery that is a “web order”.
Delete Customer Order Pick	Customer Order	With this permission, the user will be able to delete a customer order pick.
Delete Customer Order Reverse Pick	Customer Order	With this permission, the user will be able to delete a customer order reverse pick.
Dispatch Customer Order Delivery	Customer Order	With this permission, the user can Dispatch a Customer Order Delivery that is a “web order".
Dispatch Incomplete Customer Order Delivery	Customer Order	When dispatching a delivery that requires full delivery (Allow Partial Delivery = ‘No’), the entire delivery must be delivered in full. With this permission, the user will get a warning message and will be able to continue the dispatch without full delivery. Without this permission the delivery must be in full.
Edit Customer Order BOL	Customer Order	With this permission, the user will be able to edit the details of the Bill of Lading associated with a customer order delivery. This is done in the Edit Delivery screen. User must also have Edit Customer Order Delivery permission.
Edit Customer Order CFA	Customer Order	With this permission, the user will be able to capture CFAs for a customer order.
Edit Customer Order Delivery	Customer Order	With this permission, the user can edit an existing delivery for a Customer Order that is a “web order”.
Edit Customer Order Delivery Attribute	Customer Order	With this permission, the user is allowed to add/ remove the extended attributes.
Edit Customer Order Delivery CFA	Customer Order	With this permission, the user will be able to capture CFAs for a Customer Order Delivery
Edit Customer Order Pick	Customer Order	With this permission, the user will be able to edit active customer order picks.
Edit Customer Order Pick CFA	Customer Order	With this permission, the user will be able to capture CFAs for a Customer Order Pick.
Edit Customer Order Reverse Pick	Customer Order	With this permission, the user will be able to edit active reverse picks

Table 8-3 (Cont.) Security Permissions

Permission	Topic	Usage
Edit Customer Order Quick Flow	Customer Order	With this permission the user will be able to edit (assuming proper status) in the customer order quick flow, which may include picking, rejecting, and pickup / shipment.
Edit Customer Order Quick Flow Quantity	Customer Order	With this permission the user will be able to click on the item quantity and the quantity widget will open to edit the quantity. Without this permission, the quantity is not editable and only scanning is allowed.
Edit Customer Order Reverse Pick CFA	Customer Order	With this permission, the user will be able to capture CFAs for a Customer Order Reverse Pick
Edit Quantity Delivery	Customer Order	With this permission, the user will be able to tap on the item quantity and the quantity widget will open to edit the quantity.User must also have Edit Customer Order Delivery permission as well.Without this permission, the quantity is not editable via the manual dialog and only scanning is allowed.
Edit Quantity Picking	Customer Order	With this permission, the user will be able to tap on the item quantity and the quantity widget will open to edit the quantity.User must also have Edit Customer Order Pick permission. Without this permission, the quantity is not editable via the manual dialog and only scanning is allowed.
Edit Quantity Reverse Picking	Customer Order	With this permission, the user will be able to tap on the item quantity and the quantity widget will open to edit the quantity.User must also have Edit Customer Order Reverse Pick permission. Without this permission, the quantity is not editable via the manual dialog and only scanning is allowed.
Item Substitution For Picking	Customer Order	With this permission, the user will have access to the Item Substitution dialog within Customer Order Picking.
Reject Customer Order	Customer Order	With this permission, the user will be able to Reject a Customer Order.
Submit Customer Order Delivery	Customer Order	With this permission, the user will be able to Submit a Customer Order Delivery that is a “web order”. Used in conjunction with Modify Customer Order Delivery permission.
View Customer Order BOL	Customer Order	With this permission, the user can access and view the details of the Bill of Lading associated with a customer order delivery.
Container Items Limited To	Data	With this permission, the user can access data criteria associated with adding items to a container. This is selected by each individual criterion.
Counting Method	Data	With this permission, the user can access specific counting methods. This is selected by each individual counting method.
Department	Data	With this permission, the user can access specific department. This is selected by each individial department code.

Table 8-3 (Cont.) Security Permissions

Permission	Topic	Usage
Display List Diff Types	Data	With this permission, the user can access the display of diff types. This is selected by each individual diff type.
Inventory Adjustment Reason Code	Data	With this permission, the user can access specific inventory adjustment reason codes. This is selected by each individual reason code.
Item Basket Types	Data	With this permission, the user can access specific item basket types. This is selected by each individual basket types.
Location Types	Data	With this permission, the user can access specific location types. This is selected by each individual location type.
Print Format Type	Data	With this permission, the user can access specific print format types. This is selected by each individual format type.
Product Group Type	Data	With this permission, the user can access specific product group types. This is selected by each individual product group type.
Role Type	Data	With this permission, the user can access specific role types. This is selected by each individual role type.
RTV Reason Code	Data	With this permission, the user can access specific RTV reason codes. This is selected by each individual reason code.
RTV Shipment Reason Code	Data	With this permission, the user can access specific RTV shipment reason codes. This is selected by each individual reason code.
Scan List Type	Data	With this permission, the user can access specific scan list types. This is selected by each individual scan list type.
Shelf Adjustment Type	Data	With this permission, the user can access specific shelf adjustment types. This is selected by each individual adjustment type.
Shelf Replenishment Type	Data	With this permission, the user can access specific shelf replenishment types. This is selected by each individual type.
Store Order Delivery Timeslots	Data	With this permission, the user can access specific delivery timeslots. This is selected by each individual timeslot.
Transaction Type	Data	With this permission, the user can access specific transaction types. This is selected by each individual transaction type.
Transfer Destination Type	Data	With this permission, the user can access specific transfer destination types. This is selected by each individual transfer destination type.
Transfer Shipment Reason Code	Data	With this permission, the user can access specific transfer destination types. This is selected by each individual transfer destination type.
Access Adjust Container DSD Receiving	DSD Receiving	With this permission, the user will be able to adjust the container in a delivery.

Table 8-3 (Cont.) Security Permissions

Permission	Topic	Usage
Access Confirm Container DSD Receiving	DSD Receiving	With this permission, the user can confirm the container receipt from the supplier.
Access Confirm DSD Receipt	DSD Receiving	With this permission, the user will be able to confirm the Direct store delivery.
Access Create Container	DSD Receiving	With this permission, the user can create a new container in the DSD receipt.
Access Delete Container	DSD Receiving	With this permission, the user can delete a container.
Access Delete Receipt	DSD Receiving	With this permission, the user will be able to delete a direct store delivery.
Access Document DSD Receiving	DSD Receiving	With this permission, the user will be able to select the PO to apply items on the receipt.User must also have Edit container and Modify container permissions in order to do this operation.
Access DSD Receiving	DSD Receiving	With this permission, the user will have access to the DSD Receiving dialog.
Access DSD Receiving Ext. Attribute	DSD Receiving	With this permission, the user is allowed to view the extended attributes in the functional dialog..
Access Purchase Order	DSD Receiving	With this permission, the user will be able to access the Purchase Orders.
Access Reject Delivery	DSD Receiving	With this permission, the user will be able to reject a direct store delivery.
Activate Container Edits	DSD Receiving	User needs this permission combined with the other respective permission(s) to update any information on the container.
Add Unexpected Item	DSD Receiving	With this permission, the user will be able to add the unexpected items to the container.
Allow ASN Over Receiving	DSD Receiving	For PO's with an ASN: With this permission, when entering a received quantity, it is OK, to exceed the ASN quantity. Without this permission, the received quantity cannot exceed the ASN quantity.
Allow DSD Receiving With PO	DSD Receiving	With this permission, the user will be able to create a receipt against a PO but without ASN.
Allow DSD Receiving Without PO	DSD Receiving	With this permission, the user will be able to create a receipt without PO.
Allow PO Over Receiving	DSD Receiving	For PO's with and without an ASN: With this permission, when entering a received quantity, it is OK, to exceed the PO quantity. Without this permission, the received quantity cannot exceed the PO quantity.
Allow Receiving Damages	DSD Receiving	With this permission, the user will be able to receive damaged items and make all remaining quantity to be received as damaged.User must also have Edit container and Modify container permissions in order to do this operation.
Default Qty in All Containers	DSD Receiving	With this permission, the user will be able to default the received quantity for all the containers in the delivery.

Table 8-3 (Cont.) Security Permissions

Permission	Topic	Usage
Default Qty in Container	DSD Receiving	With this permission, the user will be able to default the received quantity with the remaining quantity in the container.User must also have permission to 'Modify container' in order to do this operation.
Display Expected Quantity	DSD Receiving	With this permission, the user will be able to view the expected quantity during the receipt.
Edit Container	DSD Receiving	With this permission, the user can edit the container line items, modify the line item quantities or delete the item.User must also have permission to 'Modify container' in order to do this operation.
Edit Container CFA	DSD Receiving	With this permission, the user will be able to capture CFAs for a Container in DSD Receiving.
Edit Container Info DSD Receiving	DSD Receiving	With this permission, the user will be able to edit the container information.User must also have permission to 'Modify container' in order to do this operation.
Edit Delivery CFA	DSD Receiving	With this permission, the user will be able to capture CFAs for a DSD.
Edit Delivery Info	DSD Receiving	With this permission, the user will be able to edit the delivery information.
Edit DSD Receiving Ext. Attribute	DSD Receiving	With this permission, the user is allowed to add/ remove the extended attributes.
Edit Quantity	DSD Receiving	With this permission, the user will be able to edit the quantity of the item in the delivery. User must also have Edit container and Modify container permissions in order to do this operation.
Override Not After Date Check	DSD Receiving	With this permission, the user can override the not after date check. This permission will allow the user to receive delivery where the receipt date passed the not after date in PO.
Override Supplier Discrepancies	DSD Receiving	With this permission, the user will be able to override the supplier discrepancies when quantity is validated.
Receive Direct Delivery on Shop Floor	DSD Receiving	With this permission, the user will be able to receive delivery on shop floor directly.
Access Inventory Adjustment	Inventory Adjustments	With this permission a user will have access to the Inventory Adjustment dialog.
Access Inventory Adjustment Attribute	Inventory Adjustments	With this permission, the user is allowed to view the extended attributes in the functional dialog.
Complete Inventory Adjustment	Inventory Adjustments	With this permission, the user can Confirm an inventory adjustment. User must also have data permissions for each adjustment reason on the adjustment.
Create Inventory Adjustment	Inventory Adjustments	With this permission, the user can create a new inventory adjustment. This permission must exist as well for a user to copy a "completed" inventory adjustment. User must also have data permissions for each adjustment reason on the adjustment.

Table 8-3 (Cont.) Security Permissions

Permission	Topic	Usage
Create Quick Adjustment	Inventory Adjustments	With this permission, the Adjust Inventory menu option will be available on JET Mobile in Item Lookup. Without this permission, the Adjust Inventory button will not be displayed.
Delete Inventory Adjustment	Inventory Adjustments	With this permission, the user will be able to delete an inventory adjustment. User must also have data permissions for each adjustment reason on the adjustment.
Edit Inventory Adjustment	Inventory Adjustments	With this permission, the user will be able to edit existing inventory adjustments. User must also have data permissions for each adjustment reason on the adjustment.
Edit Inventory Adjustment Attribute	Inventory Adjustments	With this permission, the user is allowed to add/ remove the extended attributes.
Edit Inventory Adjustment CFA	Inventory Adjustments	With this permission, the user will be able to capture CFAs for an inventory adjustment.
Edit Quantity	Inventory Adjustments	With this permission, the user will be able to tap on the item quantity and the quantity widget will open to edit the quantity. User must also have Edit Inventory Adjustment permission as well as data permissions for each adjustment reason on the adjustment.
Access Item Basket	Item Basket	With this permission, the user will have access to the Item Basket dialog.
All Stores Item Basket	Item Basket	With this permission, Item Baskets for All Stores can be created as well as edited. Without this permission, Item Baskets cannot be created for all stores, and they can only be viewed for all stores.
Confirm Item Basket	Item Basket	With this permission, the user will be able to Confirm an Item Basket.
Create Item Basket	Item Basket	With this permission, the user will be able to create Item Baskets.
Delete Item Basket	Item Basket	With this permission, the user will be able to delete an Item Basket.
Edit Item Basket	Item Basket	With this permission, the user will be able to edit active Item Baskets.
Edit Item Basket CFA	Item Basket	With this permission, the user will be able to capture CFAs on an Item Basket.
Edit Quantity Item Basket	Item Basket	With this permission, the user will be able to tap on the item quantity and the quantity widget will open to edit the quantity. User must also have Edit Item Basket permission. Without this permission, the quantity is not editable via the manual dialog and only scanning is allowed.
Import Item Basket	Item Basket	With this permission, the user will have the Import Item Basket menu option within an Item Basket on mobile.

Table 8-3 (Cont.) Security Permissions

Permission	Topic	Usage
Investigate Item Basket	Item Basket	With this permission, the investigate menu option in item lookup will be available and the user will be able to add an item to an existing item basket or create a new item basket if one doesn't exist for investigation in item lookup. Without this permission, the investigate menu option in item lookup will not be available.
Access Notifications	Notifications	With this permission, the bell notification icon will be displayed in the drawer/menu as well as on the Open Transactions header. The view on the notification will also be displayed.
Customer Order Pickup	Notifications	With this permission, the user will be notified, if the pick list has been created but not actioned.
Customer Order Pick Reminder	Notifications	With this permission, the user will be notified, if the pick list has been created but not actioned.
Customer Order Reauthorization	Notifications	With this permission the user will receive a notification when the payment reauthorization for a customer order is successful.
Customer Order Receipt	Notifications	With this permission, the user will be notified when customer orders are received.
Customer Order Reminder	Notifications	With this permission, the user will be notified when the customer order has not been fulfilled.
Damaged Delivery	Notifications	With this permission, the user will be notified when the delivery includes damaged items.
Display External Scanner Notifications	Notifications	With this permission, the user will be notified with a popup that appears when a scanner is connected or disconnected or has a low battery for that event.
Display Notification Warning	Notifications	With this permission, the user will receive a notification warning when a new notification is created / inserted into the system.
Finisher Delivery Unable to Auto-Receive	Notifications	With this permission, the user will receive a notification warning when a finisher delivery is not auto received.
Finisher UIN Discrepancy	Notifications	With this permission, the user will be notified when a finisher return received quantity does not match the number of serial numbers on the return. Without this permission, the user will not be notified.
Misdirected Container	Notifications	With this permission, the user will be notified when a container has been received in another location.
New Customer Order	Notifications	With this permission, the user will be notified when customer orders are created.
New Customer Order Reverse Pick	Notifications	With this permission, the user will be notified when a new cross channel customer order reverse picks arrives.
Over Received Quantity	Notifications	With this permission, the user will be notified when the number of pre-populated serial numbers exceeds the received quantity. Without this permission, the user will not be notified.

Table 8-3 (Cont.) Security Permissions

Permission	Topic	Usage
Receiving UIN Discrepancy	Notifications	With this permission, the user will be notified when the number of pre-populated serial numbers does not match the received quantity. Without this permission, the user will not be notified.
RTV Request Expiration Approaching	Notifications	With this permission, the user will be notified if the supplier return request expiration date is approaching.
RTV Unavailable request quantity	Notifications	With this permission, the user will be notified if there is not enough inventory in the unavailable bucket to send back to supplier from a return request.
Shipped Delivery Overdue	Notifications	With this permission, the user will be notified when the shipped delivery has not been received and has passed the expected date.
Store Delivery Unable to Auto-Receive	Notifications	With this permission, the user will be notified when a store delivery has discrepancies and cannot be auto received.
Store Receiving Over/Under	Notifications	With this permission, the user will be notified when a store transfer has over/under received quantities.
Transfer Request	Notifications	With this permission, the user will be notified when a transfer request is created.
Transfer Request Approved	Notifications	With this permission, the user will be notified when a transfer request is approved.
Transfer Request Expiration Approaching	Notifications	With this permission, the user will be notified when a transfer request has not been approved and the request is about to expire. This is based on the not after date set.
Transfer Request Rejected	Notifications	With this permission, the user will be notified when a transfer request is rejected.
Transfer Unavailable Request Quantity	Notifications	With this permission, the user will be notified when the requested quantity is no longer available at the requested source location.
UIN Items on Incoming ASN Failed	Notifications	With this permission, the user will be notified if an Auto Generated SN item is on the ASN with pre-generated numbers when processing thru the RIB. Without this permission, the user will not be notified.
Unexpected UIN (Store Changed)	Notifications	With this permission, the user will be notified when UINs are discovered at a store where they should not be. Without this permission, the user will not be notified.
Warehouse Delivery Unable to Auto-Receive	Notifications	With this permission, the user will be notified when the delivery includes pre-populated serial numbers and cannot be automatically received.

Table 8-3 (Cont.) Security Permissions

Permission	Topic	Usage
Access Open Transactions	Open Transactions	If the user has this permission, the Open Transactions menu option will be available in the drawer/menu and Quick Actions. Upon logging in the user will go to Open Transactions. If the user does not have this permission, the menu option will not be available in the drawer/menu or Quick Actions. Upon logging in the user will navigate to Quick Actions. . This permission is applicable only to JET mobile
Send Transaction Notification	Open Transactions	With this permission the user will be able to send a notification. This is done from within the Open Transactions dialog; the user will be able to swipe an open transaction and send a notification.
View Transactions	Open Transactions	With this permission the user will have the potential to view all the open transactions for the user's store (depending on data permissions). Without this permission, the Open Transactions dialog will still display (just without the transactions listed); however, the list of open transactions will be empty. This permission is applicable only to MAF Mobile.
Access Out of Stock Lookup	Operational Views	On EICS, with this permission, the Out of Stock operational view menu is displayed and user is allowed to view the out of stock operational view.
Access New Received Items	Operational Views	On EICS, with this permission the user will be able to access the Access New Items view in Operational Views.
Access Expiring Items Lookup	Operational Views	On EICS, with this permission the user will be able to access the Expiring Items view in Operational Views.
Access Stock Counts - Ready to Authorize	Operational Views	On EICS, with this permission the user will be able to access the Stock Counts - Ready to Authorize view in Operational Views.
Access Shopfloor Out of Stock	Operational Views	On EICS, with this permission the user will be able to access the Access Shopfloor Out of Stock view in Operational Views.
Accept RTV	RTV	With this permission, the user will be able to approve a return request. User must also have data permissions for each return reason on the return.
Access RTV	RTV	With this permission, a user will have access to the RTV dialog and the RTV menu option will appear in the Drawer.
Activate RTV Edits	RTV	User is required to have Activate RTV Edits permission in order for Edit RTV or Edit RTV Info to activate. That is, the changes made on the RTV, because the user had Edit RTV and Edit RTV Info permissions, cannot be committed unless the user has Activate RTV Edits permission.

Table 8-3 (Cont.) Security Permissions

Permission	Topic	Usage
Add Items To RTV	RTV	With this permission, the user will be able to add items to a return. User must also have Modify RTV and Edit RTV permissions as well as data permissions for each return reason on the RTV
Allow Over Accepting	RTV	With this permission, the user will be allowed to accept quantity more than the Requested quantity in the RTV Request.
Close RTV	RTV	With this permission, the user will be able to close an RTV. User must also have data permissions for each return reason on the return.
Create RTV	RTV	With this permission, the user can create a new return. User must also have data permissions for each return reason on the return.
Delete RTV	RTV	With this permission, the user can delete a return. User must also have data permissions for each return reason on the return.
Edit Quantity	RTV	With this permission, the user will be able to tap on the item quantity and the quantity widget will open to edit the quantity. User must also have Modify RTV and Edit RTV permissions as well as data permissions for each return reason on the RTV. Without this permission, the quantity is not editable via the manual dialog and only scanning is allowed.
Edit RTV	RTV	With this permission, the user will be able to edit existing RTV like line item details, qty, and so on. User must also have data permissions for each Return reason on the RTV.
Edit RTV CFA	RTV	With this permission, the user will be able to capture CFAs for an RTV document.
Edit RTV Info	RTV	With this permission, the user will be able to edit the header information of an RTV.
Reject RTV	RTV	With this permission, the user will be able to reject a return request. User must also have data permissions for each return reason on the return.
Access RTV Shipment	RTV Shipment	With this permission, the user can access shipments from RTV requests. Without this permission, the user will not be able to access the shipments from RTV requests.
Access RTV Shipment Attribute	RTV Shipment	With this permission, the user is allowed to view the extended attributes in the functional dialog.
Activate RTV Shipment Container Edits	RTV Shipment	With this permission, the user will be allowed to commit any changes made to the shipment.

Table 8-3 (Cont.) Security Permissions

Permission	Topic	Usage
Add Unexpected Items to RTV Shipment	RTV Shipment	With this permission, the user will be allowed to add items that are not present in the RTV Document, into the shipment. User must also have Modify Container and Edit Container permission as well as data permissions for each return reason on the container.
Adjust Carrier	RTV Shipment	With this permission, the user will be able to update the BOL details of a shipment even after at least one container has been confirmed.
Adjust Container RTV Shipment	RTV Shipment	With this permission, the user will be allowed to bring the container back to editable status. User must also have data permissions for each return reason on the container.
Allow dispatch without Fiscal Document ID	RTV Shipment	With this permission, the user will be able to dispatch the shipment without the Fiscal Doc ID/E-way bill ID being available
Allow over shipping RTV Shipment	RTV Shipment	With this permission, the user will be allowed to go over Approved quantity in the RTV document User must also have Modify Container and Edit Container permission as well as data permissions for each return reason on the container. Without this permission, the user will not be allowed to enter qty more than Approved qty.
Cancel Submit RTV Shipment	RTV Shipment	With this permission, the user can cancel submit RTV shipments.
Close RTV Shipment	RTV Shipment	With this permission, the user can close RTV shipments. Without this permission, the user will not be able to close RTV shipments.
Confirm RTV Shipment Container	RTV Shipment	With this permission, the user can confirm containers in the shipments. User must also have data permissions for each return reason on the container.
Create RTV Shipment	RTV Shipment	With this permission, the user can create shipments for RTV requests.
Create RTV Shipment Container	RTV Shipment	With this permission, the user will be allowed to create a container for the shipment.
Default Items to RTV Shipment	RTV Shipment	With this permission, the user will be allowed to add items from the RTV document into shipment. User must also have data permissions for each return reason on the shipment. Without this permission, the user will not be allowed to add items from the RTV document into shipment
Delete RTV Shipment	RTV Shipment	With this permission, the user can delete RTV shipments.
Delete RTV Shipment Container	RTV Shipment	With this permission, the user can delete containers in the shipments. User must also have data permissions for each return reason on the container.

Table 8-3 (Cont.) Security Permissions

Permission	Topic	Usage
Dispatch Shipment	RTV Shipment	With this permission, the user can dispatch RTV shipments.
Edit Container CFA	RTV Shipment	With this permission, the user will be able to capture CFAs for a container in RTV shipment.
Edit Container RTV Shipment	RTV Shipment	With this permission, the user will be allowed to edit the line item details, update qty, remove item, restore item, cancel the current edits, and so on, User must also have Modify Container permission as well as data permissions for each return reason on the container. Without this permission, the user will not be allowed to edit any line item details, update qty, remove item, restore item, cancel the current edits, and so on.
Edit Quantity	RTV Shipment	With this permission, the user will be able to tap on the item quantity and the quantity widget will open to edit the quantity. User must also have Modify Container and Edit Container permission as well as data permissions for each return reason on the container. Without this permission, the quantity is not editable via the manual dialog and only scanning is allowed.
Edit RTV Container Info	RTV Shipment	With this permission, the user will be allowed to edit the container header details. User must also have Activate Container Edits permission.
Edit RTV Shipment Attribute	RTV Shipment	With this permission, the user is allowed to add/ remove the extended attributes.
Edit RTV Shipment BOL	RTV Shipment	With this permission, the user will be allowed to edit the shipment BOL details.
Edit RTV Shipment Info	RTV Shipment	With this permission, the user will be allowed to edit the shipment header details.
Edit Shipment CFA	RTV Shipment	With this permission, the user will be able to capture CFAs on an RTV shipment.
Submit RTV Shipment	RTV Shipment	With this permission, the user can submit RTV shipments.
View RTV Shipment BOL	RTV Shipment	With this permission, the user will be allowed to view the shipment BOL details.
Access Role Maintenance	Security	User must have this permission for the Role Maintenance menu option to be available under Security in EICS.
Access Security	Security	With this permission the user will have access to the Security dialog in EICS.
Access User Maintenance	Security	User must have this permission for the User Assignment menu option to be available under Security in EICS.
Delete Role	Security	User must have this permission in order to delete roles.

Table 8-3 (Cont.) Security Permissions

Permission	Topic	Usage
Delete User	Security	User must have this permission in order to delete user profiles.
Edit User	Security	User must have this permission in order to assign roles and stores to a user.
Access Scan List	Shelf Replenishment	With this permission, the user will have access to Item Scan List dialog.
Access Shelf Adjustment	Shelf Replenishment	With this permission, the user will have access to Shelf Adjustment dialog.
Access Shelf Replenishment	Shelf Replenishment	With this permission, the user will have access to the in-store replenishment dialog.
Confirm Shelf Adjustment	Shelf Replenishment	With this permission, the user can confirm the shelf adjustment.
Confirm Shelf Replenishment	Shelf Replenishment	With this permission, the user can confirm the replenishment pick.
Create Scan List	Shelf Replenishment	With this permission, the user can create a new item scan list. User must also have data permissions for each scan list type to create a new scan list of that type.
Create Shelf Adjustment	Shelf Replenishment	With this permission, the user can create a new shelf adjustment.
Create Shelf Replenishment	Shelf Replenishment	With this permission, the user can create a new shelf replenishment pick. User must have the data permission for each shelf replenishment pick type to do this operation.
Default Shelf Replenishment Quantity	Shelf Replenishment	With this permission, the user can default the quantity on replenishment pick. The user must also have the Edit replenishment permission to do this operation.
Delete Scan List	Shelf Replenishment	With this permission, the user will be able to delete a scan list.
Delete Shelf Adjustment	Shelf Replenishment	With this permission, the user can delete the shelf adjustment.
Delete Shelf Replenishment	Shelf Replenishment	With this permission, the user can delete the shelf replenishment pick.
Edit Scan List	Shelf Replenishment	With this permission, the user will be able to edit and save the scan list.
Edit Scan List CFA	Shelf Replenishment	With this permission, the user will have the ability to capture CFAs for a Scan List.
Edit Scan List Quantity	Shelf Replenishment	With this permission, the user will be able to edit the quantity on the UI using the widget. User must also have the Edit Scan List permission to do this.
Edit Shelf Adjustment	Shelf Replenishment	With this permission, the user can edit and save the shelf adjustment.
Edit Shelf Adjustment CFA	Shelf Replenishment	With this permission, the user will have the ability to capture CFAs for a Shelf Adjustment.

Table 8-3 (Cont.) Security Permissions

Permission	Topic	Usage
Edit Shelf Adjustment Quantity	Shelf Replenishment	With this permission, the user can edit the quantity using the widget on the UI.
Edit Shelf Replenishment	Shelf Replenishment	With this permission, the user will be able to edit the existing shelf replenishment pick.
Edit Shelf Replenishment CFA	Shelf Replenishment	With this permission, the user will have the ability to capture CFAs for Replenishment Pick List.
Edit Shelf Replenishment Quantity	Shelf Replenishment	With this permission, the user can edit the quantity using the quantity widget on the UI. Without this permission, the quantity is not editable via the manual dialog and only scanning is allowed.
Replenishment Item Substitution	Shelf Replenishment	With this permission, the user can substitute the pick items. The user must also have the Edit replenishment permission to do this operation.
Access Quick Count	Stock Counts	With this permission, the user will have access to Quick count dialog in JET mobile application.
Access Stock Count	Stock Counts	With this permission, the user can access the Stock Counts dialog.
Access Stock Count Attribute	Stock Counts	With this permission, the user is allowed to view the extended attributes in the functional dialog.
Apply Late Sales	Stock Counts	With this permission, Apply Late Sales button will be available to the user in the Stock Count Authorization Detail screen.
Complete All Stock Count Children	Stock Counts	On JET mobile, with this permission, the user will be able to access the complete all option on the child stock count list to complete all the child stock counts under the master stock count.
Complete Child Stock Count	Stock Counts	With this permission, the user can complete the child stock count.
Confirm Authorization Stock Count	Stock Counts	With this permission, the user can confirm the authorization.
Create Ad Hoc Stock Count	Stock Counts	With this permission, the user can create a new adhoc stock count.
Delete Stock Count	Stock Counts	With this permission, the user can delete a stock count.
Edit Adhoc Stock Count	Stock Counts	With this permission, the user can edit the ad hoc stock count.
Edit Adhoc Stock Count Lock	Stock Counts	With this permission, the user will have the ability to enable and disable the Adhoc Stock count Lock for an adhoc stock count.
Edit Authorizaton Stock Count	Stock Counts	With this permission, the user can access the Stock count authorization dialog on the desktop. Ability to apply late sales.
Edit Stock Count Attribute	Stock Counts	With this permission, the user is allowed to add/ remove the extended attributes.
Edit Stock Count CFA	Stock Counts	With this permission, the user will have the ability to capture CFAs in Stock Count and Recount.

Table 8-3 (Cont.) Security Permissions

Permission	Topic	Usage
Edit Stock Count Quantity	Stock Counts	With this permission, the user can edit the quantity using the quantity widget.
Edit Unit Amount Stock Count	Stock Counts	With this permission, the user is allowed to do update for the stock counts that are of type unit and amount
Edit Unit Stock Count	Stock Counts	With this permission, the user can update the stock count of unit or problem line stock count types.
Recount Stock Count	Stock Counts	On Mobile, with this permission, when the user selects a stock count in the Stock Count List screen and the stock count is in 'Recount' status, the Recount Items screen will open in edit mode. Without this permission, the screen will open in view-only mode.
Rejected Item Stock Count	Stock Counts	With this permission, the user will have access to the Rejected Items dialog.
Snapshot Stock Count	Stock Counts	With this permission, the user is allowed to take a snapshot at the master stock count level.
Snapshot Stock Count Child	Stock Counts	With this permission, the user is allowed to take the snapshot at the child stock count level.
Stock Count Import Basket	Stock Counts	With this permission, the user will have the Import Item Basket footer menu option in the Bulk Scan screen within Stock Counts.
Update Authorization Quantity	Stock Counts	With this permission, the user can update the authorization quantity and default the last count quantity to authorized quantity while in the process of authorization.
View Variance	Stock Counts	With this permission, the user will be able to view the variance (difference) between the snapshot quantity and the total quantity counted during counting.
Access Store Orders	Store Order	With this permission, the user will have access to the Store Orders dialog.
Approve Store Orders	Store Order	With this permission, the user will be able to Approve a Store Order.
Cancel Submit Store Order	Store Order	This permission will be needed in order for the Cancel Submit option to be available in the footer menu of the Delivery Items screen.
Create Store Orders	Store Order	With this permission, the user will be able to create Store Orders.
Delete Store Orders	Store Order	With this permission, the user will be able to delete a Store Order.
Display Sales Forecast	Store Order	With this permission, the Sales Forecast on the Sales Data screen will be displayed.
Display Sales History	Store Order	With this permission, the Sales History on the Sales Data screen will be displayed.

Table 8-3 (Cont.) Security Permissions

Permission	Topic	Usage
Display Store Order Cost	Store Order	With this permission the Total Estimated Cost will be displayed on the Info screen in store orders. The unit cost will be displayed on the Store Orders Item Detail. The Refresh Cost button will be displayed in the Store Order Items footer menu.
Edit Quantity	Store Order	With this permission, the user will be able to tap on the item quantity and the quantity widget will open to edit the quantity. User must also have Edit Store Orders permission. Without this permission, the quantity is not editable via the manual dialog and only scanning is allowed.
Edit Store Order CFA	Store Order	With this permission, the user will be able to edit CFAs on the store order. The Edit CFA menu option will be available in the footer menu.
Edit Store Orders	Store Order	With this permission, the user will be able to edit active Store Orders.
Edit Submitted Status	Store Order	This permission will allow a user to edit a store order that is in 'submitted' status. Without this permission, 'submitted' status will be noneditable/view only.
Submit Store Order	Store Order	This permission will be needed in order for the Submit option to be available in the footer menu of the Store Order Items screen.
Access DCS Work Type	Technical Maintenance	With this permission, users will have access to the DCS Work Type screen in the desktop application.
Access Format Assignment	Ticketing	With this permission, the user is allowed to access the format assignment dialog in the desktop application.
Access Print Format	Ticketing	With this permission, the user is allowed to access the ticket print format dialog.
Access Print Item	Ticketing	With this permission, the user is allowed to access the print item dialog.
Access Ticket List	Ticketing	With this permission, the user can access the Ticket List.
Access Ticket Template Upload	Ticketing	With this permission, the user can access the Upload Ticket Templates screen to upload the ticketing layout.
Allow Override Ticket Price	Ticketing	With this permission, the user can override the ticket price on the ticket detail screen.
Create Format Assignment	Ticketing	With this permission, the user is allowed to create a new item basket based format assignment.
Create Ticket	Ticketing	With this permission, the user is allowed to create a new ticket in the ticketing dialog.
Delete Format Assignment	Ticketing	With this permission, the user is allowed to delete a format assignment.
Delete Ticket	Ticketing	With this permission, the user is allowed to delete a ticket in the ticketing dialog.
Delete Ticket Template	Ticketing	With this permission, the user is allowed to delete a ticket template screen.

Table 8-3 (Cont.) Security Permissions

Permission	Topic	Usage
Edit Format Assignment	Ticketing	With this permission, the user is allowed to edit an existing format assignment.
Edit Ticket	Ticketing	With this permission, the user can edit an existing ticket.
Print Ticket	Ticketing	With this permission, the user can print the tickets in the ticketing dialog.
Print Tickets from Container Items	Ticketing	With this permission, the user is allowed to generate and print tickets from the container items screen both in transaction and lookup
Accept Transfer Request	Transfer	With this permission, the user will be able to accept a transfer request.
Access Transfer	Transfer	With this permission, a user will have access the transfers.
Access Transfer context	Transfer	With this permission, a user will be able to view the Context type details in a transfer.
Access Transfer Request	Transfer	With this permission, a user will have access to the Transfer dialog in the application
Approve Transfer	Transfer	With this permission, the user will be able to approve a transfer document.
Close Transfer	Transfer	With this permission, the user will be able to close a transfer.
Create Request	Transfer	With this permission, the user will be able to create a transfer request.
Create Transfer	Transfer	With this permission, the user will be able to create a transfer document.
Delete Request	Transfer	With this permission, the user will be able to delete a transfer request.
Delete Transfer	Transfer	With this permission, the user will be able to delete a transfer document.
Edit Quantity Transfer Request	Transfer	With this permission, the user will be able to tap on the item quantity and the quantity widget will open to edit the quantity. Without this permission, the quantity is not editable via the manual dialog and only scanning is allowed.
Edit Transfer	Transfer	With this permission, the user will be able to edit a transfer document.
Edit Transfer Request	Transfer	With this permission, the user will be able to edit a transfer request.
Edit Transfer Request CFA	Transfer	With this permission, the user will be able to capture CFAs in transfer request documents.
Reject Transfer Request	Transfer	With this permission, the user will be able to reject a transfer request.
Request Transfer	Transfer	With this permission, the user will be able to submit a transfer request.
Access Quick Receiving	Transfer Receiving	With this permission, a user will have access to Transfer Quick Receiving.

Table 8-3 (Cont.) Security Permissions

Permission	Topic	Usage
Access Transfer Receiving	Transfer Receiving	With this permission, the user is allowed to view the extended attributes in the functional dialog.
Access Transfer Receiving Attribute	Transfer Receiving	With this permission, the user is allowed to view the extended attributes in the functional dialog.
Activate Container Edits	Transfer Receiving	With this permission, the user will be allowed to commit any changes made to the container. The user will be able to edit the details with the Edit permission however for the changes to be saved to the DB, the Activate Container Edits permission is needed.
Add Unexpected Item to Transfer Receiving	Transfer Receiving	With this permission, the user will be allowed to receive items that are not present in the original delivery. User must also have Modify Container and Edit Container permission. Without this permission, the user will not be allowed to receive items that are not present in the original delivery.
Adjust Container	Transfer Receiving	With this permission, the user will be able to bring back a confirmed container to editable status.
Allow Default Zero at Confirmation	Transfer Receiving	With this permission, the user will be able to confirm a transfer receipt with the option to set all non received items to zero. Without this permission, if there are any non received items on the container, the user will get a hard stop and not be able to confirm the delivery.
Allow Over Receiving Store to Store ASN	Transfer Receiving	With this permission, the user will be able to over receive an ASN from store to store..
Confirm Container	Transfer Receiving	With this permission, the user will be able to confirm a container.
Confirm Receipt	Transfer Receiving	With this permission, the user will be able to confirm a transfer delivery.
Create Container	Transfer Receiving	With this permission, the user will be able to create a container.
Default Qty in All Containers	Transfer Receiving	With this permission, the user will be able to default the expected qty in received qty field in all the containers.
Default Quantity in Container	Transfer Receiving	With this permission, the user will be able to default the expected qty in received qty field for the items in the container. User must also have Modify Container and Edit Container permission.
Delete Container	Transfer Receiving	With this permission, the user will be able to delete a container.
Display Expected Qty	Transfer Receiving	With this permission, the user will be able to view Expected Qty of an item in the containers.

Table 8-3 (Cont.) Security Permissions

Permission	Topic	Usage
Edit Container	Transfer Receiving	With this permission, the user will be allowed to edit the line item details, update qty, remove item, restore item, cancel the current edits, and so on. User must also have Modify Container permission for the changes to be committed to the DB. Without this permission, the user will not be allowed to edit any line item details, update qty, remove item, restore item, cancel the current edits, and so on.
Edit Container CFA	Transfer Receiving	With this permission, the user will be allowed to capture CFAs in the containers of transfer deliveries.
Edit Container Info	Transfer Receiving	With this permission, the user will be allowed to edit the container header details. User must also have Modify Container permission for the changes to be committed to the DB.
Edit Delivery CFA	Transfer Receiving	With this permission, the user will be allowed to capture CFAs in the transfer deliveries.
Edit Quantity	Transfer Receiving	With this permission, the user will be able to tap on the item quantity and the quantity widget will open to edit the quantity. User must also have Modify Container and Edit Container permission. Without this permission, the quantity is not editable via the manual dialog and only scanning is allowed.
Edit Receiving Info	Transfer Receiving	With this permission, the user will be able to edit the header details of a transfer delivery
Edit Transfer Receiving Attribute	Transfer Receiving	With this permission, the user is allowed to add/ remove the extended attributes.
Misdirected Container	Transfer Receiving	With this permission, the user will be allowed to copy the items from a misdirected container.
Receive On Shop Floor	Transfer Receiving	With this permission, the user will be able to receive the container on the shopfloor. User must also have Modify Container and Edit Container Info permission. Without this permission, user will not be able to receive the container on the shopfloor.
Record Receipt Damages	Transfer Receiving	With this permission, the user will be able to receive damaged items in a transfer delivery.
Access Container Attribute	Transfer Shipment	With this permission, the user is allowed to view the extended attributes in the functional dialog.
Access Shipment	Transfer Shipment	With this permission, a user will have access to the Transfer Shipment dialog for the user in the application.
Activate Container Edits	Transfer Shipment	With this permission, the user will be allowed to commit any changes made to the shipment. The user will be able to edit the details with the Edit Container permission however for the changes to be saved to the DB, the Activate Container Edits permission is needed. Without this permission, the user will not be allowed to commit any changes made to the shipment.

Table 8-3 (Cont.) Security Permissions

Permission	Topic	Usage
Add Items with No Document	Transfer Shipment	With this permission, the user will be allowed to create an adhoc document through Shipments and add items to it. This controls the visibility of 'No document' button on the Select Document screen. Without this permission, the user will not be allowed to create an adhoc document through shipments and add items to it.
Add Unexpected Item to Transfer Shipment	Transfer Shipment	With this permission, the user will be allowed to add items that are not present in the Transfer Document, into the shipment. User must also have Modify Container and Edit Container permissions. Without this permission, the user will not be allowed to add items that are not present in the Transfer document, into the shipment.
Adjust Carrier	Transfer Shipment	With this permission, the user will be able to update the BOL details of a shipment even after at least one container has been confirmed. Without this permission, the user will not be able to update the BOL details of a shipment after at least one container has been confirmed.
Adjust Container	Transfer Shipment	With this permission, the user will be allowed to bring the container back to editable status.
Allow dispatch without Fiscal Document ID	Transfer Shipment	With this permission, the user will be able to dispatch the shipment without the Fiscal Doc ID/E-way bill ID being available
Cancel Submit Shipment	Transfer Shipment	With this permission, the user can cancel submit Transfer shipments.
Confirm Container	Transfer Shipment	With this permission, the user can confirm containers in the shipments.
Create Shipment	Transfer Shipment	With this permission, the user can create shipments for Transfer documents. Without this permission, the user will not be able to create shipments for Transfer documents.
Delete Container	Transfer Shipment	With this permission, the user can delete containers in the shipments.
Delete Shipment	Transfer Shipment	With this permission, the user can delete transfer shipments.
Dispatch Shipment	Transfer Shipment	With this permission, the user can dispatch shipments.
Edit Container	Transfer Shipment	With this permission, the user will be allowed to edit the line item details, update qty, remove item, restore item, cancel the current edits, and so on., User must also have Modify Container permission. Without this permission, the user will not be allowed to edit any line item details, update qty, remove item, restore item, cancel the current edits, and so on.
Edit Container Attribute	Transfer Shipment	With this permission, the user is allowed to add/ remove the extended attributes.

Table 8-3 (Cont.) Security Permissions

Permission	Topic	Usage
Edit Container CFA	Transfer Shipment	With this permission, the user will be able to capture CFAs in the containers of transfer shipments.
Edit Container Info	Transfer Shipment	With this permission, the user will be allowed to edit the container header details. User must also have Modify Container permission.
Edit Quantity	Transfer Shipment	With this permission, the user will be able to tap on the item quantity and the quantity widget will open to edit the quantity. User must also have Modify Container and Edit Container permission. Without this permission, the quantity is not editable via the manual dialog and only scanning is allowed.
Edit Shipment BOL	Transfer Shipment	With this permission, the user will be allowed to edit the shipment BOL details.
Edit Shipment CFA	Transfer Shipment	With this permission, the user will be able to capture CFAs in transfer shipments.
Edit Shipment Info	Transfer Shipment	With this permission, the user will be allowed to edit the shipment header details.
Override Exclude Shipping Network	Transfer Shipment	With this permission, user will be able to override Shipping Network exclusion and will be able to dispatch to the stores that are not in the network.
Select Container Document	Transfer Shipment	With this permission, the user will be allowed to select transfer documents to be added to the shipment. User must also have Modify Container and Edit Container permission.
Submit Shipment	Transfer Shipment	With this permission, the user can submit shipments.
Create UIN on the Fly	UIN	With this permission, the user is allowed to create a UIN on the fly when creating an inventory adjustment using a reason code of Disposition Movement from Out (Dist) to Available to Sell (ATS) = UIN Status in Stock. User must also have Edit Inventory Adjustment permission as well as data permissions for each adjustment reason on the adjustment.

Data Permissions

Data permissions need to be created for the following:

- Container Items Limited To
- Counting Method
- Departments
- Display List Diff Types
- Inventory Adjustment Reason Code
- Item Basket Types

- Location Type
- Print Format Type
- Product Group Type
- Role Type
- RTV Reason Code
- RTV Shipment Reason Code
- Scan List Type
- Shelf Adjustment Type
- Shelf Replenishment Type
- Store Order Delivery Timeslots
- Transaction Type
- Transfer Destination Type
- Transfer Shipment Reason Code

9
File Transfer Services

This chapter covers the following topics:

- Overview
- How to Call FTS APIs
- Handling Import Data Files
- Handling Export Data Files
- File Transfer Service UI
- FTS API Specifications
- File Transfer Service Troubleshooting
- Test FTS API using Postman

Overview

Oracle Cloud Infrastructure Object Storage is an internet-scale, high-performance storage platform that offers reliable and cost-efficient data durability.

File Transfer Service (FTS) for the Store Inventory Cloud Services is available as JSON REST services. These APIs allows you to manage uploading and downloading files to Object Storage.

Access to files is through a Pre-Authenticated Request (PAR), which is a URL that requires no further authentication to upload or download to the application's object storage. To retrieve a PAR, you must use the appropriate FTS services.

The FTS APIs enables external application to import files to and export files from Object Storage used by the solutions.

These APIs provides following services:

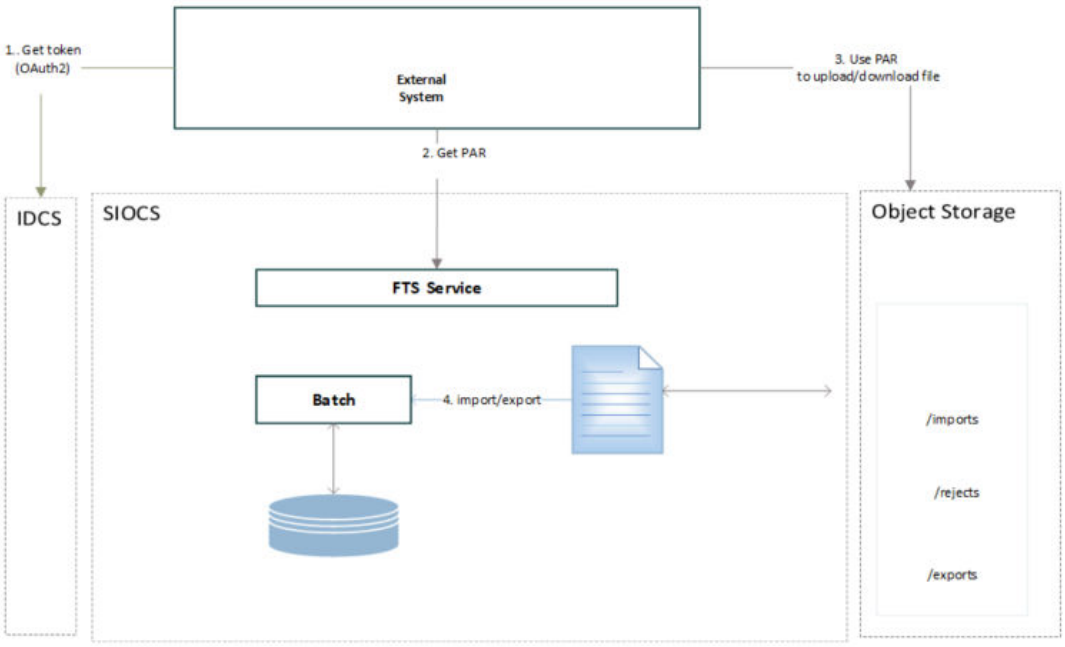
- Ping to check FTS Service health
- List storage prefixes
- List files in object storage
- Move files from object storage
- Delete Files from object storage
- Request Upload PAR
- Request Download PAR

The general process flow below describes how the external solution application interacts with FTS service for transferring files to cloud solution service:

1. The external application gets an Oauth2 token from IDCS.
2. The external application makes an FTS request with the Oauth2 token to request Pre-Authentication.

3. Once the PAR is received, the external application uploads a file to object storage using the URL included within the response.
4. The file uploads to application object storage and will be processed by the application batch jobs.

Figure 9-1 File Transfer Service Process Flow



In addition to public FTS endpoints, SIOCS also provides a File Transfer Service User Interface to view files in cloud solution object storage, to upload and download file interactively once logged into the SIOCS web client. Refer to File Transfer Service UI section for details.

How to Call FTS APIs

To interact with FTS, you must use the REST APIs provided. The endpoints URLs are relative to cloud solution integration base URL, and endpoints also include the object storage bucket name which is allocated for your environment for file services.

- Service Base URL
- FTS Bucket Name
- FTS Endpoints
- Preparing for Authorization
- Retrieving Access Client Token
- FTS API Call Common Headers
- How to Use FTS API to find Object Storage Prefixes
- How to Use FTS APIs to Upload Files to Object Storage
- How to Use FTS API to List Files in Object Storage

- How to Use FTS APIs to Download Files from Object Storage

Service Base URL

The Cloud service base URL follows the format:

https://rex.retail.<Region Name>.ocs.oraclecloud.com/<Customer Subnamespace>/siocs-int-services/api/

> **Note:**
>
> The <Region Name> and <Customer Subnamespace> part of the URL should be replaced with the one specific to your environment. This will be the same as your cloud service Application URL provided in the Welcome email.

FTS Bucket Name

For each customer environment, logical containers (buckets) are created in Object Storage for storing objects used by the cloud application. The file transfer bucket name is created and set when the environment is provisioned. The bucket name is required to move files between Oracle Cloud and your local system using file transfer services.

rgbu_rex_cnprod_<cust_env>

Example:

rgbu_rex_cnprod_rgbu-rex-custA-stg1-siocs

The ' File Transfer Service Bucket Name' is a restricted system configuration parameter on the EICS System Configuration screen. A customer Admin user (with the IDCS application role *sim_admin_users*) can perform the following steps to view the bucket name.

1. Log in to EICS web client as customer Admin user.
2. Go to **Configuration System**.
3. Check the values setting for name **File Transfer Service Bucket Name**.

FTS Endpoints

Open API documents can be viewed via the following URL:

https://{external_load_balancer}/{cust_env}/siocs-int-services/public/api/Fts.json

The table below lists the API end points for different file operations. See FTS API Specifications for details.

Table 9-1 FTS Endpoints

Service	Method	FTS Endpoint URLs
Ping	GET	{Service Base URL}/fts/ping
List Prefixes	GET	{Service Base URL}/fts/{FTS Bucket Name}/listprefixes
List Files	GET	{Service Base URL}/fts/{FTS Bucket Name}/listfiles
Move Files	POST	{Service Base URL}/fts/{FTS Bucket Name}/movefiles

Table 9-1 (Cont.) FTS Endpoints

Service	Method	FTS Endpoint URLs
Delete Files	POST	{Service Base URL}/fts/{FTS Bucket Name}/delete
Request Upload PAR	POST	{Service Base URL}/fts/{FTS Bucket Name}/upload
Request Download PAR	POST	{Service Base URL}/fts/{FTS Bucket Name}/download

> **Note:**
>
> The example in this section uses curl command line tools. You may also use Postman to test the FTS REST APIs for testing purpose. Refer to Test FTS API using Postman.

Preparing for Authorization

FTS Client Id and Client Secret

Obtaining a Token

FTS APIs use OAuth2.0 for authorization.

To generate a token from IDCS, an IDCS application client will need to be created for you to use.

The Customer Administration users must create their own client credential IDCS application using the Oracle Retail Home Cloud Service. For additional details, refer to Oracle® Retail Home Administration Guide- Chapter: Oauth Application Configuration chapter – Section: Creating OAuth Client Applications.

The App name and scope that should be used for the FTS IDCS application creation should be environment specific using the format :

App Name- RGBU_SIOCS_<ENV>_EICS_FTS_INT

Scope- rgbu:siocs:integration-<ENV>

Example:

App Name- RGBU_SIOCS_STG1_ EICS_FTS_INT

Scope- rgbu:siocs:integration-STG1

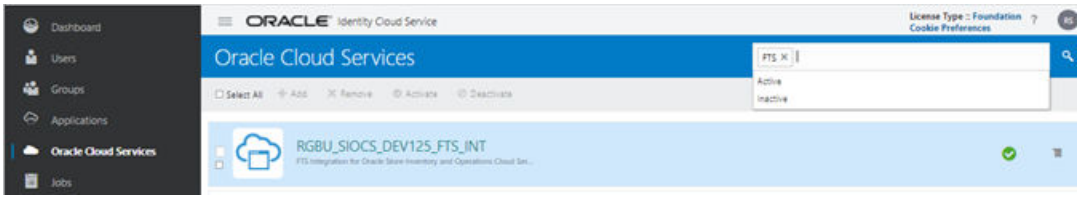
Steps to retrieve the FTS Client ID and Client Secret from IDCS:

1. Customer's IDCS Administrator log into Oracle Identity Cloud Service (IDCS) console.
2. In the left navigation panel, select **Oracle Cloud Service.**
3. On the search field, type in "FTS".
4. From the search result, find your FTS client application for cloud environment.

   FTS Client ID is like: RGBU_SIOCS_<ENV>_EICS_FTS_INT_APPID

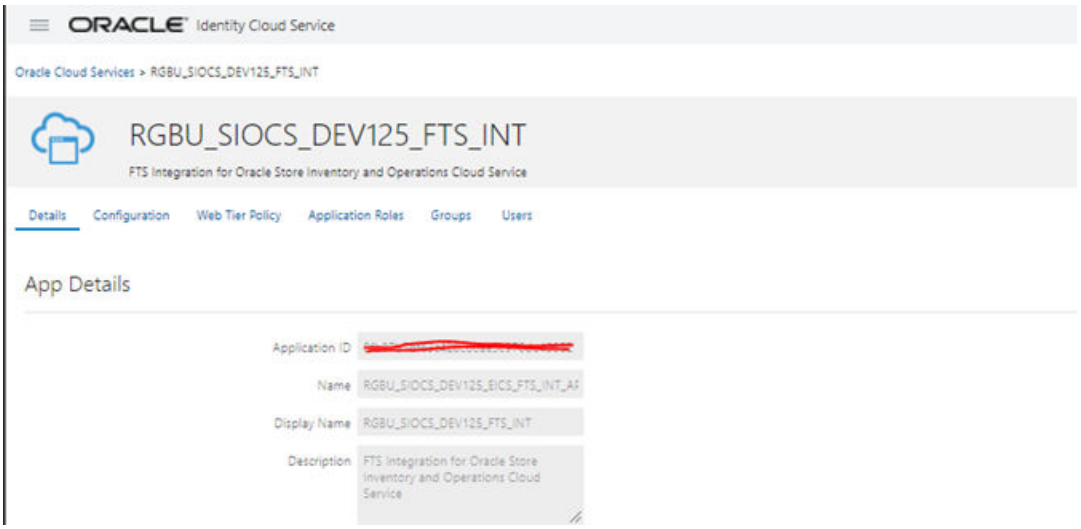
   (Example <ENV>: DEV1, STG1, PROD1 ..)

Figure 9-2 FTS Client Application



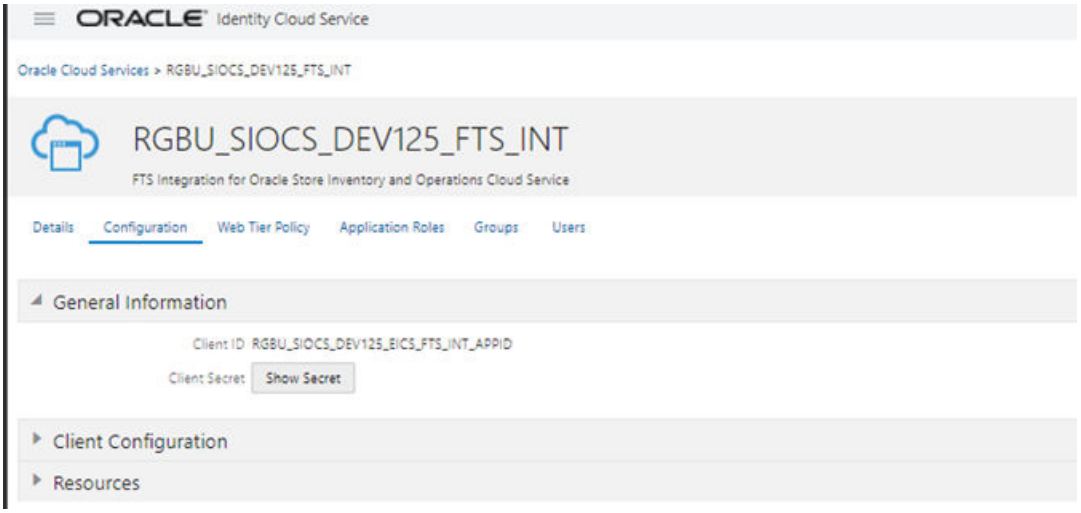
5. Click the client application, which will take you to the **Application Detail Panel**.

Figure 9-3 Application Detail Panel



6. Select the **Configuration** tab to view client Id.

Figure 9-4 Configuration Tab



7. Click **Show Secret** to see the password.

OAuth Scopes for FTS

Custom environment specific scope.

The scope pattern that is used in the FTS IDCS application creation template is rgbu:siocs:integration-{env}

For example:

rgbu:siocs:integration-STG1

IDCS OAuth2 Token URL

IDCS token URL to obtain Oauth2 token.

Example IDCS_TOKENT_URL:

https://idcs-XXXXXXXX.identity.oraclecloud.com/

Using the above URL,

IDCS_TOKENT_URL = {IDCL_BASE_URL}/oauth2/v1/token

Retrieving Access Client Token

The following is required in headers for making OAuth2.0 enabled REST Services.

- Please contact customer's IDCS administrator for FTS Client ID and Client Secret.
- An access token using the Client ID and secret from IDCS.

Example: get access Token Use Curl

```
export ACCESS_TOKEN="$(curl -u <Client ID>:<Secret> -H 'Content-Type:
application/x-www-form-urlencoded;charset=UTF-8' --request POST https://
<IDCS_BASE_URL>/oauth2/v1/token -d 'grant_type=client_credentials&scope=<Scope>'
| jq -r '.access_token')"
```

In above example, substitute the variables with proper values for your environment. See FTS Client Id and Client Secret section for obtaining Credential Client ID and Client Secret.

> **Note:**
>
> You need to have curl and jq client tool installed on your client machine for using curl for testing.

For example:

```
export ACCESS_TOKEN="$(curl -u RGBU_SIOCS_ZZZZ_EICS_FTS_INT_APPID:<secret> -H
'Content-Type: application/x-www-form-urlencoded;charset=UTF-8' --request POST
https://idcs-ZZZZ/oauth2/v1/token -d
'grant_type=client_credentials&scope=rgbu:siocs:integration-X' | jq -r
'.access_token')"
```

FTS API Call Common Headers

Each call to FTS Endpoint should contain the following Request headers:

- **Content-Type**: application/json
- **Accept**: application/json

- **Accept**: Language: en
- **Authorization**: Bearer {ACCESS_TOKEN}

Before calling FTS API, you need to get the ACCESS_TOKEN use step Retrieving Access Client Token.

How to Use FTS API to find Object Storage Prefixes

First you need to get the ACCESS_TOKEN use step Retrieving Access Client Token, then you may call the endpoint List Prefixes as below:

Sample Request:

```
curl --request GET https://rex.retail.<Region Name>.ocs.oraclecloud.com/<Customer
Subnamespace>/ /siocs-int-services/api/fts/vvvvv-siocs/listprefixes -H 'content-
type: application/json' -H 'Accept: application/json' -H 'Accept-Language: en' -H
"Authorization: Bearer ${ACCESS_TOKEN}"
```

Sample Response:

```
{"values":["archives","rejects","imports","exports"]}
```

How to Use FTS APIs to Upload Files to Object Storage

- Step1: Request upload PAR
- Step2: Use PAR to upload data files to Object Storage

Step1: Request upload PAR

First get the ACCESS_TOKEN use step Retrieving Access Client Token, then call the endpoint Request Upload PAR as below:

Sample Request:

```
curl --request POST https://rex.retail.<Region Name>.ocs.oraclecloud.com/
<Customer Subnamespace>/ /siocs-int-services/api/fts/{bucketname}/upload -H
'content-type: application/json' -H 'Accept: application/json' -H 'Accept-
Language: en' -H "Authorization: Bearer ${ACCESS_TOKEN}" -d "{\"listOfFiles\":
[{\"storagePrefix\": \"imports\",\"fileName\": \"EXTPC_1.dat\"},
{\"storagePrefix\": \"imports\",\"fileName\": \"RFID_1.dat\"}]}"
```

Sample Response:

```
{"par-List":[{"id":"zzzzzzz/:imports/
EXTPC_1.dat","name":"EXTPC_1.dat","accessUri":"https://objectstorage.us-ZZZ-
siocs/o/imports/
EXTPC_1.dat","accessType":"ObjectWrite","timeExpires":"2022-02-13T21:39:40.265Z",
"timeCreated":"2022-02-13T21:34:40.329Z","objectName":"imports/EXTPC_1.dat"},
{"id":"ZZZZ:imports/RFID_1.dat","name":"RFID_1.dat","accessUri":"https://zzzz-
siocs/o/imports/
RFID_1.dat","accessType":"ObjectWrite","timeExpires":"2022-02-13T21:39:40.411Z","
timeCreated":"2022-02-13T21:34:40.472Z","objectName":"imports/RFID_1.dat"}]}
```

Step2: Use PAR to upload data files to Object Storage

Use the accessUri returned in the get PAR response to upload the data file.

Sample Request:

```
curl https://ZZZZZ-siocs/o/imports/RFID_1.dat --upload-file C:\\temp\\RFID_1.dat
```

How to Use FTS API to List Files in Object Storage

First get the ACCESS_TOKEN using step Retrieving Access Client Token, then call the endpoint List Files as below:

Sample Request:

```
curl --request GET https://<external_load_balancer>/<cust_ env>/siocs-int-
services/api/fts//<bucketname>/listfiles?contains=RFID -H 'content-type:
application/json' -H 'Accept: application/json' -H 'Accept-Language: en' -H
"Authorization: Bearer ${ACCESS_TOKEN}"
```

Sample Response:

```
{"lim-it":999,"count":1,"offset":0,"hasMore":false,"resultSet":[{"name":"imports/
RFID_1.dat","createdDate":"2022-02-13T21:35:26Z","modifiedDate":"2022-02-13T21:35
:26Z","scanStatus":"Passed","scanDate":"2022-02-13T21:35:56.187Z","md5":"xxxx==",
"version":"xxxxx","etag":"zzzzzzz","size":75}]}
```

How to Use FTS APIs to Download Files from Object Storage

- Step1: Find what files are available for downloads
- Step2: Request Download PAR for downloading data files from Object Storage
- Step3: Download the file using the par returned from step2

Step1: Find what files are available for downloads

First get the ACCESS_TOKEN using step Retrieving Access Client Token, then call the endpoint List Files as below:

Sample Request:

```
curl --request GET https://<external_load_balancer>/<cust_ env>//siocs-int-
services/api/fts/<bucketname>/listfiles?contains=RFID -H 'content-type:
application/json' -H 'Accept: application/json' -H 'Accept-Language: en' -H
"Authorization: Bearer ${ACCESS_TOKEN}"
```

Sample Response:

```
{"lim-it":999,"count":1,"offset":0,"hasMore":false,"resultSet":[{"name":"imports/
RFID_1.dat","createdDate":"2022-02-13T21:35:26Z","modifiedDate":"2022-02-13T21:35
:26Z","scanStatus":"Passed","scanDate":"2022-02-13T21:35:56.187Z","md5":"xxxxx==",
"version":"xxxxx","etag":"ZZZZZ","size":75}]}
```

Step2: Request Download PAR for downloading data files from Object Storage

First get the ACCESS_TOKEN using step Retrieving Access Client Token, then call the endpoint Request Download PAR as below:

Sample Request:

```
curl --request POST https://ZZZZZZ-siocs/siocs-int-services/internal/fts/
rgbu_rex_cndevcorp_rgbu-rex-rgbu-dev125-siocs/download -H 'content-type:
application/json' -H 'Accept: application/json' -H 'Accept-Language: en' -H
"Authorization: Bearer ${ACCESS_TOKEN}" -d "{\"listOfFiles\":
[{\"storagePrefix\": \"imports\",\"fileName\": \"RFID_1.dat\"}]}"
```

Sample Response:


```
{"par-List":[{"id":"i91P0nFIIsgj05qrUH2ibTZ2npmbTdq1TKsGtWOerAYaE6/MYZE7840lR/
QEhaFk:imports/RFID_1.dat","name":"RFID_1.dat","accessUri":"https://
objectstorage.us-phoenix-1.oraclecloud.com/p/
ZG89KsLS_5SY7D2p7nVQt8KfJ6rLJ40FSmI97zASLRK2VrsICbvoRP0bgoQGxk3S/n/ZZZZZ-siocs/o/
imports/RFID_1.dat","accessType":"ObjectRead","timeEx-
pires":"2022-02-13T23:07:00.962Z","timeCreated":"2022-02-13T23:02:01.105Z","objec
tName":"imports/RFID_1.dat"}]}
```


Step3: Download the file using the par returned from step2

```
curl -o <destinationFileName> -X GET <PAR>
```

For example:

```
curl -o RFID_1_download.dat -X GET https://ZZZZZ-siocs/o/imports/RFID_1.dat
```

Handling Import Data Files

This section describes the general steps for an external solution application to transfer batch data files from external system to cloud application object storage.

The data to be processed can be provided as a single data file, or a zip file contains multiple data files.

The application batch imports the inbound data files from Object Storage, after the files have passed an anti-virus and malware scan. Once the files are downloaded from Object Storage, the batch process deletes the files from Object Storage to ensure it is not re-processed in next batch run. Rejected records are placed in the rejects file when applicable.

Supported Import Data Files

Table 9-2 Supported Import Data Files

File Name	Description	File Layout
Clearance File Import	The file is processed by Clearance File Import Batch. For additional details, see Batches.	Filename Format: Clearance_Tx_<YYYYMMddHHMMss>.csv See Appendix: Batch File Layout Specifications for details.
Initial Inventory Import File	The file is processed by Initial Inventory Import Batch. For additional details, see Batches.	File name prefix: EXTSTK_* See Appendix: Batch File Layout Specifications for details.

Table 9-2 (Cont.) Supported Import Data Files

File Name	Description	File Layout
Price Change File Import	The file is processed by Price Change File Import Batch. For additional details, see Batches.	Filename Format: PriceChange_Tx_<YYYYMMddHHMMss>.csv See Appendix: Batch File Layout Specifications for details.
ReSA Import File	The file is processed by Retail Sale Audit Import Batch. For additional details, see Batches.	Zip Filename Format SIMT_< YYYYMMDDHH24MISS>.zip See Appendix: Batch File Layout Specifications for details.
RFID Import File	The file is processed by Third Party RFID Import Batch. For additional details, see Batches.	Zip Filename Format RFID_<YYYYMMDDHH24MISS>.zip See Appendix: Batch File Layout Specifications for details.
Store Sequence Import	The file is processed by Store Sequence Import Batch. For additional details, see Batches.	Filename Format: SSEQ date in YYYYMMDDHH24MISS format>_<loc id>.dat See Appendix: Batch File Layout Specifications for details.
Third Party Price Import File	The file is processed by Third Party Price File Import Batch. For additional details, see Batches.	Zip Filename Format EXTPC_<YYYYMMDDHH24MISS>.zip See Appendix: Batch File Layout Specifications for details.
Third Party Stock Count Import File	The file is processed by Third Party Stock Count Import Batch. For additional details, see Batches.	Zip Filename Format STK_<YYYYMMDDHH24MISS>.zip See Appendix: Batch File Layout Specifications for details.

Upload Import Data Files to Object Storage

To upload data files to object storage, the external solution application needs to perform following steps:

1. The external application gets the Oauth2 token from IDCS.
2. The external application makes an FTS request with the Oauth2 token to requesting Pre-Authentication.
3. Once the PAR is received, the external application uploads the file to object storage using the URL included within the response.
4. Files uploaded to application object storage will be processed by cloud application batches.

Handling Export Data Files

The following describes the supported export data files which are supported by cloud application. These export data files are available for external solution applications to download.

Supported Export Data Files

Table 9-3 Supported Export Data Files

Export File Name	Description	File Name Format
Inventory Extract File	The file is generated by via Inventory export batch. For additional details, see Batches.	Filename Format: PRODUCT_LOCATION_INV_* See Appendix: Batch File Layout Specifications for details.
Stock Count Export File	The stock count export file is generated when a unit and amount stock count authorization is completed.	Zip Filename Format STK_* See Appendix: Batch File Layout Specifications for details.

Steps to Download Export Data Files from Object Storage

For retailer to download the export data files from application object storage, perform the following steps:

1. The external solution application gets the Oauth2 token from IDCS.
2. The external solution application calls the FTS service with the Oauth2 token to list the available export files in Object Storage which are generated by cloud app.
3. The external solution application calls the FTS service with the Oauth2 token, requesting Pre-Authentication to download files from object storage used by cloud app.
4. Once the PAR is received, the file is downloaded using the URL included within its response. A PAR is valid for 10 minutes. A download can take longer than 10 minutes, but it must be started within 10 minutes of the PAR being received.

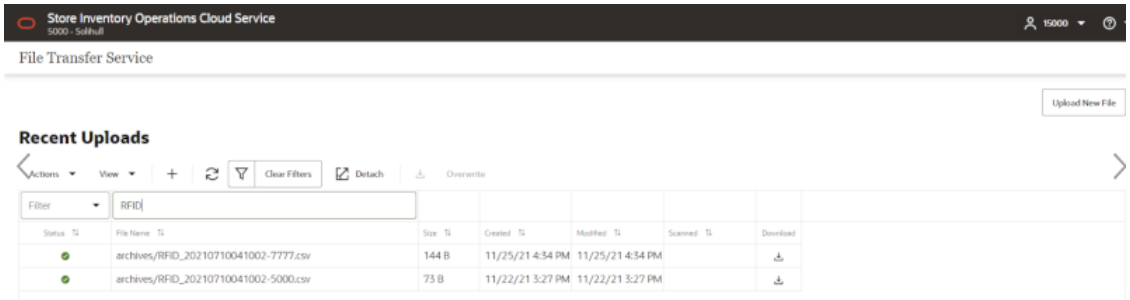
File Transfer Service UI

SIOCS provides an UI which is used to upload or download a file or view a list of files in object storage.

To access this screen, the application user needs to be assigned the **Access File Transfer Service** security permission.

The IDCS or OCI IAM application role *admin_users* is required for the user to perform the upload/download operations.

Figure 9-5 File Transfer Service UI

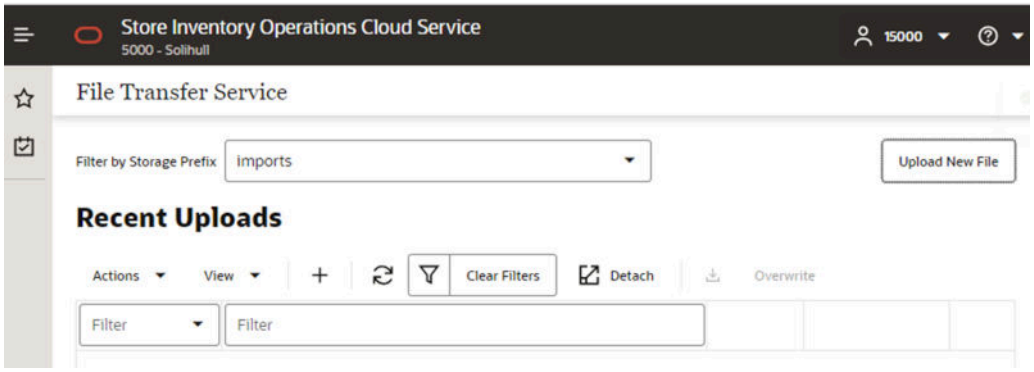


The main form lists the recently uploaded files.

Actions:

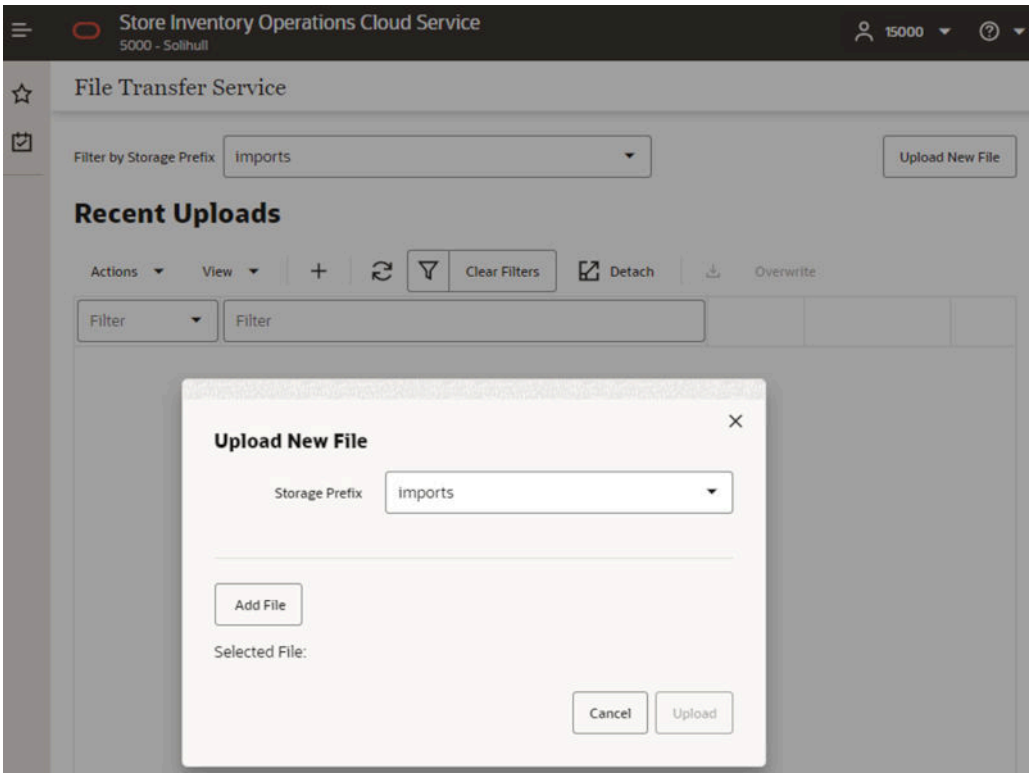
- To filter the files by store prefix, select a file storage prefix.
- To filter by file name by choosing the **Actions** choice selector on the screen.
- To upload new files, click **Upload New File** button:

Figure 9-6 Upload New File



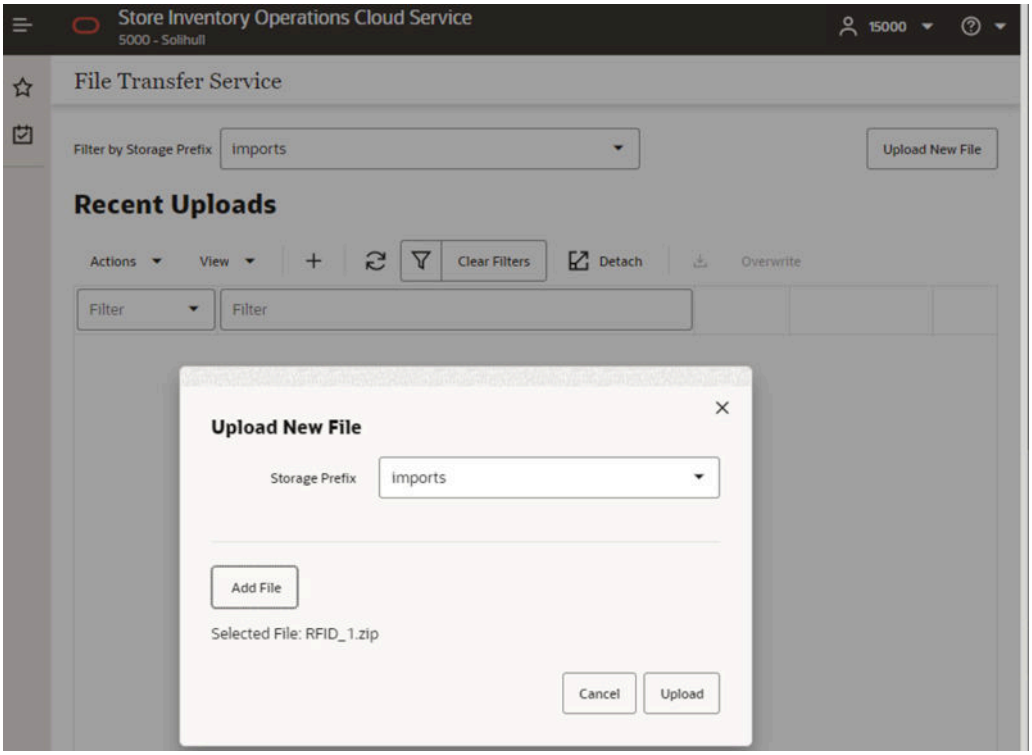
In the **Upload New File** popup dialog, choose storage prefix **Imports** and click **Add File** button.

Figure 9-7 Upload New File Dialog



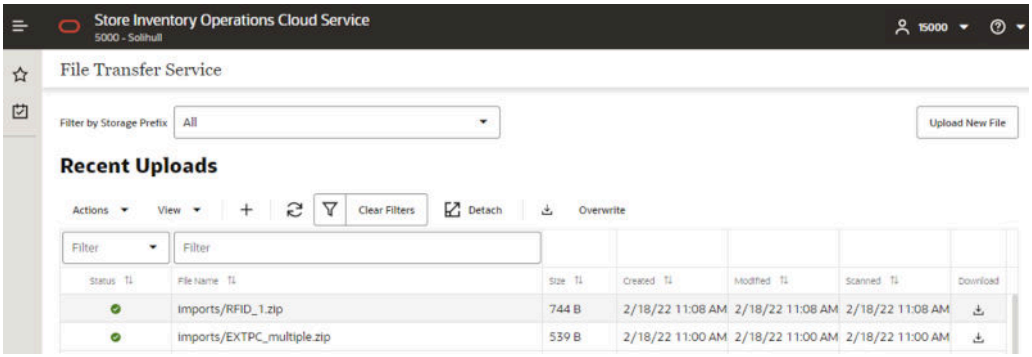
Next, choose files from your client machine, then click **Upload**:

Figure 9-8 File Added



Once the uploaded file has passed a virus scan, the file is ready for a cloud application batch to import the file from object storage into the application.

Figure 9-9 Recent Uploads



Note:

The uploaded import data files will be processed by scheduled batch import job. You may run an adhoc import batch job for testing purpose, if choose so, make sure to run the adhoc job outside of job schedule window for the select batch (or disable the job schedule for the selected batch. Once the adhoc job is completed, you will need to re-enable the batch schedule for the batch).

FTS API Specifications

This section describes FTS API specifications.

- Ping
- List Prefixes
- List Files
- Move Files
- Delete Files
- Request Upload PAR
- Request Download PAR

Ping

Returns the status of the service and provides an external health-check.

Method	**GET**
Endpoint	{Service Base URL}/fts/ping
HTTP Header	See Common Request Headers in making FTS API Call Common Headers.
Parameters	[{ "name": "pingMessage", "description": "Optional value to be included in the ping response.", "in": "query", "required": false, "schema": { "type": "string" } }],
Request Body	None

Response	"200": {
	"description": "OK - The service operation produced a successful response."
	},
	"400": {
	"description": "Bad Request - The path params or query params or body was not valid for this operation."
	}

List Prefixes

Returns a list of the known storage prefixes. These are analogous to directories and are restricted to predefined choices per service. SIOCS has list of pre-defined storage prefixes: import, exports, rejects and archives.

Method	**GET**
Endpoint	{Service Base URL}/fts/{FTS Bucket Name}/listprefixes
HTTP Header	See Common Request Headers in making FTS API Call Common Headers.

Method	GET

Parameters	[
	{
	"name": "bucketName",
	"description": "Bucket identifier.",
	"in": "path",
	"required": true,
	"schema": {
	"type": "string"
	}
	}
	],

Request Body	None

Response	A JSON array of strings containing the known prefixes.
	{
	"200": {
	"description": "OK - The service operation produced a successful response."
	},
	"400": {
	"description": "Bad Request - The path params or query params or body was not valid for this operation."
	}
	}

List Files

Returns a list of the files within a given storage prefix.

Method	GET
Endpoint	{Service Base URL}/fts/{FTS Bucket Name}/listfiles
HTTP Header	See Common Request Headers in making FTS API Call Common Headers.

Method	GET
Parameters	{ "name": "bucketName", "description": "Bucket identifier.", "in": "path", "required": true, "schema": { "type": "string" } }, { "name": "prefix", "description": "The object filter in object storage.", "in": "query", "required": false, "schema": { "type": "string" } }, { "name": "contains", "description": "The object filter in object storage.", "in": "query", "required": false, "schema": { "type": "string" } }, { "name": "scanStatus", "description": "The object filter in object storage.", "in": "query", "required": false, "schema": { "type": "string" } },

Method	GET

	{
	"name": "offset",
	"description": "The object filter in object storage.",
	"in": "query",
	"required": false,
	"schema": {
	"type": "integer"
	}
	},
	{
	"name": "limit",
	"description": "The object filter in object storage.",
	"in": "query",
	"required": false,
	"schema": {
	"type": "integer"
	}
	},
	{
	"name": "sort",
	"description": "The object filter in object storage.",
	"in": "query",
	"required": false,
	"schema": {
	"type": "string"
	}
	}
	],

Request Body	None

Response	A JSON resultSet containing array of files. For each file, there is metadata including: name, size, created and modified dates, scan status and date, scan output message.
	{
	"200": {
	"description": "OK - The service operation produced a successful response."
	},
	"400": {
	"description": "Bad Request - The path params or query params or body was not valid for this operation."
	}

Move Files

Moves one or more files between storage prefixes, while additionally allowing the name to be modified.

Method	POST
Endpoint	{Service Base URL}/fts/{FTS Bucket Name}/movefiles
HTTP Header	See Common Request Headers in making FTS API Call Common Headers.

Parameters	[
	{
	"name": "bucketName",
	"description": "Bucket identifier.",
	"in": "path",
	"required": true,
	"schema": {
	"type": "string"
	}
	}
	]

Request Body	{"listOfFiles": [{"currentPath": { "storagePrefix": "string", "fileName": "string"}, "newPath": { "storagePrefix": "string", "fileName": "string" } } }}

Delete Files

Deletes one or more files.

Method	POST
Endpoint	{Service Base URL}/fts/{FTS Bucket Name}/delete
HTTP Header	See Common Request Headers in making FTS API Call Common Headers.

Parameters	[
	{
	"name": "bucketName",
	"description": "Bucket identifier.",
	"in": "path",
	"required": true,
	"schema": {
	"type": "string"
	}
	}
	]

Request Body	A JSON array of files to be deleted. One or more pairs of storagePrefix and filename elements can be specified within the array. Required: true{ " {"listOfFiles": [[{ " {"storagePrefix": "string", "fileName": "string" }]}

Response	A JSON array of each file deletion attempted and the result.
	{
	"200": {
	"description": "OK - The service operation produced a successful response."
	},
	"400": {
	"description": "Bad Request - The path params or query params or body was not valid for this operation."

Request Upload PAR

Request PAR for uploading one or more files.

Method	POST
Endpoint	{Service Base URL}/fts/{FTS Bucket Name}/upload
HTTP Header	See Common Request Headers in making FTS API Call Common Headers.

Parameters	[
	{
	"name": "bucketName",
	"description": "Bucket identifier.",
	"in": "path",
	"required": true,
	"schema": {
	"type": "string"
	}
	}]

Request Body	A JSON array of files to be uploaded. One or more pairs of storagePrefix and filename elements can be specified within the array.
	Required: true
	{ "listOfFiles":
	[
	{
	"storagePrefix": "string",
	"fileName": "string"
	}
	]
	}

Method	POST

Response	A parList containing an array containing elements corresponding to the request including the PAR accessUri and name of file.
	{
	"parList": [
	{
	"id": "string",
	"name": "string",
	"accessUri": "string",
	"objectName": "string",
	"accessType": "string",
	"timeExpires": "2021-09-07T16:35:27.390Z",
	"timeCreated": "2021-09-07T16:35:27.390Z"
	}
	]
	}
	Response Status:
	{
	"200": {
	"description": "OK - The service operation produced a successful response."
	},
	"400": {
	"description": "Bad Request - The path params or query params or body was not valid for this operation."
	}

Request Download PAR

Request PAR for downloading one or more files.

Method	POST
Endpoint	{Service Base URL}/fts/{Bucket Name}/download
HTTP Header	See Common Request Headers in making FTS API Call Common Headers.

Parameters	[
	{
	"name": "bucketName",
	"description": "Bucket identifier.",
	"in": "path",
	"required": true,
	"schema": {
	"type": "string"
	}
	}
	]

Request Body	A JSON array of files to be downloaded. One or more pairs of storagePrefix and filenames can be specified within the array. Required: true
	{
	"listOfFiles": [
	{
	"storagePrefix": "string",
	"fileName": "string"
	}
	]
	}

Response	A parList containing an array containing elements corresponding
	to the request including the PAR accessUri and name of file.
	"parList": [
	{
	"id": "string",
	"name": "string",
	"accessUri": "string",
	"objectName": "string",
	"accessType": "string",
	"timeExpires": "2021-09-07T16:35:27.390Z",
	"timeCreated": "2021-09-07T16:35:27.390Z"
	}
	]
	}
	Response Status:
	{
	"200": {
	"description": "OK - The service operation produced a successful response.
	"

File Transfer Service Troubleshooting

These troubleshooting topics covers common file transfer service issues and possible solutions.

Troubleshooting File Transfer Service Internal Server Error

1. Try to connect to File Transfer Ping endpoint. If you can connect ping endpoints, continue to step2.
2. Try to invoke List Files endpoint, if get response status 200, continue to step3.
3. Verify the bucket name. The bucket name should have value like rgbu_rex_cnprod_<cust_env>
4. Make sure the bucket name in service request matches the configuration value set for 'File Transfer Service Bucket Name' in the system configuration screen.

   If the above steps do not resolve the internal server error, you may raise a Service Request on My Oracle Support.

Test FTS API using Postman

- Step 1: Get Client Access Token
- Step 2: Call FTS Endpoints

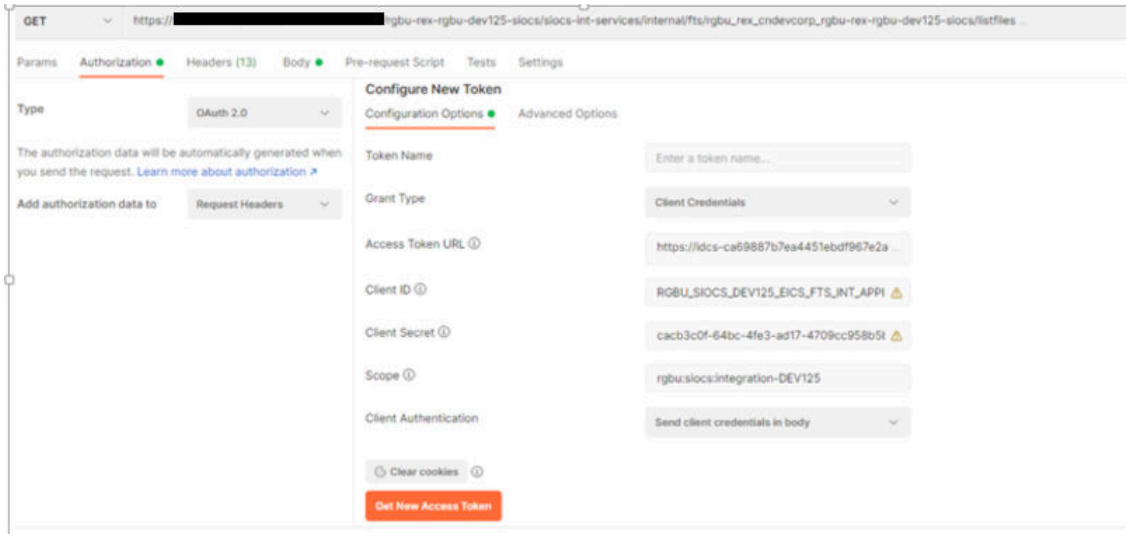
Step 1: Get Client Access Token

OAuth tokens can also be obtained by REST client tools like postman for testing purposes.

When using Postman testing, fill in the required details:

- **Authorization**: OAuth 2.0
- **Access Token URL**: https://{IDCS_BASE_URL}/oauth2/v1/token
- **Client ID**: Client if of the OAuth
- **Client Secret**: Client secret of OAuth Client app
- **Grant Type**: client_credentials
- **Scope**: The scope pattern that is used in the FTS IDCS app creation template is rgbu:siocs:integration-{env}{env index}

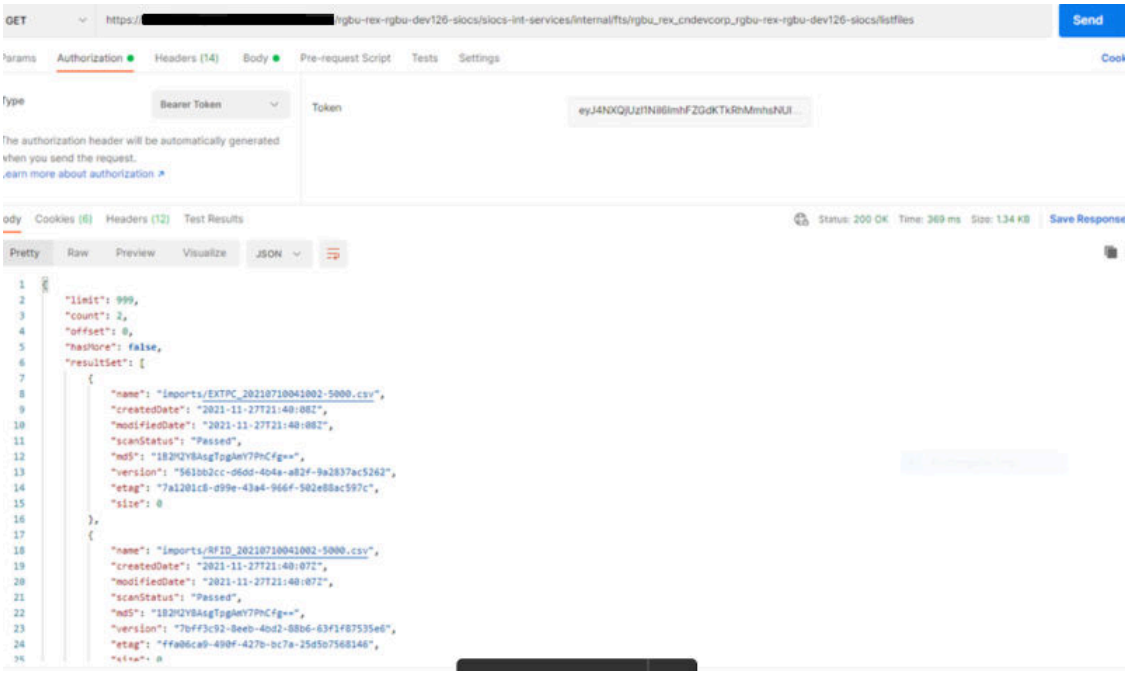
Figure 9-10 Get Client Access Token



Step 2: Call FTS Endpoints

Change **Authorization Type** to **Bearer Token**, use the access token returned from step1 as the **Token Value** as below:

Figure 9-11 Call FTS Endpoints



A

Appendix: Report Formats

Reports

Table A-1 Reports

Report Name	Report Parameters
Figure A-1	pick_id, copies
Figure A-2	delivery_id, store_timezone, locale_id, copies
Figure A-3	delivery_id, store_timezone, locale_id, copies
Figure A-4	pick_id, store_timezone, locale_id, copies
Figure A-5	pick_id, store_timezone, locale_id, copies
Figure A-6	order_id, store_timezone, locale_id, copies
Figure A-7	reverse_pick_id, store_timezone, locale_id, copies
Figure A-8	receipt_id, store_timezone, locale_id, copies
Figure A-9	receipt_id, store_timezone, copies
Figure A-10	Inv_Adjust_ID, copies
Figure A-11	inv_adj_id, store_timezone, copies
Figure A-12	Item_basket_id, store_timezone, locale_id, copies
Figure A-13	itemid, storeid, store_timezone, locale_id, copies
Figure A-14	purchase_order_id, store_timezone, locale_id, copies
Figure A-15	replenish_gap_id, copies, store_timezone, locale_id
Figure A-16	item_id, from_date, to_date, locale_id, copies
Figure A-17	return_id, store_timezone, locale_id, copies
Figure A-18	shelf_adjust_id, store_timezone, locale_id, copies
Figure A-19	shelf_replenish_id, store_timezone, locale_id, copies
Figure A-20	store_id, stock_count_id, copies
Figure A-21	stock_count_id, copies
Figure A-22	stock_count_id, store_id, copies
Figure A-23	stock_count_id, stock_count_child_id, phase, store_timezone, locale_id, copies
Figure A-24	store_order_id, store_timezone, locale_id
Figure A-25	carton_ID, copies
Figure A-26	delivery_id, store_timezone, locale_id, copies
Figure A-27	carton_id, locale_id,
Figure A-28	delivery_id , store_timezone, locale_id, , copies
Figure A-29	transfer_id, store_timezone, locale_id, copies

Table A-1 (Cont.) Reports

Report Name	Report Parameters
Figure A-30	shipment_id, store_timezone, locale_id, copies
Figure A-31	carton_id, store_timezone, locale_id, copies
Figure A-32	shipment_id, store_timezone, locale_id, copies
Figure A-33	carton_id, locale_id
Figure A-34	carton_id, copies
Figure A-35	carton_id, locale_id
Figure A-36	ship_number, store_timezone, locale_id, copies
Figure A-37	carton_id, store_timezone, locale_id, copies
Figure A-38	ship_number, store_timezone, locale_id, copies
Figure A-39	carton_id, locale_id, copies

Report Formats

The following section describes the report formats.

Figure A-1 Customer Order Bin Label Report

Figure A-2 Customer Order Delivery BOL Report


Customer Order Delivery BOL Report

BOL ID: 1158 Create Date: 08/09/2017
Delivery ID: 274
Customer Order ID: New

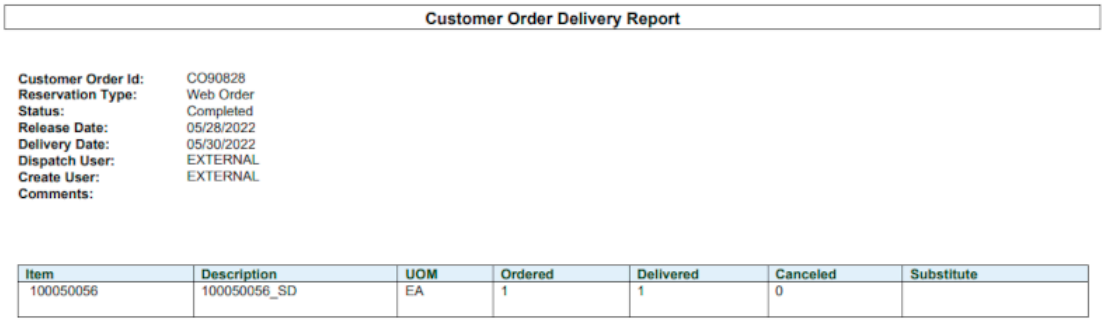
Item	Description	Substitute	UOM	Qty	Price	Amount
SIM_107	SIM_107		EA	3	125	375

Delivery Charge Total Lines	USD8.99 1
Legalese Fine Print	
Comments	

CANCELED

Printed: 1/8/2018 Page Number: 1

Figure A-3 Customer Order Delivery Report


Customer Order Delivery Report

Customer Order Id: CO90828
Reservation Type: Web Order
Status: Completed
Release Date: 05/28/2022
Delivery Date: 05/30/2022
Dispatch User: EXTERNAL
Create User: EXTERNAL
Comments:

Item	Description	UOM	Ordered	Delivered	Canceled	Substitute
100050056	100050056_SD	EA	1	1	0	

Figure A-4 Customer Order Pick Discrepancy Report

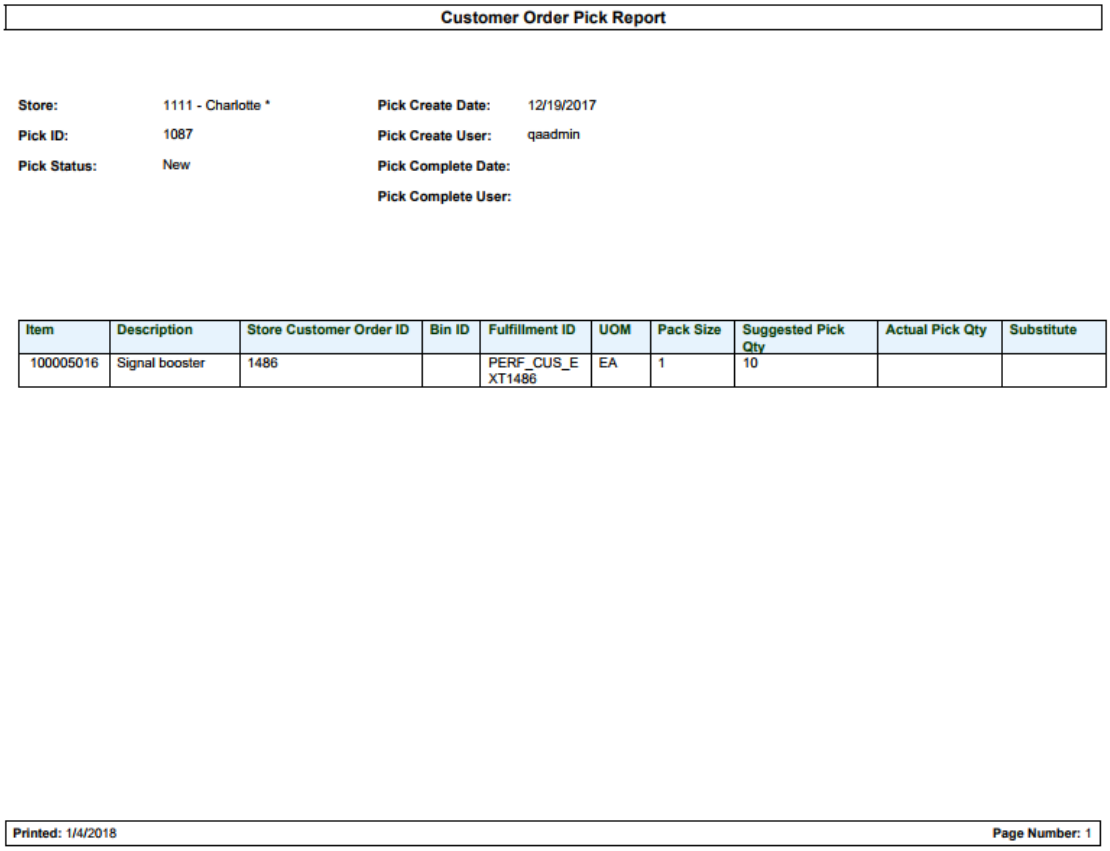
Customer Order Pick Discrepancy Report

Store: 1511 - Phoenix
Pick Create Date: 12/19/2017
Pick ID: 1086
Pick Create User: QAADMIN
Pick Status: In Progress

Item	Description	Store Customer Order ID	Bin ID	Fulfillment ID	UOM	Pack Size	Old Pick Qty	Adjusted Pick Qty
100005016	Signal booster	301	142	LG1	EA	1	2	0

Printed: 1/4/2018 **Page Number:** 1

Figure A-5 Customer Order Pick Report



Customer Order Pick Report

Store: 1111 - Charlotte *
Pick ID: 1087
Pick Status: New
Pick Create Date: 12/19/2017
Pick Create User: qaadmin
Pick Complete Date:
Pick Complete User:

Item	Description	Store Customer Order ID	Bin ID	Fulfillment ID	UOM	Pack Size	Suggested Pick Qty	Actual Pick Qty	Substitute
100005016	Signal booster	1486		PERF_CUS_EXT1486	EA	1	10		

Printed: 1/4/2018 **Page Number:** 1

Figure A-6 Customer Order Report

Customer Order Report

Store: 1311
Store Customer Order Id: 21
Customer Order Id: Pick3
Fulfillment Order Id: Pick3
Status: Canceled
Reservation Type: Web Order
Comments: Testing the External Comments. Do they work?

Create Date: 04-20-2022
Release Date: 03-30-2022
Delivery Date: 04-01-2022

Delivery Type: Ship To Customer
Carrier: Other
Service:
Allow Partial Delivery: Yes

Item	Description	UOM	Order Qty	Picked Qty	Delivered Qty	Canceled Qty	Last Update Date	Comments	Substitute
100000147	100000147_SD	EA	2	0	0	2	05-30-2022	Test comments.	

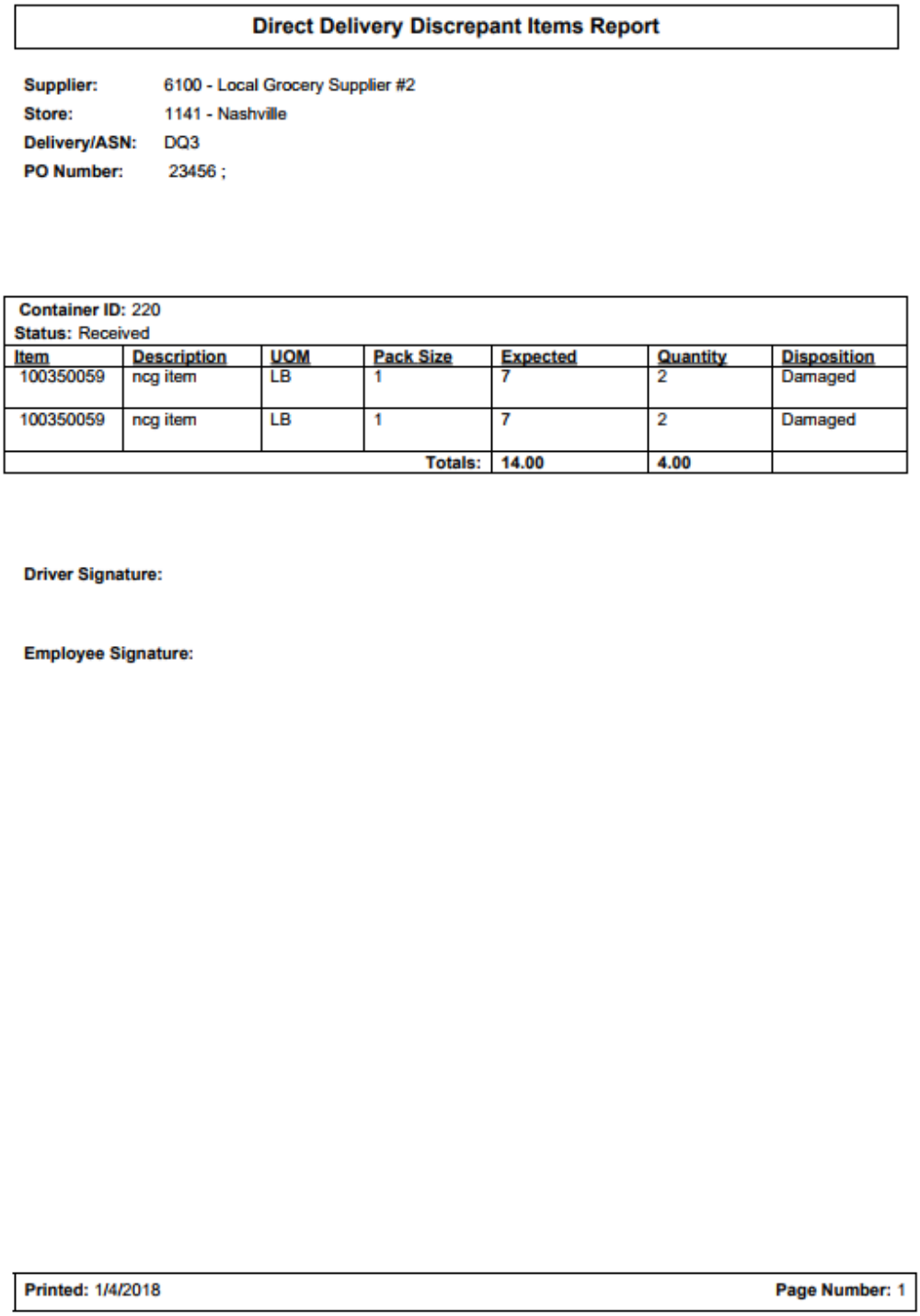
Figure A-7 Customer Order Reverse Pick Report

Customer Order Reverse Pick Report

Store:	1311			
Reverse Pick ID:	61	**Order Status:**	Canceled	
Store Customer Order ID:	77	**Reverse Pick Status:**	Completed	
Customer Order ID:	CO26621	**Reservation Type:**	Web Order	**Comments:**
Fulfillment Order ID:	CO15217	**Reverse Pick Create Date:**	05/31/2022	
Create User:	siocssysop-qa20			

Item	Description	UOM	Order Qty	Picked Qty	Delivered Qty	Canceled Qty	Sugg. Reverse Qty	Qty	Substitute
100050056	100050056_SD	EA	1	0	0	1	1	1	

Figure A-8 Direct Delivery Discrepant Items Report



Direct Delivery Discrepant Items Report

Supplier: 6100 - Local Grocery Supplier #2
Store: 1141 - Nashville
Delivery/ASN: DQ3
PO Number: 23456 ;

Container ID: 220
Status: Received

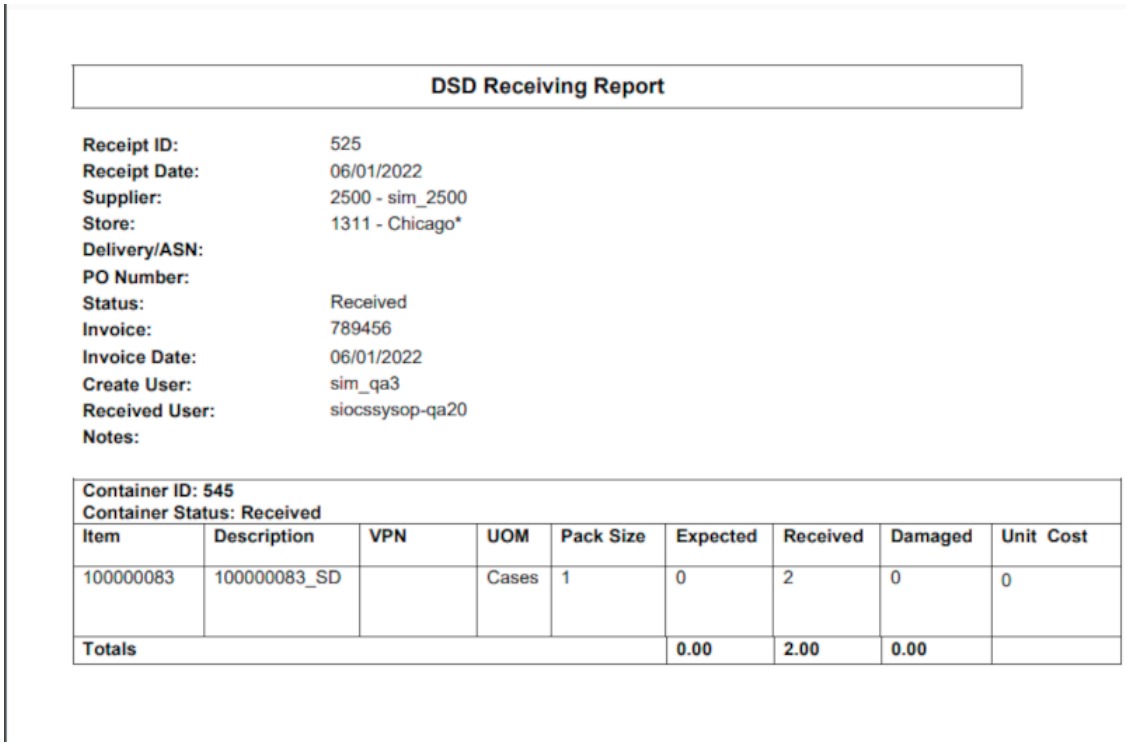
Item	Description	UOM	Pack Size	Expected	Quantity	Disposition
100350059	ncg item	LB	1	7	2	Damaged
100350059	ncg item	LB	1	7	2	Damaged
			Totals:	**14.00**	**4.00**	

Driver Signature:

Employee Signature:

Printed: 1/4/2018 **Page Number:** 1

Figure A-9 Direct Delivery Report



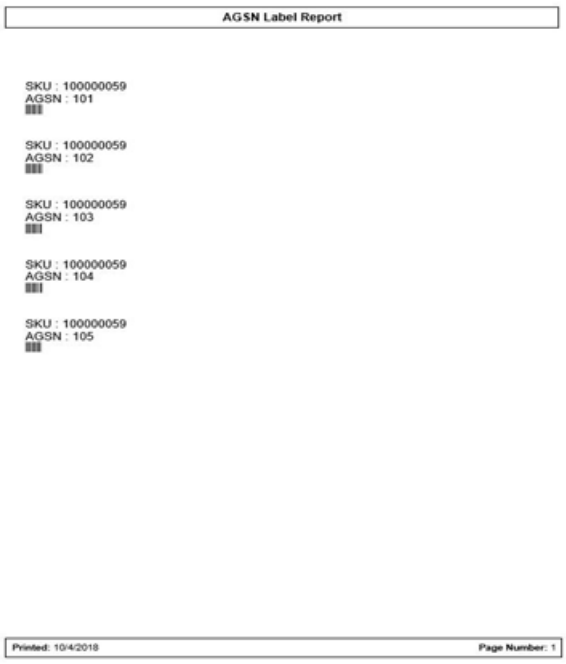
DSD Receiving Report

Receipt ID: 525
Receipt Date: 06/01/2022
Supplier: 2500 - sim_2500
Store: 1311 - Chicago*
Delivery/ASN:
PO Number:
Status: Received
Invoice: 789456
Invoice Date: 06/01/2022
Create User: sim_qa3
Received User: siocssysop-qa20
Notes:

Container ID: 545
Container Status: Received

Item	Description	VPN	UOM	Pack Size	Expected	Received	Damaged	Unit Cost
100000083	100000083_SD		Cases	1	0	2	0	0
Totals					0.00	2.00	0.00	

Figure A-10 Inventory Adjustment AGSN Report



AGSN Label Report

SKU : 100000059
AGSN : 101

SKU : 100000059
AGSN : 102

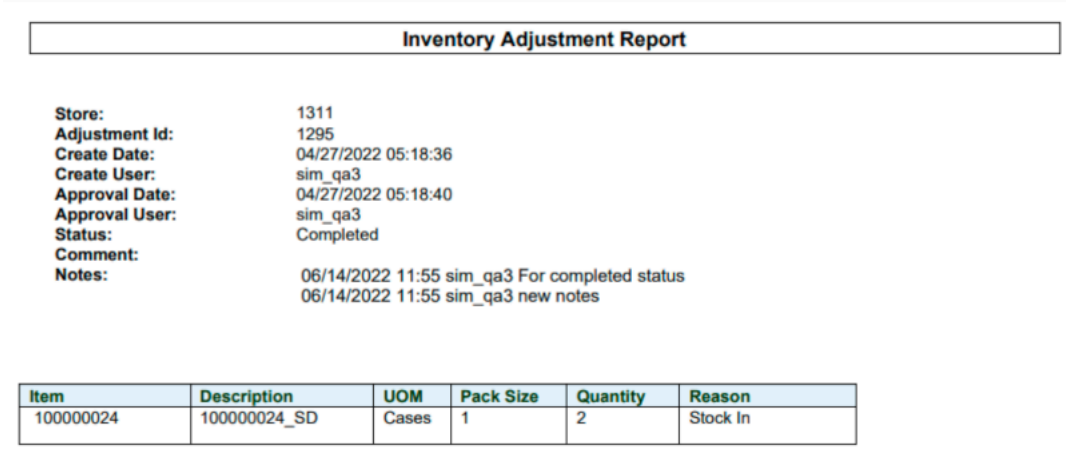
SKU : 100000059
AGSN : 103

SKU : 100000059
AGSN : 104

SKU : 100000059
AGSN : 105

Printed: 10/4/2018 Page Number: 1

Figure A-11 Inventory Adjustment Report

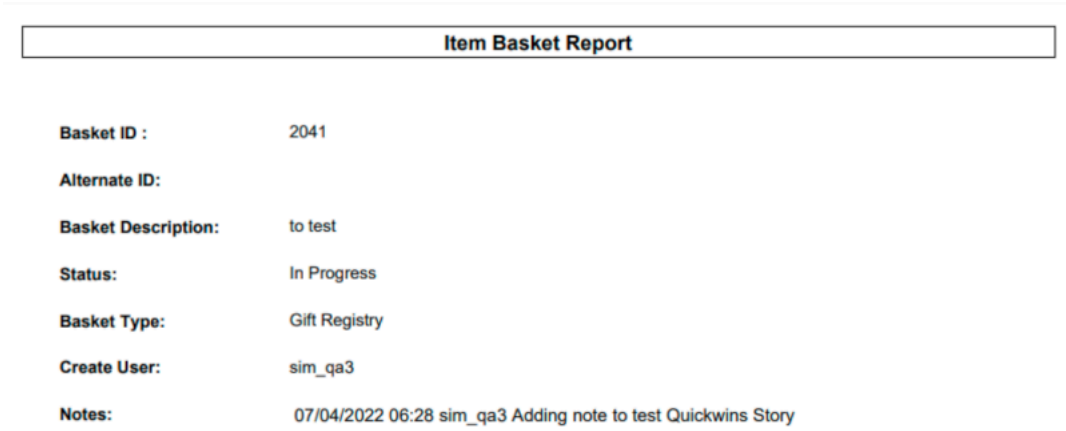


Inventory Adjustment Report

Store: 1311
Adjustment Id: 1295
Create Date: 04/27/2022 05:18:36
Create User: sim_qa3
Approval Date: 04/27/2022 05:18:40
Approval User: sim_qa3
Status: Completed
Comment:
Notes: 06/14/2022 11:55 sim_qa3 For completed status
06/14/2022 11:55 sim_qa3 new notes

Item	Description	UOM	Pack Size	Quantity	Reason
100000024	100000024_SD	Cases	1	2	Stock In

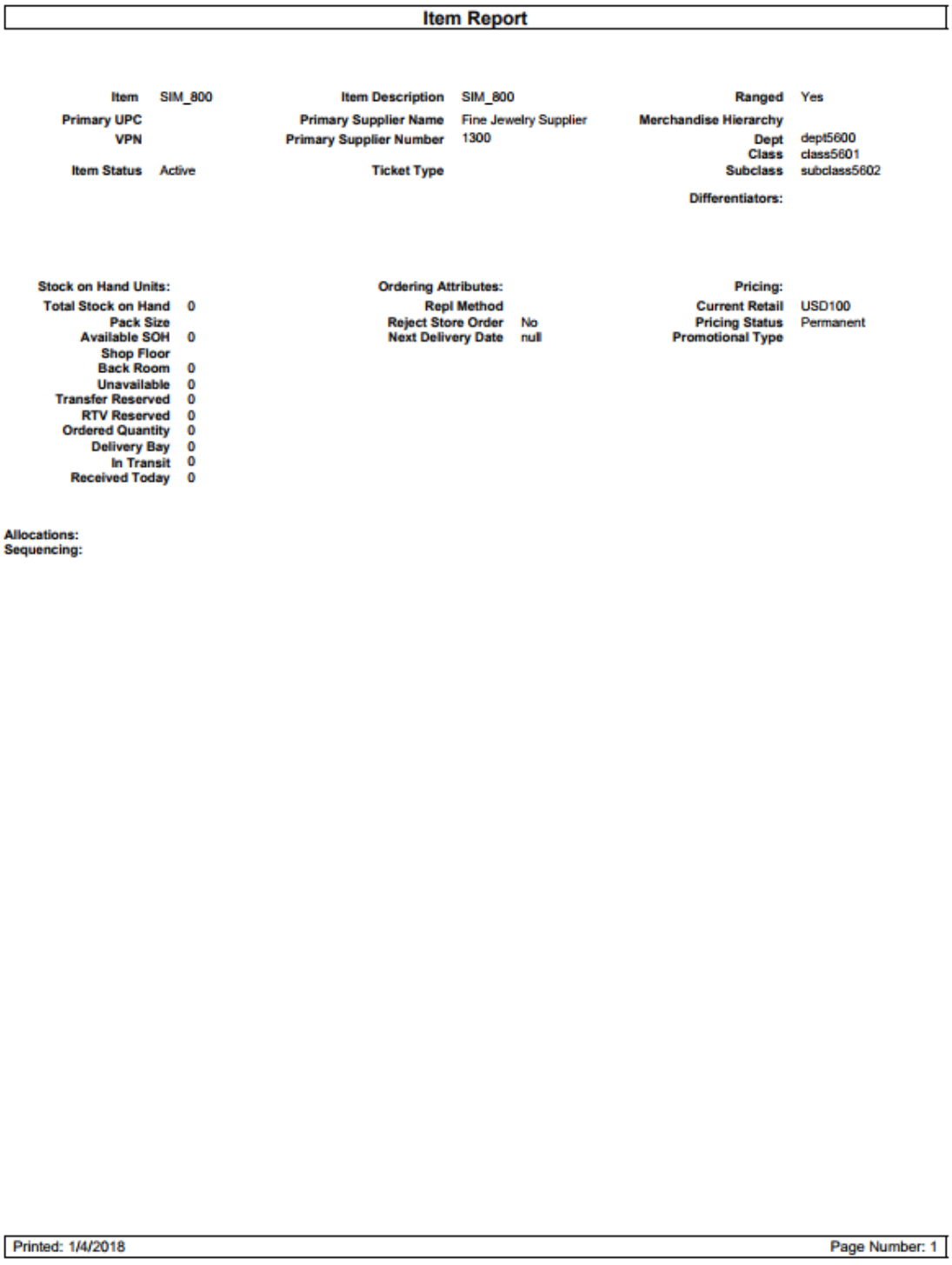
Figure A-12 Item Basket Report



Item Basket Report

Basket ID : 2041
Alternate ID:
Basket Description: to test
Status: In Progress
Basket Type: Gift Registry
Create User: sim_qa3
Notes: 07/04/2022 06:28 sim_qa3 Adding note to test Quickwins Story

Figure A-13 Item Detail Report



Item Report

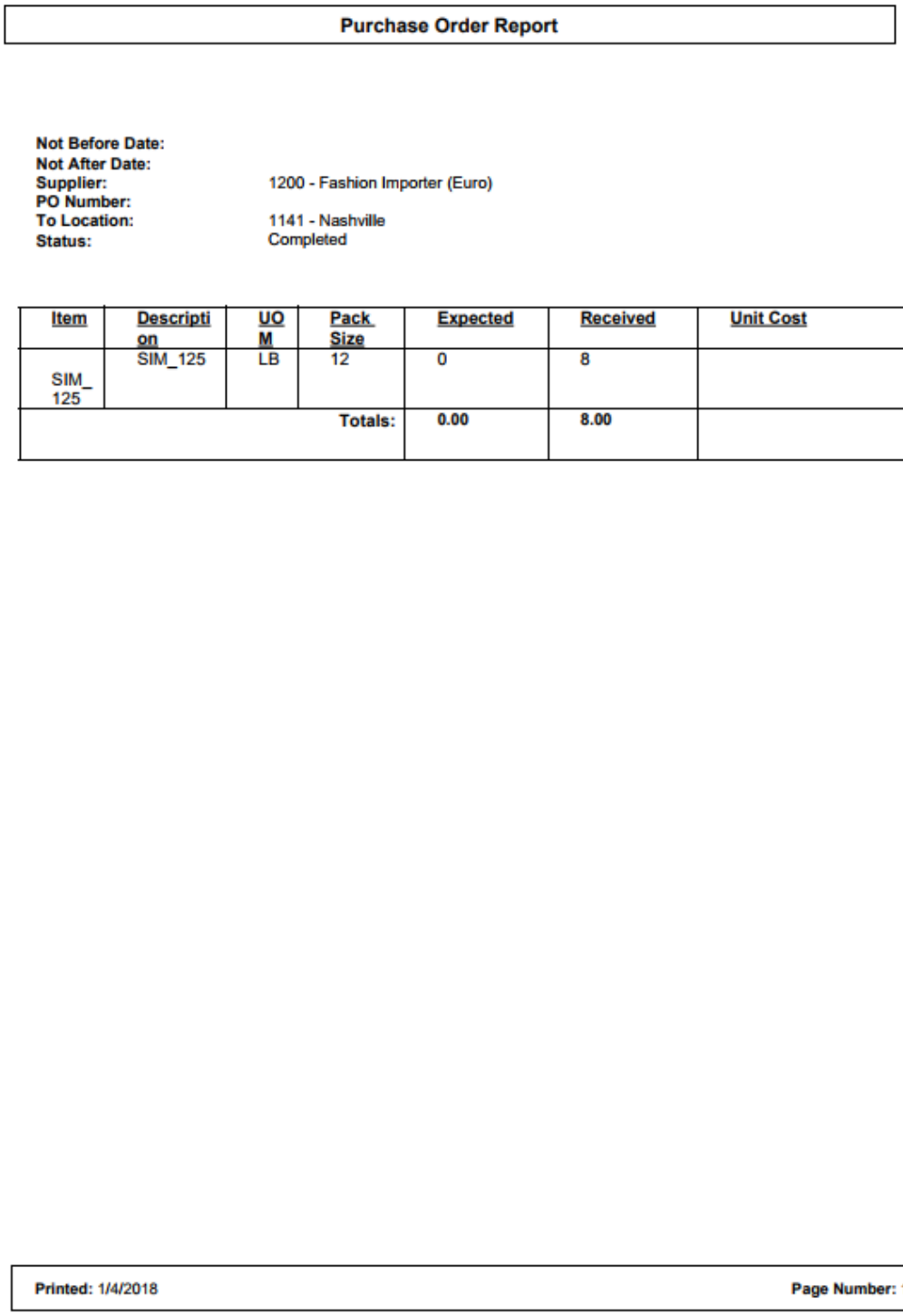
Item	SIM_800	Item Description	SIM_800	Ranged	Yes
Primary UPC		Primary Supplier Name	Fine Jewelry Supplier	Merchandise Hierarchy	
VPN		Primary Supplier Number	1300	Dept	dept5600
				Class	class5601
Item Status	Active	Ticket Type		Subclass	subclass5602
				Differentiators:	

Stock on Hand Units:		Ordering Attributes:		Pricing:	
Total Stock on Hand	0	Repl Method		Current Retail	USD100
Pack Size		Reject Store Order	No	Pricing Status	Permanent
Available SOH	0	Next Delivery Date	null	Promotional Type	
Shop Floor					
Back Room	0				
Unavailable	0				
Transfer Reserved	0				
RTV Reserved	0				
Ordered Quantity	0				
Delivery Bay	0				
In Transit	0				
Received Today	0				

Allocations:
Sequencing:

Printed: 1/4/2018 Page Number: 1

Figure A-14 Purchase Order Report



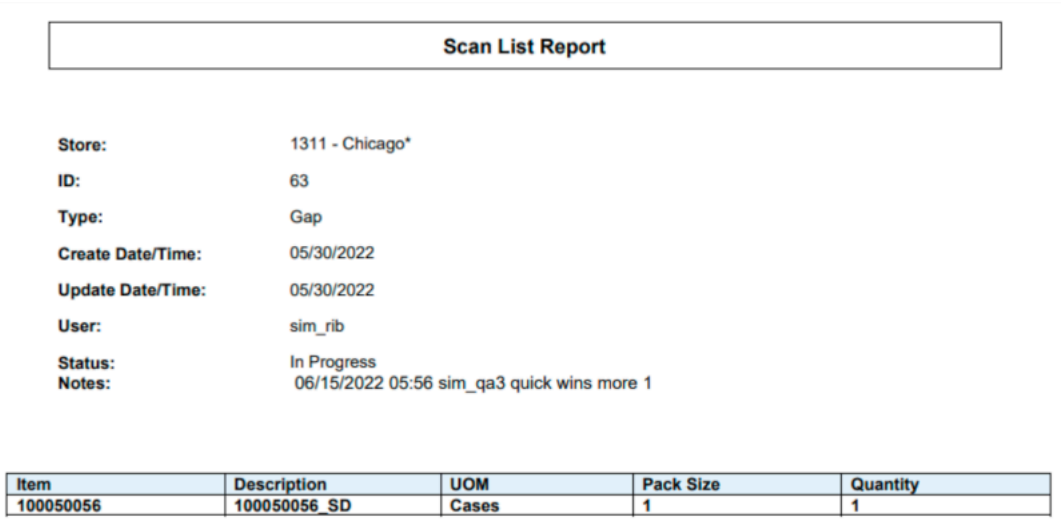
Purchase Order Report

Not Before Date:
Not After Date:
Supplier: 1200 - Fashion Importer (Euro)
PO Number:
To Location: 1141 - Nashville
Status: Completed

Item	Description	UOM	Pack Size	Expected	Received	Unit Cost
SIM_125	SIM_125	LB	12	0	8	
			Totals:	**0.00**	**8.00**	

Printed: 1/4/2018 **Page Number:** 1

Figure A-15 Replenishment Gap Report

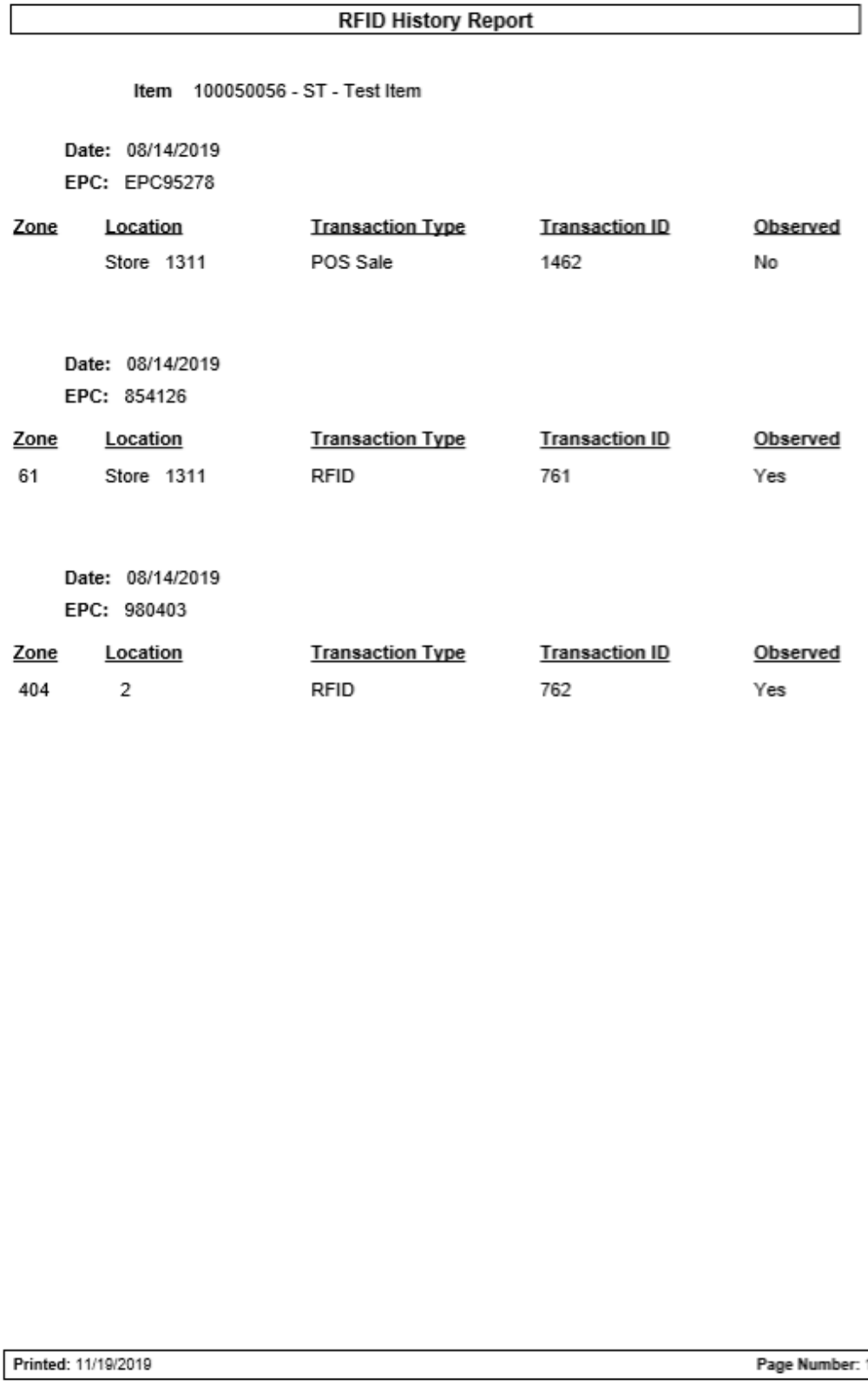


Scan List Report

Store: 1311 - Chicago*
ID: 63
Type: Gap
Create Date/Time: 05/30/2022
Update Date/Time: 05/30/2022
User: sim_rib
Status: In Progress
Notes: 06/15/2022 05:56 sim_qa3 quick wins more 1

Item	Description	UOM	Pack Size	Quantity
100050056	100050056_SD	Cases	1	1

Figure A-16 RFID History Report



RFID History Report

Item 100050056 - ST - Test Item

Date: 08/14/2019
EPC: EPC95278

Zone	Location	Transaction Type	Transaction ID	Observed
	Store 1311	POS Sale	1462	No

Date: 08/14/2019
EPC: 854126

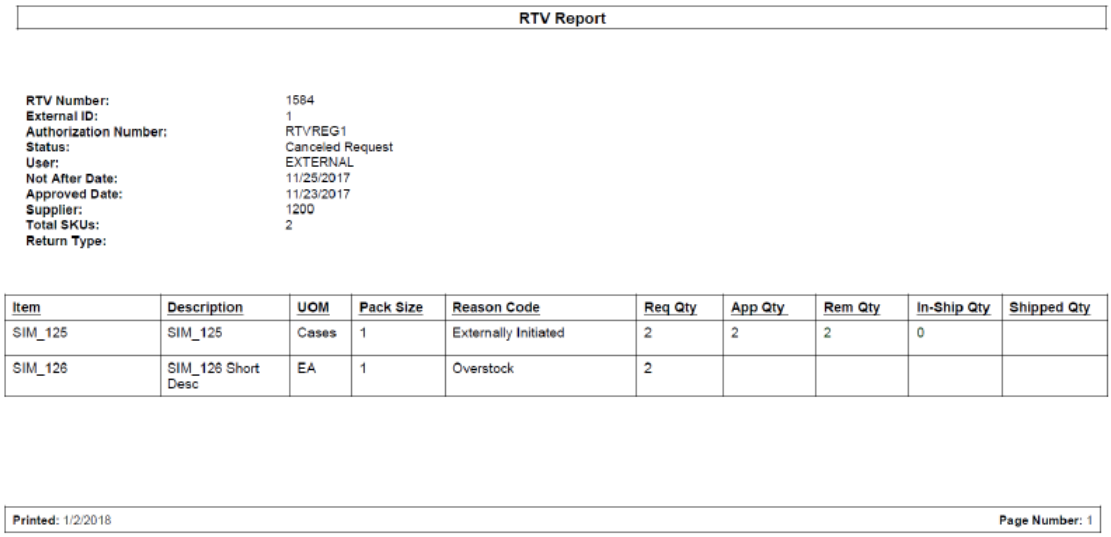
Zone	Location	Transaction Type	Transaction ID	Observed
61	Store 1311	RFID	761	Yes

Date: 08/14/2019
EPC: 980403

Zone	Location	Transaction Type	Transaction ID	Observed
404	2	RFID	762	Yes

Printed: 11/19/2019 Page Number: 1

Figure A-17 RTV Report



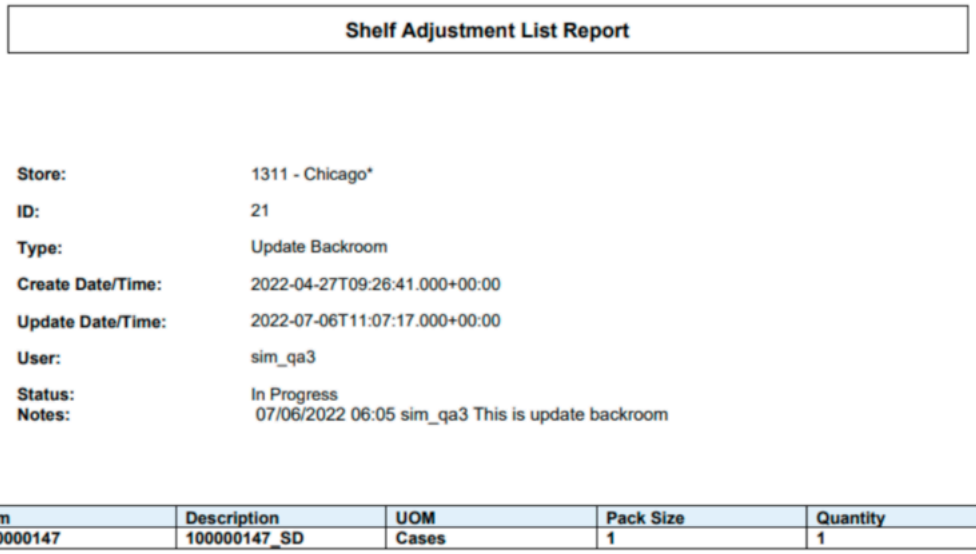
RTV Report

RTV Number: 1584
External ID: 1
Authorization Number: RTVREG1
Status: Canceled Request
User: EXTERNAL
Not After Date: 11/25/2017
Approved Date: 11/23/2017
Supplier: 1200
Total SKUs: 2
Return Type:

Item	Description	UOM	Pack Size	Reason Code	Req Qty	App Qty	Rem Qty	In-Ship Qty	Shipped Qty
SIM_125	SIM_125	Cases	1	Externally Initiated	2	2	2	0	
SIM_126	SIM_126 Short Desc	EA	1	Overstock	2				

Printed: 1/2/2018 Page Number: 1

Figure A-18 Shelf Adjustment Report

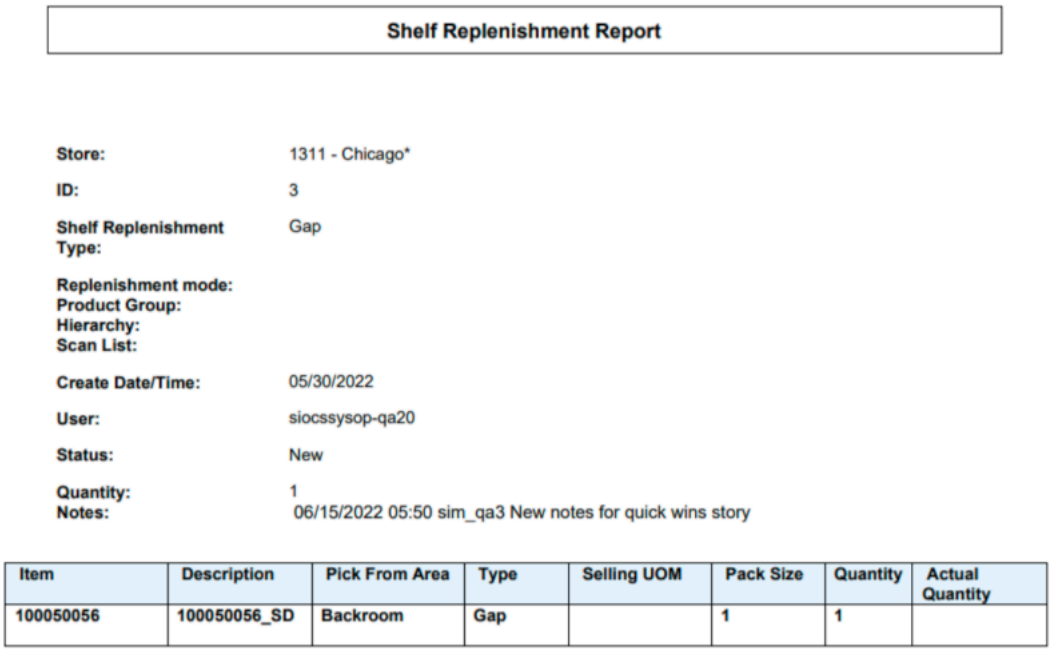


Shelf Adjustment List Report

Store: 1311 - Chicago*
ID: 21
Type: Update Backroom
Create Date/Time: 2022-04-27T09:26:41.000+00:00
Update Date/Time: 2022-07-06T11:07:17.000+00:00
User: sim_qa3
Status: In Progress
Notes: 07/06/2022 06:05 sim_qa3 This is update backroom

Item	Description	UOM	Pack Size	Quantity
100000147	100000147_SD	Cases	1	1

Figure A-19 Shelf Replenishment Report

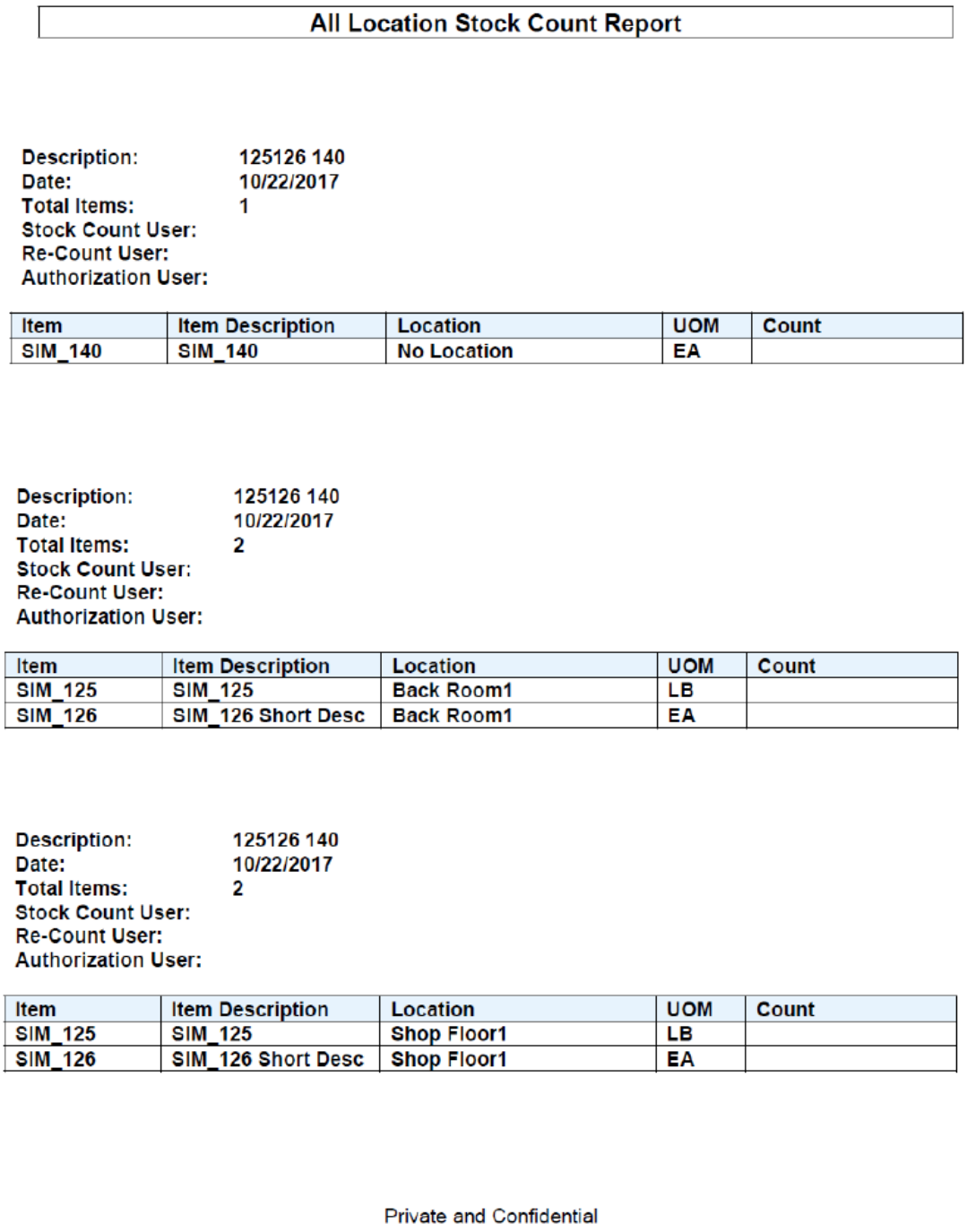


Shelf Replenishment Report

Store: 1311 - Chicago*
ID: 3
Shelf Replenishment Type: Gap
Replenishment mode:
Product Group:
Hierarchy:
Scan List:
Create Date/Time: 05/30/2022
User: siocssysop-qa20
Status: New
Quantity: 1
Notes: 06/15/2022 05:50 sim_qa3 New notes for quick wins story

Item	Description	Pick From Area	Type	Selling UOM	Pack Size	Quantity	Actual Quantity
100050056	100050056_SD	Backroom	Gap		1	1	

Figure A-20 Stock Count All Location Report



All Location Stock Count Report

Description: 125126 140
Date: 10/22/2017
Total Items: 1
Stock Count User:
Re-Count User:
Authorization User:

Item	Item Description	Location	UOM	Count
SIM_140	SIM_140	No Location	EA	

Description: 125126 140
Date: 10/22/2017
Total Items: 2
Stock Count User:
Re-Count User:
Authorization User:

Item	Item Description	Location	UOM	Count
SIM_125	SIM_125	Back Room1	LB	
SIM_126	SIM_126 Short Desc	Back Room1	EA	

Description: 125126 140
Date: 10/22/2017
Total Items: 2
Stock Count User:
Re-Count User:
Authorization User:

Item	Item Description	Location	UOM	Count
SIM_125	SIM_125	Shop Floor1	LB	
SIM_126	SIM_126 Short Desc	Shop Floor1	EA	

Private and Confidential

Figure A-21 Stock Count Export Report [XML Format]

```
<STOCK_COUNT_EXPORT>

<STOCK_COUNT>
        <COUNT_ID>662</COUNT_ID>
        <STORE_ID>1511</STORE_ID>
        <DESCRIPTION>125126 140</DESCRIPTION>
        <LIST_STOCK_COUNT_LINE_ITEM>

        <STOCK_COUNT_LINE_ITEM>
                <ITEM_ID>SIM_126</ITEM_ID>
                <ITEM_DESC>SIM_126 Short Desc</ITEM_DESC>
                <ITEM_SNAPSHOT></ITEM_SNAPSHOT>
                <LIST_UINS>
                        <UINS>

                <UIN/>
                        </UINS>
                </LIST_UINS>
        </STOCK_COUNT_LINE_ITEM>
        <STOCK_COUNT_LINE_ITEM>
                <ITEM_ID>SIM_125</ITEM_ID>
                <ITEM_DESC>SIM_125</ITEM_DESC>
                <ITEM_SNAPSHOT></ITEM_SNAPSHOT>
                <LIST_UINS>
                        <UINS>

                <UIN/>
                        </UINS>
                </LIST_UINS>
        </STOCK_COUNT_LINE_ITEM>
        <STOCK_COUNT_LINE_ITEM>
                <ITEM_ID>SIM_140</ITEM_ID>
                <ITEM_DESC>SIM_140</ITEM_DESC>
                <ITEM_SNAPSHOT></ITEM_SNAPSHOT>
                <LIST_UINS>
                        <UINS>

                <UIN/>
                        </UINS>
                </LIST_UINS>
        </STOCK_COUNT_LINE_ITEM>

        </LIST_STOCK_COUNT_LINE_ITEM>
    </STOCK_COUNT>
</STOCK_COUNT_EXPORT>
```

Figure A-22 Stock Count Rejected Item Report

Rejected Items Report

Stock Count Description: Nithin Stk Cnt
Stock Count Group: 41
Schedule Date: 1/8/14
Total Rejected Items: 1

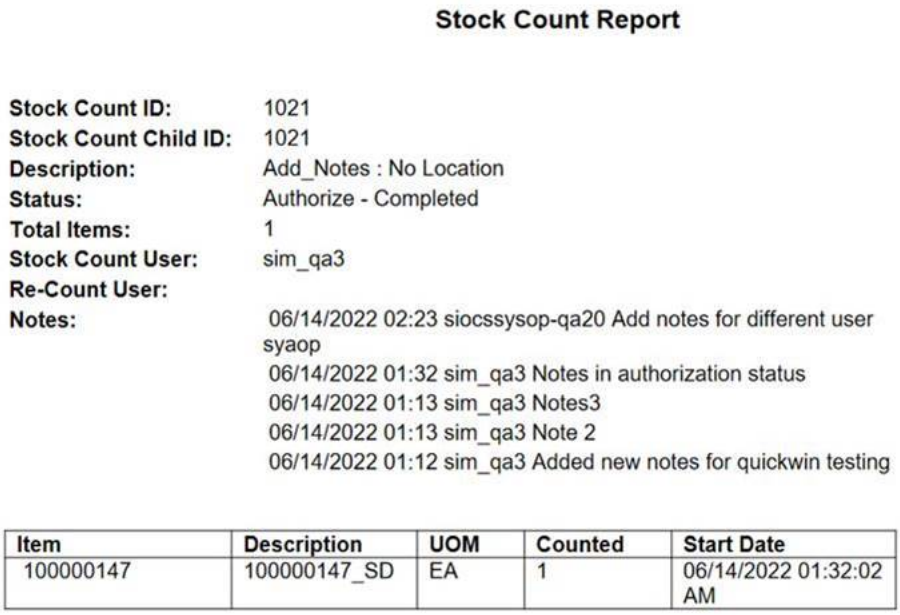
SIM Item Id	Item Description	Rejected Item ID	Rejected UIN	Count Quantity	Count Location	Status	Comments
		100177107		1		Item Rejected	

Stock Count Description: TEST Schedule
Stock Count Group: 141
Schedule Date: 11/1/13
Total Rejected Items: 5

SIM Item Id	Item Description	Rejected Item ID	Rejected UIN	Count Quantity	Count Location	Status	Comments
		100000657		2		Item Not On Count	
		100006021		2		Item Not On Count	
		1000006021		1		Item Rejected	

Private and Confidential

Figure A-23 Stock Count Report

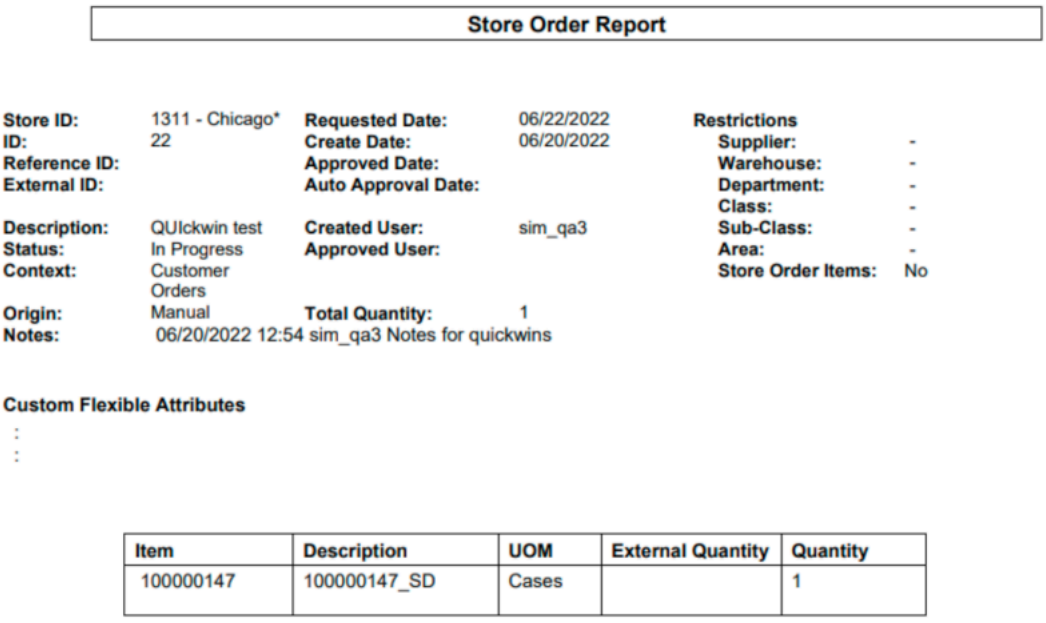


Stock Count Report

Stock Count ID: 1021
Stock Count Child ID: 1021
Description: Add_Notes : No Location
Status: Authorize - Completed
Total Items: 1
Stock Count User: sim_qa3
Re-Count User:
Notes: 06/14/2022 02:23 siocssysop-qa20 Add notes for different user syaop
06/14/2022 01:32 sim_qa3 Notes in authorization status
06/14/2022 01:13 sim_qa3 Notes3
06/14/2022 01:13 sim_qa3 Note 2
06/14/2022 01:12 sim_qa3 Added new notes for quickwin testing

Item	Description	UOM	Counted	Start Date
100000147	100000147_SD	EA	1	06/14/2022 01:32:02 AM

Figure A-24 Store Order Report



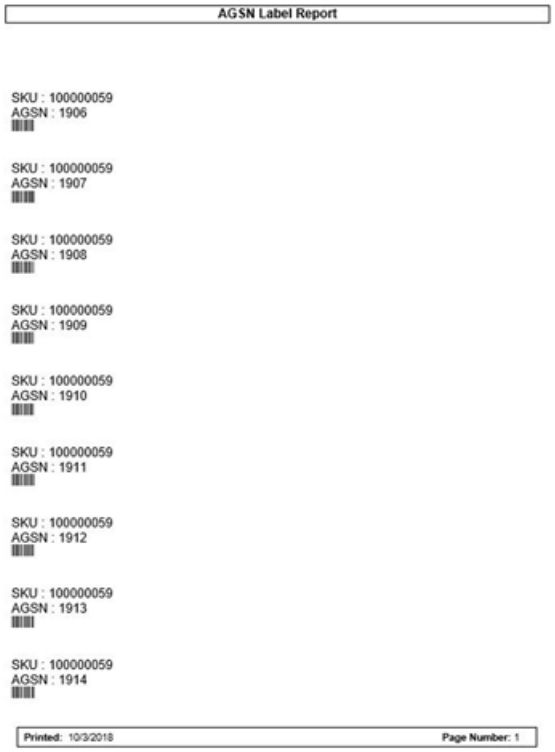
Store Order Report

Store ID: 1311 - Chicago*
ID: 22
Reference ID:
External ID:
Description: QUIckwin test
Status: In Progress
Context: Customer Orders
Origin: Manual
Notes: 06/20/2022 12:54 sim_qa3 Notes for quickwins
Requested Date: 06/22/2022
Create Date: 06/20/2022
Approved Date:
Auto Approval Date:
Created User: sim_qa3
Approved User:
Total Quantity: 1
Restrictions
Supplier: -
Warehouse: -
Department: -
Class: -
Sub-Class: -
Area: -
Store Order Items: No

Custom Flexible Attributes

Item	Description	UOM	External Quantity	Quantity
100000147	100000147_SD	Cases		1

Figure A-25 Transfer Delivery AGSN Report


AGSN Label Report

SKU : 100000059
AGSN : 1906

SKU : 100000059
AGSN : 1907

SKU : 100000059
AGSN : 1908

SKU : 100000059
AGSN : 1909

SKU : 100000059
AGSN : 1910

SKU : 100000059
AGSN : 1911

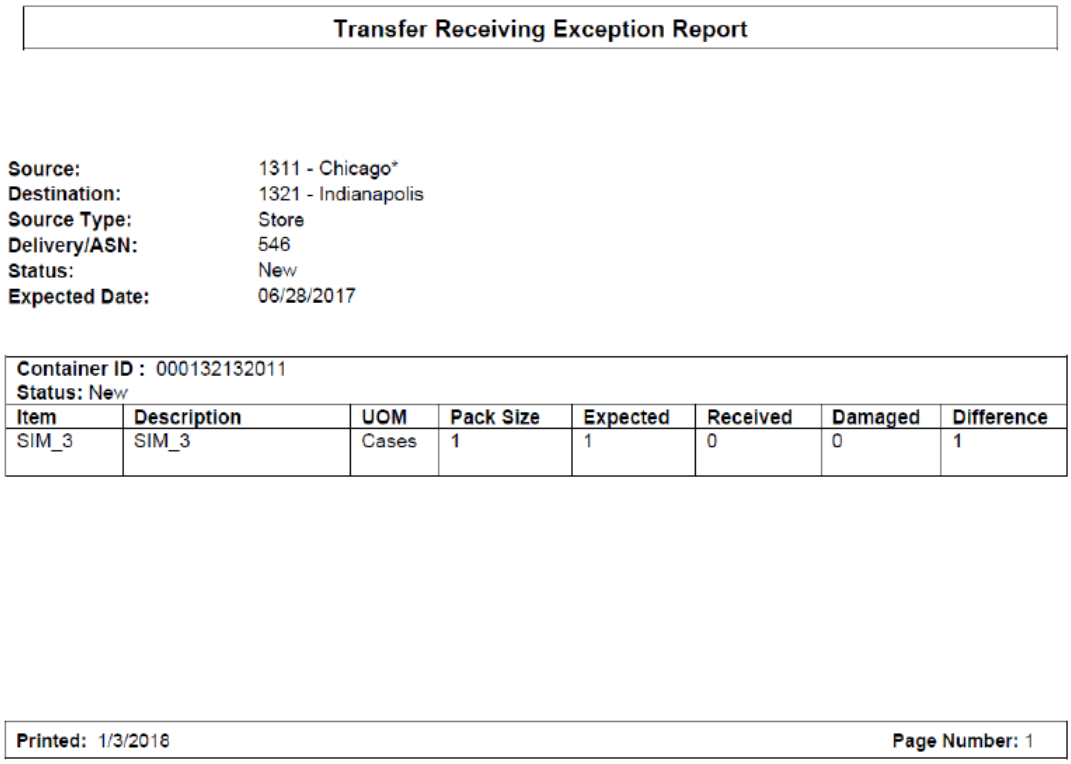
SKU : 100000059
AGSN : 1912

SKU : 100000059
AGSN : 1913

SKU : 100000059
AGSN : 1914

Printed: 10/3/2018 Page Number: 1

Figure A-26 Transfer Delivery Exception Report



Transfer Receiving Exception Report

Source: 1311 - Chicago*
Destination: 1321 - Indianapolis
Source Type: Store
Delivery/ASN: 546
Status: New
Expected Date: 06/28/2017

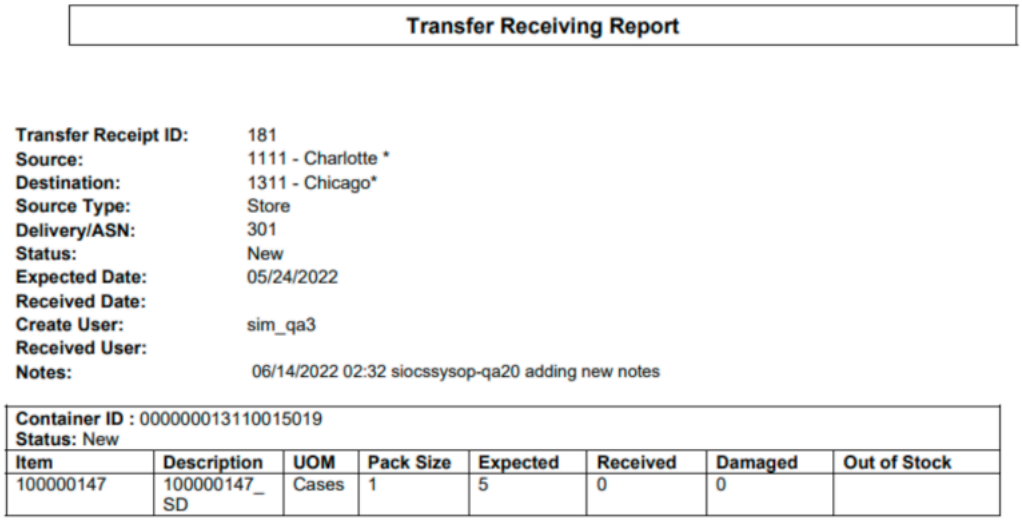
Container ID : 000132132011
Status: New

Item	Description	UOM	Pack Size	Expected	Received	Damaged	Difference
SIM_3	SIM_3	Cases	1	1	0	0	1

Printed: 1/3/2018 **Page Number:** 1

Figure A-27 Transfer Delivery Label

Figure A-28 Transfer Delivery Report



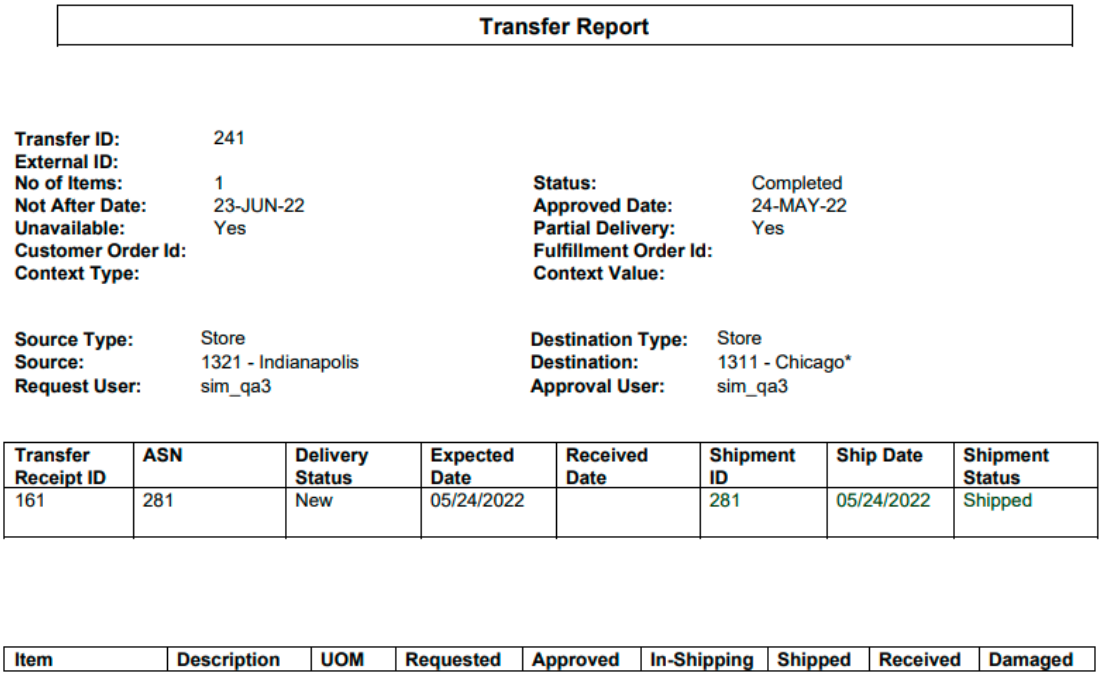
Transfer Receiving Report

Transfer Receipt ID: 181
Source: 1111 - Charlotte *
Destination: 1311 - Chicago*
Source Type: Store
Delivery/ASN: 301
Status: New
Expected Date: 05/24/2022
Received Date:
Create User: sim_qa3
Received User:
Notes: 06/14/2022 02:32 siocssysop-qa20 adding new notes

Container ID : 000000013110015019
Status: New

Item	Description	UOM	Pack Size	Expected	Received	Damaged	Out of Stock
100000147	100000147_ SD	Cases	1	5	0	0	

Figure A-29 Transfer Report



Transfer Report

Transfer ID: 241
External ID:
No of Items: 1
Not After Date: 23-JUN-22
Unavailable: Yes
Customer Order Id:
Context Type:

Status: Completed
Approved Date: 24-MAY-22
Partial Delivery: Yes
Fulfillment Order Id:
Context Value:

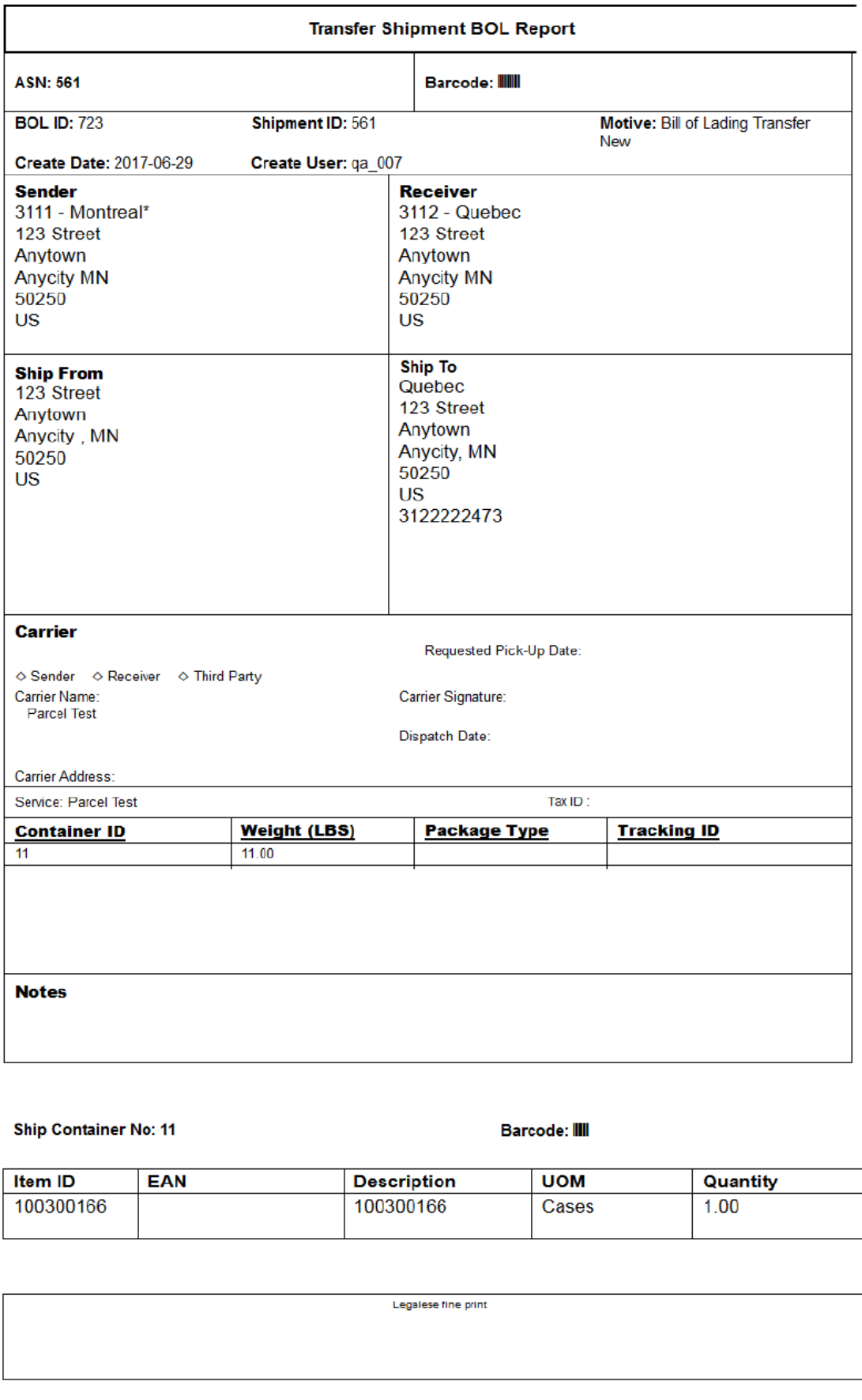
Source Type: Store
Source: 1321 - Indianapolis
Request User: sim_qa3

Destination Type: Store
Destination: 1311 - Chicago*
Approval User: sim_qa3

Transfer Receipt ID	ASN	Delivery Status	Expected Date	Received Date	Shipment ID	Ship Date	Shipment Status
161	281	New	05/24/2022		281	05/24/2022	Shipped

Item	Description	UOM	Requested	Approved	In-Shipping	Shipped	Received	Damaged

Figure A-30 Transfer Shipment BOL Report



Transfer Shipment BOL Report

ASN: 561 **Barcode:**

BOL ID: 723 **Shipment ID:** 561 **Motive:** Bill of Lading Transfer New

Create Date: 2017-06-29 **Create User:** qa_007

Sender
3111 - Montreal*
123 Street
Anytown
Anycity MN
50250
US

Receiver
3112 - Quebec
123 Street
Anytown
Anycity MN
50250
US

Ship From
123 Street
Anytown
Anycity , MN
50250
US

Ship To
Quebec
123 Street
Anytown
Anycity, MN
50250
US
3122222473

Carrier

Requested Pick-Up Date:

◇ Sender ◇ Receiver ◇ Third Party

Carrier Name:
Parcel Test

Carrier Signature:

Dispatch Date:

Carrier Address:

Service: Parcel Test Tax ID :

Container ID	Weight (LBS)	Package Type	Tracking ID
11	11.00		

Notes

Ship Container No: 11 **Barcode:**

Item ID	EAN	Description	UOM	Quantity
100300166		100300166	Cases	1.00

Legalese fine print

Driver signature	Date	Receiver signature	Date

Figure A-31 Transfer Shipment Carton Report

Transfer Shipment Container Report

Source:	1311 - Chicago*
Destination:	1321 - Indianapolis
Destination Type:	Store
Ship Date:	
Shipment ID:	360
Authorization Number:	
Status:	In Progress
Container:	379
Container Status:	In Progress
Create User:	siocssysop-qa20
Confirm User:	

Document: 317

Item	Description	UOM	Pack Size	Ship Quantity	Reason Code
100050056	100050056_SD	Cases	1	2	

Figure A-32 Transfer Shipment Report

Transfer Shipment Report

Source:	1311 - Chicago*
Destination:	1141 - Nashville
Destination Type:	Store
Ship Date:	03/28/2022
Shipment ID:	1
Authorization Number:	
Status:	Shipped
Create User:	siocssysop-qa20
Dispatch User:	siocssysop-qa20
Notes:	

Container ID : 00000011410000016
Container Status: Shipped

Item	Description	UOM	Document	Pack Size	Ship Quantity	Reason Code
100000147	100000147_SD	Cases	1	1	3	

Figure A-33 Transfer Shipping Label

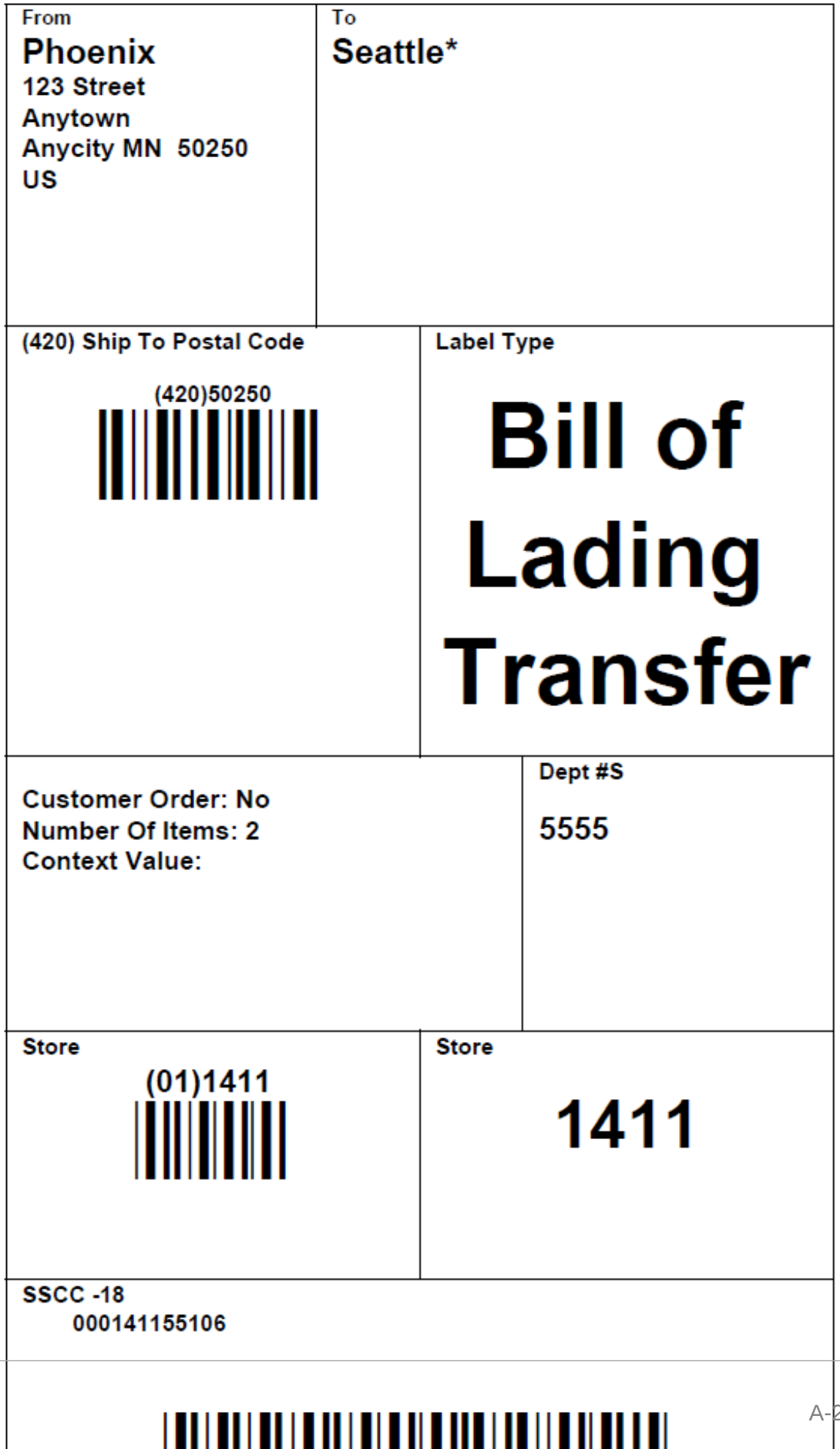
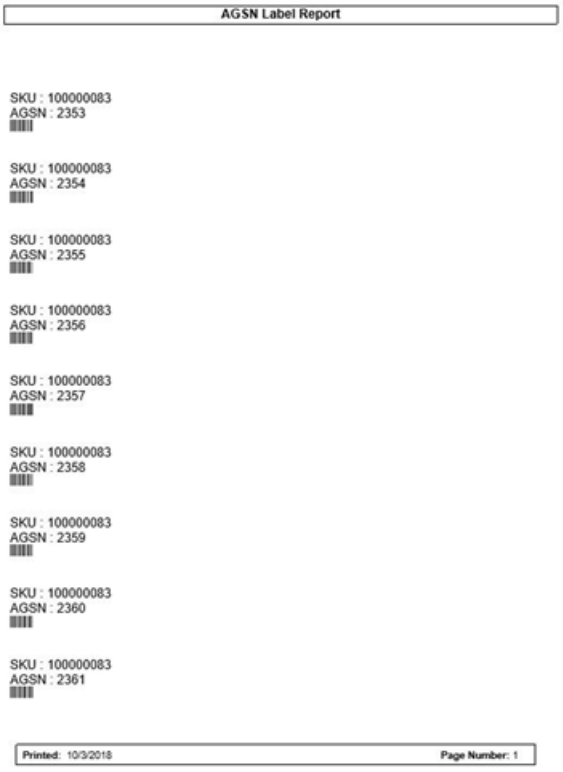


Figure A-34 Vendor Delivery AGSN Report


AGSN Label Report

SKU : 100000083
AGSN : 2353

SKU : 100000083
AGSN : 2354

SKU : 100000083
AGSN : 2355

SKU : 100000083
AGSN : 2356

SKU : 100000083
AGSN : 2357

SKU : 100000083
AGSN : 2358

SKU : 100000083
AGSN : 2359

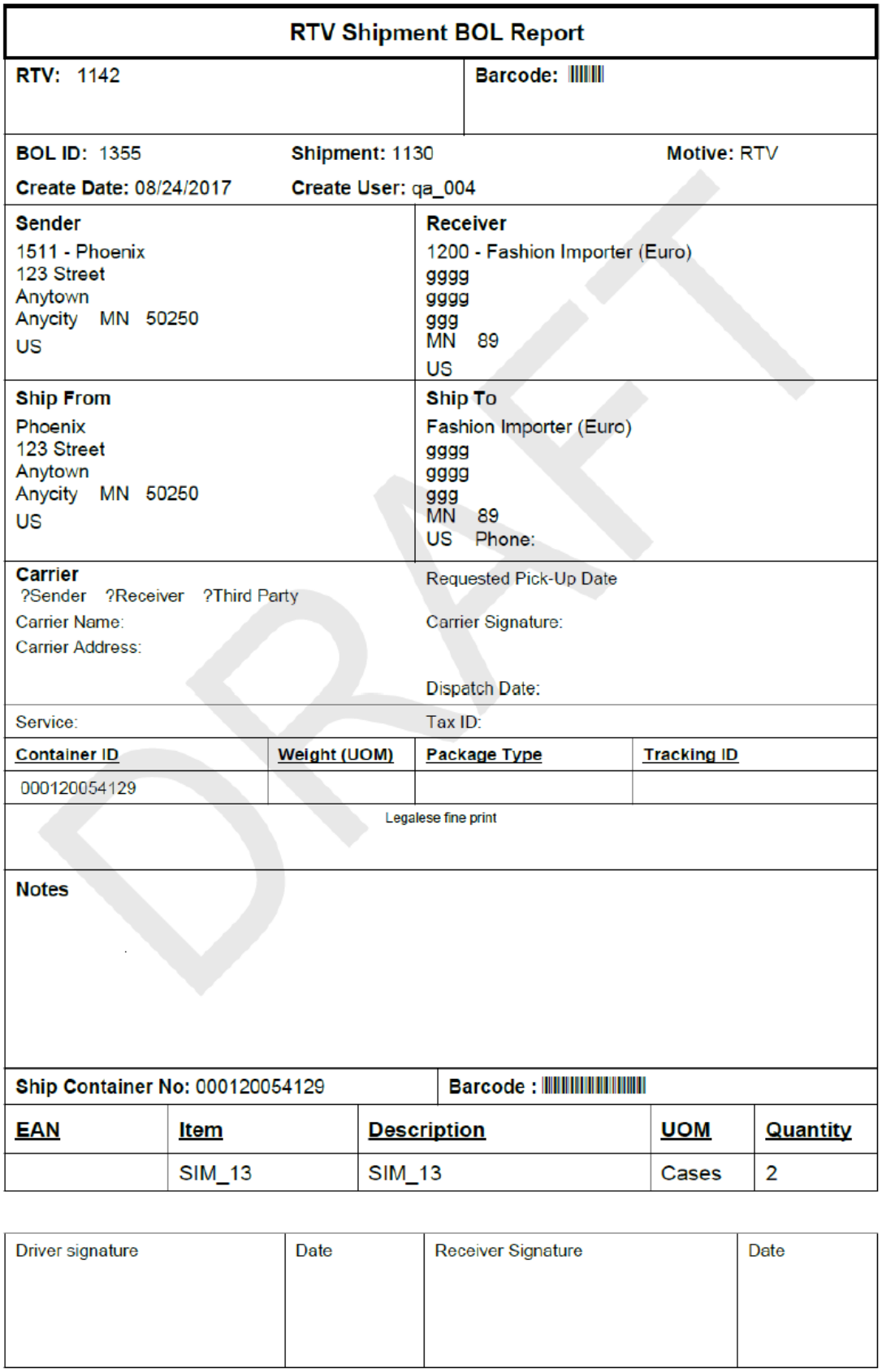
SKU : 100000083
AGSN : 2360

SKU : 100000083
AGSN : 2361

Printed: 10/3/2018 Page Number: 1

Figure A-35 Vendor Delivery Label

Figure A-36 Vendor Shipment BOL Report



RTV Shipment BOL Report

RTV: 1142 **Barcode:**

BOL ID: 1355 **Shipment:** 1130 **Motive:** RTV

Create Date: 08/24/2017 **Create User:** qa_004

Sender
1511 - Phoenix
123 Street
Anytown
Anycity MN 50250
US

Receiver
1200 - Fashion Importer (Euro)
gggg
gggg
ggg
MN 89
US

Ship From
Phoenix
123 Street
Anytown
Anycity MN 50250
US

Ship To
Fashion Importer (Euro)
gggg
gggg
ggg
MN 89
US Phone:

Carrier
?Sender ?Receiver ?Third Party
Carrier Name:
Carrier Address:

Requested Pick-Up Date
Carrier Signature:
Dispatch Date:

Service:
Tax ID:

Container ID	Weight (UOM)	Package Type	Tracking ID
000120054129			

Legalese fine print

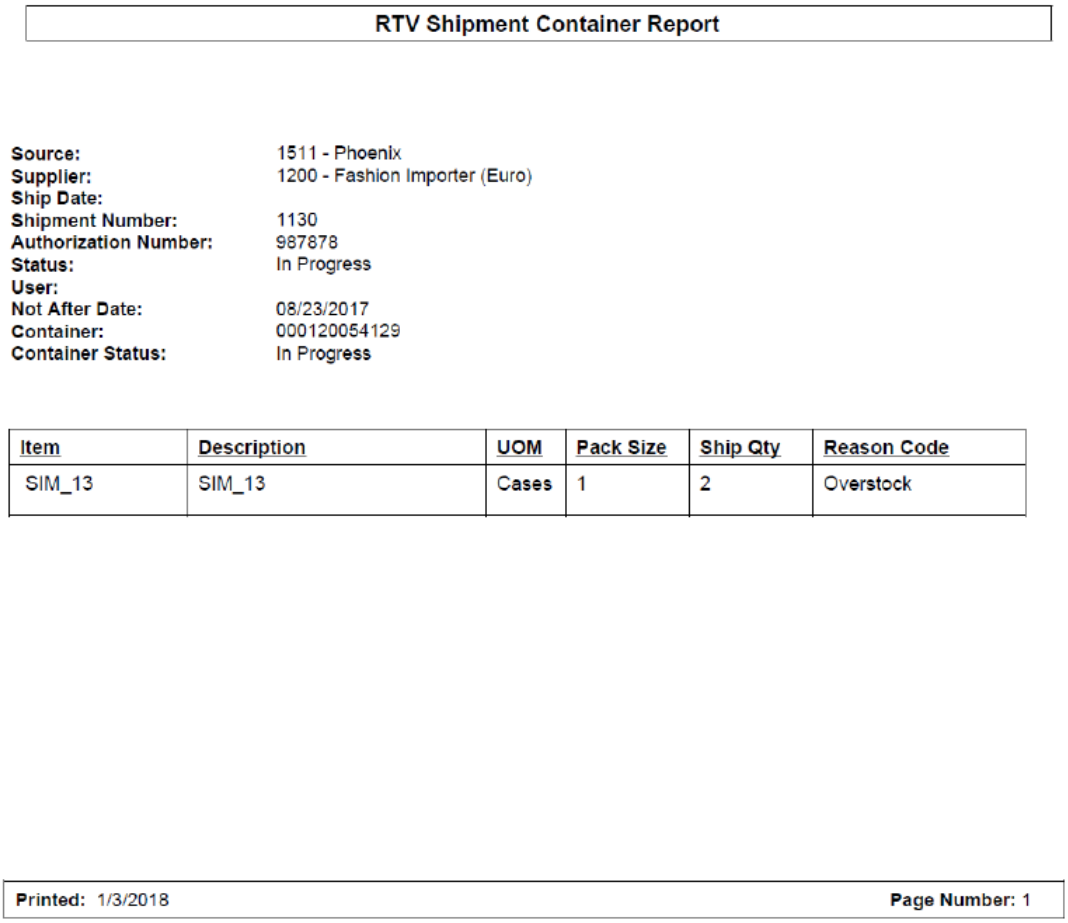
Notes

Ship Container No: 000120054129 **Barcode :**

EAN	Item	Description	UOM	Quantity
	SIM_13	SIM_13	Cases	2

Driver signature	Date	Receiver Signature	Date

Figure A-37 Vendor Shipment Carton Report



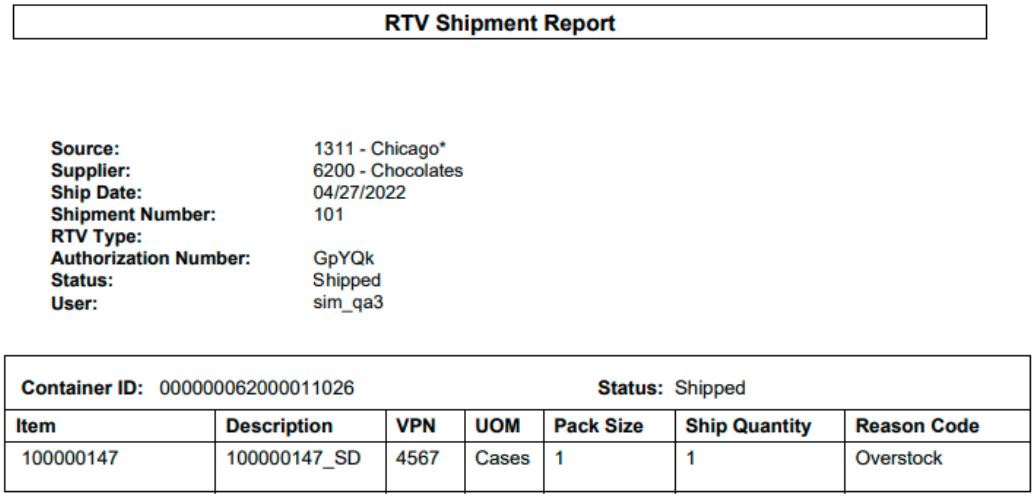
RTV Shipment Container Report

Source: 1511 - Phoenix
Supplier: 1200 - Fashion Importer (Euro)
Ship Date:
Shipment Number: 1130
Authorization Number: 987878
Status: In Progress
User:
Not After Date: 08/23/2017
Container: 000120054129
Container Status: In Progress

Item	Description	UOM	Pack Size	Ship Qty	Reason Code
SIM_13	SIM_13	Cases	1	2	Overstock

Printed: 1/3/2018 **Page Number:** 1

Figure A-38 Vendor Shipment Report



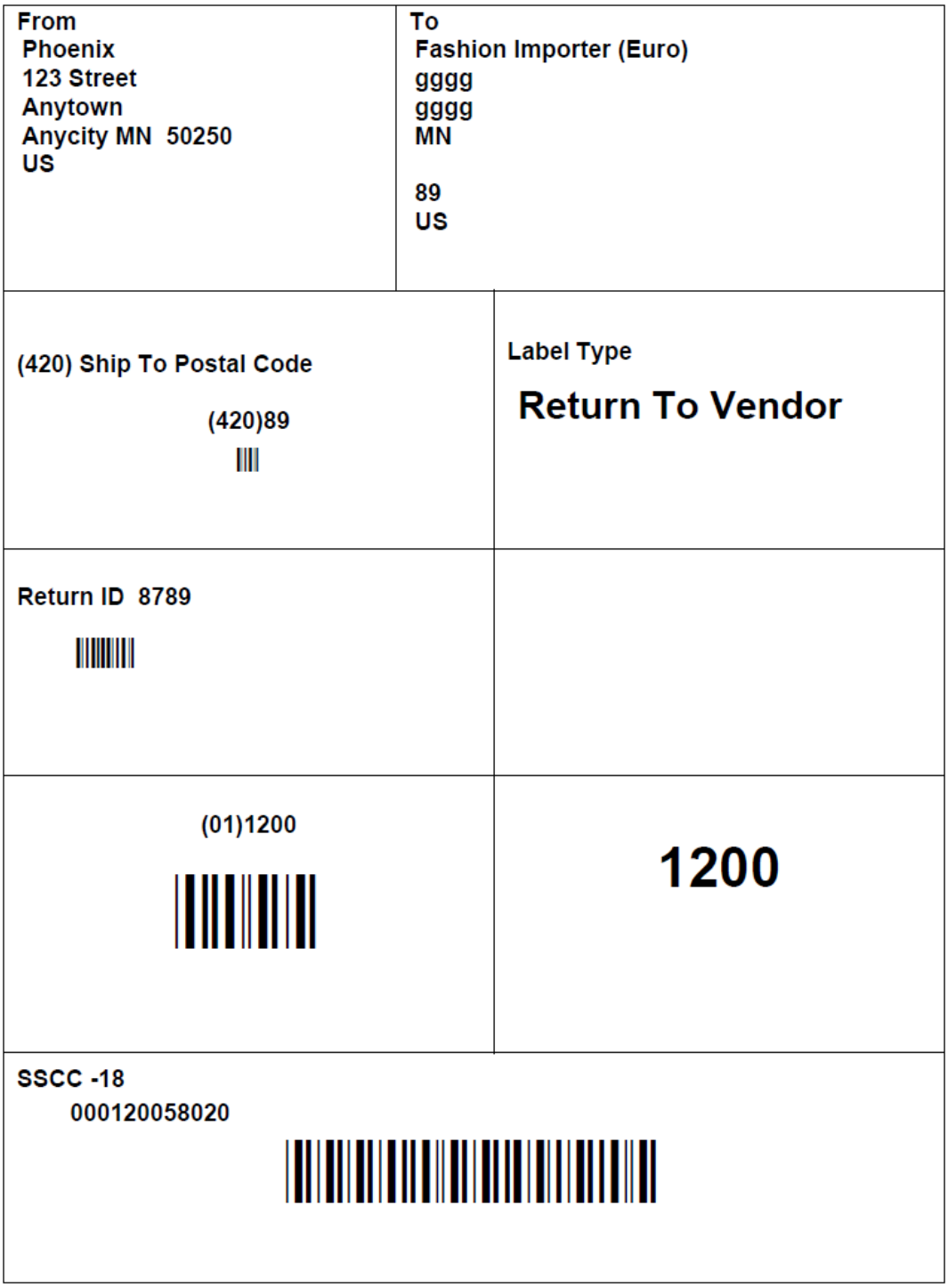
RTV Shipment Report

Source: 1311 - Chicago*
Supplier: 6200 - Chocolates
Ship Date: 04/27/2022
Shipment Number: 101
RTV Type:
Authorization Number: GpYQk
Status: Shipped
User: sim_qa3

Container ID: 000000062000011026 **Status:** Shipped

Item	Description	VPN	UOM	Pack Size	Ship Quantity	Reason Code
100000147	100000147_SD	4567	Cases	1	1	Overstock

Figure A-39 Vendor Shipment Label



B

Appendix: Batch File Layout Specifications

This chapter describes the batch file layout specifications for the following batches:

- Clearance Import File Specification
- Inventory Extract
- POS Transaction Import File Specification
- Price Change Import File Specification
- Retail Sale Audit Import File Specification
- Stock Count Results Export File Specification
- Store Sequence Data Import File Specification
- Third Party RFID File Specification
- Third Party Price File Layout
- Third Party Initial Inventory File Layout
- Third Party Stock Count Import File Layout
- Warehouse Available Inventory Import Specification

Clearance Import File Specification

Filename Format

Clearance_Tx_{YYYYMMddHHMMss}.csv

File prefix: Clearance_Tx

> **Note:**
>
> If the file comes in as a zip file, the file prefix must match the specified file prefix, and the file inside the zip must have file extension .csv.

File Layout

Comma Delimited File.

Table B-1 Clearance Import File Layout

Name	Type	Required(x indicating required)	Description
REC_ID	NUMBER(10)	x	The external record id (payload id.
RECORD_TYPE	VARCHAR2(50)	x	Record type, valid values: Create/Update/Delete.
CLEARANCE_ID	NUMBER(15)	x	Clearance id.
ITEM	VARCHAR2(25)		Item id.
LOCATION	NUMBER(10)		Location id.
LOCATION_TYPE	VARCHAR2(30)		The location type. S (STORE), W(WAREHOUSE) (Notes: only location type of 'S' is relevant to SIOCS)
EFFECTIVE_DATE	TIMESTAMP		Effective date.
RETAIL	NUMBER(20,4)		The clearance price.
UOM	VARCHAR2(25)		Unit Of Measure.
CURRENCY	VARCHAR2(25)		Price currency.
RE-SET_INDICATOR	NUMBER(1)	x	Indicates if the clearance event is a reset. valid values: 0 - the record is not a reset; 1 - the record is a reset.

Sample File

1,Create,1041231,100637113,5000,S,2022-06-30 12:06:00.0000000000,12.72,EA,USD,0

2,Create,1041231,100637121,5001,S,2022-06-30 12:06:00.0000000000,12.72,EA,USD,0

Inventory Extract

Filename Format

PRODUCT_LOCATION_INV_<store Id>_<extract date in yyyyMMddHHmmss>.DAT

File Layout

The input file is in Pipe ('|') delimited format.

Table B-2 Inventory Extract File

Record Name	Field Name	Field Type	Description
File Header	file type record descriptor	Char(5)	hardcode FHEAD
File Header	file line identifier	Number(10)	ID of current line being processed, hardcode **1**
File Header	file type	Char(4)	hardcode PLINV

Table B-2 (Cont.) Inventory Extract File

Record Name	Field Name	Field Type	Description
File Header	file create date	Date(14)YYYYMMDDHHMISS	date written by job program
File Header	loc_type	Char(1)	hardcode **S**
File Header	location	Number(10)	Location id
Transaction record	file type record descriptor	Char(5)	hardcode **FDETL**
Transaction record	file line identifier	Number(10)	ID of current line being processed, internally incremented
Transaction record	item type	Char(3)	hardcode **ITM**
Transaction record	item value	Char(25)	item ID
Transaction record	Stock on hand	Number(12,4)	total units or total weight
Transaction record	Available stock on hand	Number(12,4)	Available units or weight
Transaction record	SUOM	Number(12,4)	Stock unit of measure
Transaction record	Last Update Date	Date(14) YYYYMMDDHHMISS	
File trailer	file type record descriptor	Char(5)	hardcode **FTAIL**
File trailer	Number of data records	Number(12)	

Sample File

FHEAD|000000001|20220607090000|S|5030

FDETL|000000002|ITM|35|40|10|20220607084100

FDETL|000000003|ITM|124|34|15|20220605103215

FTAIL|000000004|3

POS Transaction Import File Specification

Filename Format

<file prefix>_<date in YYYYMMDDHH24MISS format>_<loc id>.dat

Where file prefix value is "SIMTLOG" and loc id is the store identifier. This allows file to be unique for every upload.

Example:

SIMTLOG_20180129133250_1111.dat

Zip File Format

<file prefix>_<date in YYYYMMDDHH24MISS format>.zip

Where file prefix value is "SIMTLOG". The zip file can contain one or more files from same or different stores.

Example:

SIMTLOG_20180129133250.zip

File Layout

The input file is in Pipe ('|') delimited format.

Table B-3 T-LOG File

Record Name	Field Name	Field Type	Default Value	Description
FILE HEADER	FILE HEADER File Type Record Descriptor	VARCHAR2(5)	FHEAD	Identifies the File Record Type
FILE HEADER	Location Number	NUMBER(10)		Store Number
FILE HEADER	Business Date	VARCHAR2(14)		Business Date of transactions in YYYYMMDDHHSS format
FILE HEADER	File Creation Date	VARCHAR2(14)	SYSDATE	File Create Date in YYYMMDDHHMSS format
TRANSACTION HEADER	File Type Record Descriptor	VARCHAR2 (5)	THEAD	Identifies the File Record Type
TRANSACTION HEADER	Transaction Number	VARCHAR2(128)		The unique transaction reference number generated by ORXPOS/ OMS.
TRANSACTION HEADER	Transaction Date and Time	VARCHAR2(14)		Date transactions were processed in ORXPOS/OMS
TRANSACTION HEADER	Customer Order ID	VARCHAR2(128)		External customer order ID, if transaction is a customer order
TRANSACTION HEADER	Customer Order Comments	VARCHAR(512)		Comments on the customer order
TRANSACTION DETAIL	File Type Record Descriptor	VARCHAR2(5)	TDETL	Identifies the File Record Type
TRANSACTION DETAIL	Item ID	VARCHAR2(25)		ID number of the item.
TRANSACTION DETAIL	UIN	VARCHAR2(128)		This is the UNIQUE_ID value from RTLOG
TRANSACTION DETAIL	Item Quantity	NUMBER(12,4)		Quantity of the item on this transaction
TRANSACTION DETAIL	Selling UOM	VARCHAR2(4)		UOM at which this item was sold

Table B-3 (Cont.) T-LOG File

Record Name	Field Name	Field Type	Default Value	Description
TRANSACTION DETAIL	Reason Code	NUMBER(4)		Reason entered by cashier for some transaction types. Required for voids, returns, for example.
TRANSACTION DETAIL	Comments	VARCHAR(512)		Comments for this line item
TRANSACTION DETAIL	Transaction Code	VARCHAR2(25)		The type of sale represented by this line item. Valid value are SALE,RETURN,VOID_SALE,VOID_RETURN,ORDER_NEW,ORDER_FULFILL,ORDER_CANCEL,ORDER_CANCEL_FULFILL
TRANSACTION DETAIL	Reservation Type	VARCHAR(25)		Reservation type if POS transaction is a customer order. Valid values are SPECIAL_ORDER, WEB_ORDER, PICKUP_AND DELIVERY,LAYAWAY
TRANSACTION DETAIL	Fulfillment Order Number	VARCHAR2(48)		Fulfillment Order Number from OMS
TRANSACTION DETAIL	Drop Ship Indicator	VARCHAR(1)		'P' if it is drop ship otherwise 'N'
TRANSACTION TAIL	File Record Type Descriptor	VARCHAR2(5)	TTAIL	Identifies the File Record Type
TRANSACTION TAIL	Transaction Record Counter	NUMBER(6)		Number of TDETL records in this transaction set.
FILE TAIL	File Record Type Descriptor	VARCHAR2(5)	FTAIL	Identifies the File Record Type
FILE TAIL	File Record Counter	NUMBER(10)		Number of records/transactions processed in current file (only records between head and tail)

Price Change Import File Specification

Filename Format

PriceChange _Tx_<YYYYMMddHHMMss>.csvFile prefix: PriceChange _Tx

> **Note:**
>
> If the file comes in as a zip file, the file prefix must match the specified file prefix, and the file inside the zip must have file extension .csv.

File Layout

Comma Delimited File.

Table B-4 Price Change Import File Layout

Name	type	Required(x indicating required)	Description
REC_ID	NUMBER(10)	x	The external record id (payload id.
RECORD_TYPE	VARCHAR2(50)	x	Record type, valid values: Create/Update/Delete.
PRICE_CHANGE_ID	NUMBER(15)	x	The price change ID.
ITEM	VARCHAR2(25)		Item id.
LOCATION	NUMBER(10)		Location id.
LOCATION_TYPE	VARCHAR2(30)		The location type. S (STORE), W(WAREHOUSE) (Notes: SIOCS only takes the location type of 'S', Warehouse type will be skipped)
EFFECTIVE_DATE	TIMESTAMP yyyy-mm-dd hh:mm:ss.fffffffff for example 2021-04-09 11:00:00.000000000		Effective date of price change.
RETAIL	NUMBER(20,4)		The retail with for the item and location based on the price change.
UOM	VARCHAR2(25)		The retail Unit Of Measure.
CURRENCY	VARCHAR2(25)		The currency for the location.
RETAIL_CHANGE_IND	NUMBER(6)		Indicates whether the retail changed with this price change.
MULTI_UNIT_IMPACT	VARCHAR2(4)	x	Indicates if the Price Change has impact to Multi Unit retail. Valid value are AU - Multi Unit information is added or updated; R - Multi Unit in-formation is removed; N - Multi unit information is not changed.
MULTI_UNITS	NUMBER(12,4)		Number of multi units.
MULTI_UNIT_RETAIL	NUMBER(20,4)		The Multi Unit Retail value.
MULTI_UNIT_SELLING_UOM	VARCHAR2(4)		The Multi Unit Retail Selling UOM.
MULTI_UNIT_RETAIL_CURRENCY	VARCHAR2(3)		The Multi Unit Retail Currency.

Sample File

30003, Create,650664,100637121,5000,S,2022-07-01 12:06:00.0000000000,14.72,EA,USD,1,N,,,,USD

30004,Create,650699,100637113,5000,S,2022-07-02 12:06:00.0000000000,28.72,EA,USD,1,N,,,,USD

Retail Sale Audit Import File Specification

Filename Format

SIMT_< YYYYMMDDHH24MISS>.zip

The zip file can contain one or more files:

SIMT_<YYYYMMDDHH24MISS>_<loc id>.dat

Where loc id is the store identifier.

Example:

SIMT_20180129133250_1111.dat

File Format

The input file uses pipe ('|') delimited format.

File Layout

Table B-5 ReSA File Layout

Record Name	Field Name	Field Type	Default Value	Description
FHEAD	FILE Type Record Descriptor	VARCHAR2(5)	FHEAD	Identifies the File Record Type
FHEAD	File Line ID	VARCHAR(10)		Sequential file line number
FHEAD	File Type Definition	VARCHAR2(4)	SIMT	Identifies the File Type
FHEAD	Location Number	NUMBER(10)		Store Number
FHEAD	Business Date	VARCHAR2(14)	N/A	Business Date of transactions in YYYYMMDDHHSS format
FHEAD	File Creation Date	VARCHAR2(14)	N/A	File Create Date in YYYMMDDHHMSS format
THEAD	Record Descriptor	VARCHAR2 (5)	TDETL	Identifies the File Record Type
THEAD	File Line ID	VARCHAR(10)		Sequential file line number
THEAD	Transaction Number	NUMBER(10)		The unique transaction reference number generated by ORXPOS/OMS
THEAD	Revision Number	NUMBER(3)		The version of the transaction being sent

Table B-5 (Cont.) ReSA File Layout

Record Name	Field Name	Field Type	Default Value	Description
THEAD	Transaction Date and Time	VARCHAR2(14)		Transaction date in YYYYMMDDHHMMSS format. Corresponds to the date that the transaction occurred.
THEAD	Transaction Type	VARCHAR2(14)		Transaction Type Code (for example, SALE, RETURN, SPLORD)
THEAD	Pos created flag	VARCHAR2(1)		'Y' identifies that the transaction occurred at ORXPOS, 'N' identifies that the transaction was created in ReSA
TDETL	Record Descriptor	VARCHAR2(5)	TDETL	Identifies the File Record Type
TDETL	File Line ID	VARCHAR(10)	0000000001	Sequential file line number.
TDETL	Item Sequence Number	NUMBER(4)		The order in which items were entered during a transaction
TDETL	Item	VARCHAR2(25)		ID number of the item.
TDETL	Item Number Type	VARCHAR2(6)		Type of Item sold. Can be 'ITEM', 'REF', 'GCN', 'NMITEM'
TDETL	Item Status	VARCHAR2(6)		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 PVLCO - Post Void Layaway Complete PVORD - Post Void Order Complete
TDETL	Serial Number	VARCHAR2(128)		This is the UNIQUE_ID value from RTLOG
TDETL	Pack Indicator	VARCHAR2(1)		Pack indicator of item sold or returned
TDETL	Catch Weight Indicator	VARCHAR2(1)		Indicates if item is a catchweight item
TDETL	Item Quantity Sign	VARCHAR2(1)		Determines if the Total Sale Quantity is positive or negative 'P' - Positive 'N' - Negative

Table B-5 (Cont.) ReSA File Layout

Record Name	Field Name	Field Type	Default Value	Description
TDETL	Item Quantity Value	NUMBER(20)		Total sales value of goods sold/ returned (4 implied decimal places), for example, Total Quantity * 10000
TDETL	Standard UOM	VARCHAR2(4)		Standard UOM of the Item
TDETL	Selling UOM	VARCHAR2(4)		UOM at which this item was sold
TDETL	Wastage Type	VARCHAR2(6)		Wastage type of item sold or returned
TDETL	Wastage Percentage	NUMBER(12)		Wastage Percent*10000 (4 implied decimal places), wastage percent of item sold or returned
TDETL	Drop Ship Indicator	VARCHAR2(1)	N	This will always be N for Export
TDETL	Actual Weight Quantity	NUMBER(12)		Actual Weight Quantity*10000 (4 implied decimal places), the actual weight of the item, only populated if catchweight_ind = 'Y'
TDETL	Actual Weight Sign	Char(1)		Sign of the actual weight
TDETL	Reason Code	VARCHAR2(6)		Reason entered by cashier for some transaction types
TDETL	Sale Value	NUMBER(20)		Total Sales Value * 10000 (4 implied decimal places), sales value, net sales value of goods sold
TDETL	Sales Sign	VARCHAR2(1)		Determines if the Total Sales Value is positive or negative 'P' - Positive 'N' - Negative
TDETL	Unit Retail	NUMBER(20,4)		Unit retail with 4 implied decimal places
TDETL	Sales Type	VARCHAR2(1)		Indicates if the line item is a Regular Sale, a CO serviced by OMS (External CO), or a CO serviced by Inventory management application (In-Store CO)
TDETL	Customer Order Number	VARCHAR2(50)		Customer Order Number
TDETL	Customer Order Type	Char(6)		Customer order type
TDETL	Fulfillment Order Number	VARCHAR2(50)		Fulfillment Order Number from OMS
TDETL	Customer Order Line Number	NUMBER (10)		Customer order line number
TTAIL	Record Type Descriptor	VARCHAR2(5)	TTAIL	Identifies the File Record Type
TTAIL	File Line ID	NUMBER(10)		Sequential file line number

Table B-5 (Cont.) ReSA File Layout

Record Name	Field Name	Field Type	Default Value	Description
TTAIL	Transaction Record Counter	NUMBER(6)		Number of TDETL records in this transaction set
FTAIL	File Record Type Descriptor	VARCHAR2(5)	FTAIL	Identifies the File Record Type
FTAIL	File Line ID	NUMBER(10)		Sequential file line number
FTAIL	File Record Counter	NUMBER(10)		Number of records/transactions processed in current file (only records between head and tail)

Sample Data File

FHEAD|0000000001|SIMT|5141|20210307111049|20210307144046

THEAD|0000000002|1141|1|1|20210307000000|SALE|N

TDETL|0000000003|1|100000147|ITEM|S||||P|3||EA|||N||||||||||

TTAIL|0000000004|1

THEAD|0000000005|270888|1|1|20210307000000|RETURN|N

TDETL|0000000006|1|100000147|ITEM|R||||N|3||EA|||N||||||||||

TTAIL|0000000007|1

FTAIL|0000000008|6

Stock Count Results Export File Specification

The stock count result export file is generated when unit amount stock count authorization completes. The stock count authorization process can be a manual authorization or invoked by third party stock count batch for an auto-authorized unit amount stock count. This export file can be uploaded to RMS by RMS file to update their inventory with the actual physical stock count.

Table B-6 Stock Count Export File

Record Name	Field Name	Field Type	Description
File Header	file type record descriptor	Char(5)	hardcode FHEAD
File Header	file line identifier	Number(10)	ID of current line being processed, hardcode **000000001**
File Header	file type	Char(4)	hardcode STKU
File Header	file create date	Date(14)YYYY MMDDHHMISS	date written by convert program
File Header	stocktake_date	Date(14)YYYY MMDDHHMISS	take_head.stocktake_date
File Header	cycle count	Number(8)	stake_head.cycle_count

Table B-6 (Cont.) Stock Count Export File

Record Name	Field Name	Field Type	Description
File Header	loc_type	Char(1)	hardcode **W** or **S**
File Header	location	Number(10)	stake_location.wh or stake_location.store
Transaction record	file type record descriptor	Char(5)	hardcode **FDETL**
Transaction record	file line identifier	Number(10)	ID of current line being processed, internally incremented
Transaction record	item type	Char(3)	hardcode **ITM**
Transaction record	item value	Char(25)	item ID
Transaction record	inventory quantity	Number(12,4)	total units or total weight
Transaction record	location description	Char(30)	Where in the location the item exists. For example, Back Stockroom or Front Window Display
File trailer	file type record descriptor	Char(5)	hardcode **FTAIL**
File trailer	file line identifier	Number(10)	ID of current line being processed, internally incremented
File trailer	file record count	Number(10)	Number of detail records

Store Sequence Data Import File Specification

Sequencing functionality provides users the ability to know the relative location of an item in a store. Sequencing a store improves store processes and reduces the time that employees spend looking for items. The retailer can sequence all items in the store and create unique locations to hold the items.

Sequencing defines how many items can be stored in a particular location, and allows the definition of a capacity for that item location combination. The capacity is used for in-store replenishment when generating the shelf replenishment pick list. Sequencing is used within Stock Counts, Customer Order Picking, Transfer Request, and Shelf Replenishment to aid the user in proceeding to the next item during the transaction for efficiency. Lastly, the Sequencing Primary Location is displayed to the user on the Item Detail screen.

Filename Format

<file prefix>_<date in YYYYMMDDHH24MISS format>_<loc id>.dat

Where file prefix value is "SSEQ" and loc id is the store identifier. This allows file to be unique for every upload.

Example:

SSEQ_20180129133250_1111.dat

Zip Filename Format

<file prefix>_<date in YYYYMMDDHH24MISS format>.zip

Where file prefix value is "SSEQ". The zip file can contain one or more files from same or different stores. The complete file needs to be added for zip file for job to pick it for processing.

Example:

SSEQ_20180129133250.zip

File Format

The input file would be in pipe ('|') delimited format.

File Layout

Table B-7 Store Sequence Import File

Record Name	Field Name	FieldType	Description
File Header	file type record descriptor	Char(5)	hardcode FHEAD
File Header	Store ID	Number(10)	Store identifier
File Header	Delete	DELETALL	Optional flag to delete previous records
Sequence record	file type record descriptor	Char(5)	hardcode SHEAD
Sequence record	Area type	Number(9)	The Store Sequence Area. 0 = None, 1 = Shopfloor, 2 = Backroom
Sequence record	Child sequenced	Varchar2(1)	‘Y’ if child is sequenced, ‘N’ if not
Sequence record	Department ID	Number(12)	Department ID
Sequence record	Class ID	Number(12)	Class ID
Sequence record	Description	Varchar2(255)	Description of Store Sequence
Sequence record	Not sequenced	Varchar2(1)	Y indicates a default sequence containing all items that have not been sequenced elsewhere
Sequence record	Sequence Order	Number(20)	The order the store sequence is in compared to other store sequences
Sequence detail	file type record descriptor	Char(5)	hardcode SDETL
Sequence detail	Item ID	Varchar2(25)	Item ID
Sequence detail	Primary location	Varchar2(1)	Indicator if the location specified is the primary location for the item, Y if is primary location for item, N otherwise

Table B-7 (Cont.) Store Sequence Import File

Record Name	Field Name	FieldType	Description
Sequence detail	Item sequence order	Number(20)	Order of item within store sequence
Sequence detail	Capacity	Number(11,2)	The size of the location appropriate to unit of measure
Sequence detail	Ticket quantity	Number(11,2)	The quantity of tickets that need to be printed or used for the item inventory location
Sequence detail	Ticket format ID	Number(10)	Item ticket format identifier
Sequence trailer	File type record descriptor	Char(5)	hardcode STAIL
File trailer	File type record descriptor	Char(5)	hardcode FTAIL

Sample Data File

FHEAD|5000

SHEAD|1|N|||ShopFloor5|N|1

SDETL|100695153|Y|1|100|1||1|0

STAIL

FTAIL

Third Party RFID File Specification

Filename Format

ext_rfid _<YYYYMMDDHHMMSS>.csv

File Layout

Comma Delimited File.

Table B-8 Third Party RFID File Specification

Field Name	Description	Required	Type
ACTION	CREATE and DELETE are the only two valid actions for RFI.	Yes	VARCHAR2(20)
EPC	Electronic product code (SGTIN-96).	Yes	VARCHAR(256)
ITEM_ID	Identifier of the item/sku.	Yes	VARCHAR2(25)
LOCATION_ID	Location identifier.	Yes	NUMBER(10)
LOCATION_TYPE	Location Type, 1 - store, 2 - warehouse.	Yes	NUMBER(2)

Table B-8 (Cont.) Third Party RFID File Specification

Field Name	Description	Required	Type
ZONE_ID	The zone within the location that the RFID is located.	No	NUMBER(15)
EVENT_DATE	The timestamp of the RFID read.	No	TIMESTAMP(6)

Sample File

RFID_{YYYYMMDDHHMMSS}_{LOC}_{LOC_TYPE}.csv

"REPLACE","11111111111111111111","100637113",5000,1,1001,

"03-07-2021 0:00"

"REPLACE","11111111111111111112","100637148",5000,2,1022,

"05-10-2021 0:00"

File Contents Explanation

- It is expected that the RFID provider to ensure the record uniqueness (A unique record is identified by store/item/effective date time), within a file, each record must be unique. The record action is denoted by action type, only one dataset action is allowed. EICS only support CREATE OR DELETE as dataset action for third party rfid, UPDATE type is not supported, use replace for updating a record.
- Split the Data into multiple files. EICS loads the data in parallel from multiple files. Loading files from multiple files in parallel provides performance advantage than loading from a single file. It is recommended to file provider to split the data into multiple files to load data efficiently in parallel loading, each file contains single store is recommended.
- Compress the data files. If data file contains large datasets, it is recommended that compress the load files individually, when loading the data file. Use EICS System Configuration Console to specify the file suffix (for example, zip).

Third Party Price File Layout

Filename Format

EXTPC_{YYYYMMDDHHMMSS}_{LOC}_{LOC_TYPE}.csv

Table B-9 Third Party Price Import File Specification

Field Name	Description	Required	Type
RECORD_ACTION	CREATE, UPDATE, DELETE .	Yes	CHAR(20)
ITEM_ID	The unique alphanumeric value for the transaction level item.	Yes	CHAR(25)
STORE_ID	The number that uniquely identifies the store.	Yes	Number(10)

Table B-9 (Cont.) Third Party Price Import File Specification

Field Name	Description	Required	Type
EFFECTIVE_DATE	The date on which the price change became effective. The Dates must be GMT as the file will parse and process the dates as GMT dates. yyyy-mm-dd hh:mm:ss.fffffffff for example, 2021-04-09 11:00:00.000000000	Yes	Timestamp
END_DATE	Promotion end date. The Dates must be GMT as the file will parse and process the dates as GMT dates. yyyy-mm-dd hh:mm:ss.fffffffff for example, 2021-04-09 11:00:00.000000000	No	Timestamp
PRICE_TYPE	The item price type. Valid values: 200- Clearance 201- Promotional 202- Regular 230- Independent clearance reset.	Yes	NUMBER(3)
PROMOTION_NAME	Promotion name.	No	CHAR(160)
SELLING_UNIT_RETAIL	Contains the current single unit retail in the selling unit of measure.	Yes	NUMBER(20,4)
SELLING_UNIT_RETAIL_CURRENCY	Contains the selling unit retail currency.	Yes	CHAR(3)
SELLING_UOM	Contains the selling unit of measure for an items single-unit retail.	Yes	CHAR(4)
MULTI_UNITS	Contains the current multi-units. If the record is being written as a result of a change in the multi-unit retail, then this field contains the new multi-units.	No	NUMBER(12,4)
MULTI_UNIT_RETAIL	Contains the current multi-unit retail in the selling unit of measure.	No	NUMBER(20,4)
MULTI_UNIT_RETAIL_CURRENCY	Contains the multi-unit retail currency.	No	CHAR(3)
MULTI_UNIT_SELLING_UOM	Contains the selling unit of measure for an items multi-unit retail.	No	CHAR(4)
CREATE_DATETIME	Contains the record creation date. yyyy-mm-dd hh:mm:ss.fffffffff for example, 2021-04-09 11:00:00.000000000	No	Timestamp
REC_ID	The id of the record.	Yes	NUMBER(15)
RETAIL_CHANGE_IND	Indicates whether the retail changed with this price change. Valid values are: 0 - retail price not changed 1 - retail price changed	No	NUMBER(6)

Table B-9 (Cont.) Third Party Price Import File Specification

Field Name	Description	Required	Type
MULTI_UNIT_IMPACT	Indicates if the Price Change has impact to Multi Unit retail. Valid values are: AU - Multi Unit information is added or updated R - Multi Unit information is removed N - Multi unit information is not changed.	Yes	CHAR(4)
PRICE_EVENT_ID	The id of the price event.	Yes	NUMBER(15)

Sample File

REPLACE,100637113,5000,2021-04-09 11:00:00,,202,,149.99,USD,EA,,,,,2021-04-07 11:00:00,1,1,N,9999

File Contents Explanation

- It is expected that the pricing provider will ensure the record uniqueness (A unique record is identified by store/item/effective date time), within a file. Each record must be unique. The record action is denoted by action type, only a dataset action is allowed for unique store/item/date.
- For example, for store 5000, item A, a price on date 2018 Dec 10 00:00:00 record in the file can be one of the following (CREATE, DELETE). The same record with more than one dataset action will be rejected. EICS only supports CREATE OR DELETE as dataset action for third party pricing.
- The same file cannot have two records with this combination store/item/effective with different price type, if clearance need to be on today, then this file should only have a single record for clearance type.
- The clearance record can have an end date if the end date is known at time of the clearance creation.
- For independent clearance reset event (to end all active clearance for a store/item which does not have end date), the pricing provider needs to send clearance reset record (with price type =203), the import process ends any active clearance for item store timeline (set the end date to the clearance reset effective date). The clearance reset record is only for ending the active item store clearance, the price in the clearance reset record is not used for updating.
- In EICS, there is no client UI which requires or uses the promotion, clearance or price change identifier. For data import integration backend processing, the record is uniquely identified by item/store/effective date time and price type external pricing change identifier has no meaning to our system. Promotion name is used in EICS as context type; therefore it is included in the integration interface.
- Split the Data into Multiple Files. EICS loads the data in parallel from multiple files. Loading files from multiple files in parallel provides performance advantage overloading from a single file. It is recommended to file provider to split the data into multiple files to load data efficiently in parallel loading. Each file contains single store is recommended.
- The Dates must be GMT as the file will parse and process the dates as GMT dates.

Third Party Initial Inventory File Layout

Filename Format

<EXTSTK _<date YYYYMMDDHH24MISS >.zip

The zip file can contain one or more files from same or different stores:

EXTSTK_<date in YYYYMMDDHH24MISS format>.dat

DataFilename format

<file prefix>_<date in YYYYMMDDHH24MISS format>_<loc id>.dat

Where file prefix value is EXTSTK_ and loc id is the store identifier. This allows file to be unique for every upload.

Example: EXTSTK_20180129133250_1111.dat

File Layout

Pipe-delimited (|) file

Table B-10 Initial Inventory Import File

Record Name	Field Name	Field Type	Default Value	Description
FHEAD	Record Descriptor	Char(5)	FHEAD	File head marker
	Store Number	Char(10)		Store number file was uploaded for. It is assumed only one store is passed in per file. (Required)
FDETL	Record Descriptor	Char(5)	FDETL	Detail record marker
	Upload Date	Date(14)		Indicates date/time item was physically counted. (YYYYMMDDHH24MISS) For example, 20180129134600 (Required for UIN Records)
	Area Number	Char(10)		10-digit code indicating where in the store the item is located. (Optional)
	UPC or Item Number	Char(25)		25-digit universal product code. (Required)
	Count Quantity	Number(12, 4)		Quantity counted for item, required. This field must allow for decimals when counting in UOM other than each. (Required)
	UIN(Item Serial Number)	Char(128)		Unique identification serial number for item, required if current item requires serial number.
FTAIL	Record Descriptor	Char(5)	FTAIL	File tail marker

Sample File

FHEAD|5000|

FDETL|20180129235959|1|100665085|1|ItemSerialNum1234|

FDETL|201180129140000|1|100665085|1|ItemSerialNum9999|

FDETL|20180129000000|1|100665085|1||

FTAIL|

Third Party Stock Count Import File Layout

Filename Format

<file prefix>_<date YYYYMMDDHH24MISS >.zip

Where file prefix value is STK.

Example:

STK_20180129133250.zip

The zip file can contain one or more files from same or different stores:

Data Filename Format

<file prefix>_<date in YYYYMMDDHH24MISS format>_<loc id>.dat

Where file prefix value is STK and loc id is the store identifier.

Example:

STK_20180129133250_1111.dat

File Layout

Pipe-delimited (|) file

Table B-11 Third Party Stock Count Import File

Record Name	Field Name	Field Type	Default Value	Description
FHEAD	Record Descriptor	Char(5)	FHEAD	File head marker
	Store Number	Char(10)		Store number file was uploaded for. It is assumed only one store is passed in per file. (Required)
	Stock Count ID	Number(12)		Unique identifier for item. Assumption is application will always take first stock count ID listed. (Required)

Table B-11 (Cont.) Third Party Stock Count Import File

Record Name	Field Name	Field Type	Default Value	Description
FDETL	Record Descriptor	Char(5)	FDETL	Detail record marker
	Stock Count Date	Date(14)		Indicates date/time item was physically counted. (YYYYMMDDHH24MISS) For example, 20180129134600 (Required) Note: If not using timestamp, use 00 for time.
	Area Number	Char(10)		10-digit code indicating where in the store the item is located. (Optional)
	UPC or Item Number	Char(25)		25-digit universal product code. (Required)
	Count Quantity	Number(12, 4)		Quantity counted for item, required. This field must allow for decimals when counting in UOM other than each. (Required)
	UIN(Item Serial Number)	Char(128)		Unique identification serial number for item, required if current item requires serial number.
FTAIL	Record Descriptor	Char(5)	FTAIL	File tail marker

Sample File

FHEAD|5000|1074|

FDETL|20180129235959|1|100665085|1|ItemSerialNum1234|

FDETL|201180129140000|1|100665085|1|ItemSerialNum9999|

FDETL|20180129000000|1|100665085|1||

FTAIL|

Warehouse Available Inventory Import Specification

Filename Format

InvAvailWh_Tx_{YYYYMMddHHMMss}.csv

File Layout

- All files should be in CSV (comma-separated values) format, with a ".csv" filename extension.

- The batch jobs also support zipped files which will be extracted upon download and processed individually. Files contained within .zip files must adhere to the same filename format.
- Empty or blank fields within a record will be considered null. Every column must be present even if it is empty or null.
- String fields containing a comma or double quote must be quoted (with double quotes), a double quote in a field must be represented by 2 double quote characters. Line breaks within quoted fields are not supported.

Table B-12 Warehouse Available Inventory Import File Layout

Field Name	Description	Required	Data Type
ACTION	The record action type. Valid values: REPLACE	Yes	VARCHAR2(20)
ITEM_ID	The unique identifier of the item - references the ITEM_ID column in the ITEM table.	Yes	VARCHAR2 (25)
WAREHOUSE_ID	Virtual warehouse id	Yes	NUMBER (10, 0)
LOC_TYPE	Type of location. W represents the virtual warehouse.'	Yes	VARCHAR2(1)
AVAIL_QTY	Available quantity of the item at the location. This qty is calculated by subtracting transfer reserved qty, customer reserved qty, non_sellable inventory and RTV from stock on hand.	Yes	NUMBER(20,4)
STOCK_ON_HAND	Current stock on hand for the item.	Yes	NUMBER(20,4)
STANDARD_UOM	The standard unit of measure of the warehouse item.	No	VARCHAR2 (4)
PHYSICAL_WH	Physical warehouse that Is assigned to the virtual warehouse.	Yes	NUMBER(10)
QUANTITY_RESERVED	Reserved quantity.	No	NUMBER(20,4)
QUANTITY_IN_TRANSIT	In transit quantity.	No	NUMBER(20,4)

Example File

File Name: InvAvailWh_Tx_{YYYYMMddHHMMss}.csv

REPLACE,100637113,9999,W,100,150,EA,8888,,,

C

Appendix: Auto-Authorized Third-Party Stock Count Process Overview

This section describe overview steps to setup and auto authorize a third party stock count:

1. In the **Operations/Product Group** dialog, create and save a new product group with the following attributes:

   Type: Select Unit or Unit and Amount

   Counting Method: Select Third Party

   A**uto Authorize**: Select this check box

   > **Note:**
   >
   > If auto authorize is selected, the processing of the stock count will attempt to do many automated steps when loading the third party stock count information. If auto authorize is not selected, after loading the file information the authorization process is manual.

2. In the **Operations/Product Group Component** dialog, update the created product group with the desired items to count and save. To count all items in all departments, set **All Department** attribute to **Yes**.
3. In the **Operations/Product Group Schedule** dialog, create a product group schedule for the previous created product group.

   > **Note:**
   >
   > If creating a schedule for a unit count that is active on the current date, you will have the option of generating the stock count immediately.

4. In the **Admin/Technical Maintenance/Job Admin** dialog, create and start a new job.

   Choose **Generate Unit Stock Count** to generate unit counts.

   Choose **Generate Unit and Amount Stock Count** to generate unit and amount counts.

> **Note:**
>
> After the generate stock count batch has completed, you can log onto the mobile application, and from the Main Menu, you can navigate to **Inventory Management / Stock Counts / Stock Count List** dialog. Select the generated stock count and you will notice stock count child records have been created for each department. The batch creates stock count groups for all items for all departments for the store, including items with SOH values of zero grouped by department. The stock count will be in new status, as will each of the child department records.

5. The next step of the process is to take a snapshot of the stock count. This is most often done manually but can also be done with an automated job. The snapshot must be taken before uploading the third-party flat file.

   Manual. On the mobile application, you will need to use the application to take the appropriate snapshot.

   Automated. For a unit and amount stock count, you can run the **Admin/Technical Maintenance/Job Admin** dialog previously used to generate the stock count, you can execute the **Stock Count Unit and Amount Snapshot** batch job.

   > **Note:**
   >
   > Selecting **Take Snapshot** in the mobile application or running the batch job takes a snapshot of the current SOH figure and assigns this to every item in the stock count records. The snapshot button is displayed only if there is an extracted **Third Party Stock Count** or **Unit and Amount stock count** on the **Stock Count List** screen. You must first select at least one record from the **Third Party Stock Count** in order for the snapshot to be taken. Status of the stock count will change to In Progress. This will indicate that the snapshot has occurred. The user will not be able to access the stock count records until the file has been uploaded. If the user double-clicks one of the department stock counts on the list screen, the application will prompt with the message "The stock count will not be accessible until the import process has completed". The user will not be able to drill into the detail screen if the third-party file has not yet been imported into the application.

6. Once the snapshot is taken and the workforce is done counting the items, the appropriate third-party stock count file should be loaded into the system.

7. Once the third-party count file is in place, you can access the **Admin/Technical Maintenance/Job Admin** dialog and execute the **Third Party Stock Count Import** batch job.

   > **Note:**
   >
   > When the batch is complete, each item within the count will be updated with the appropriate counted quantity and timestamps assigned. In addition, any item errors will be tracked and written to the database as rejected or unprocessed items. If auto authorize was not chosen, no further processing will take place. Authorization and rejected items management can then be dealt with.

Third Party Processing

1. When the third-party file import process starts, it will attempt to snapshot the stock count if the snapshot has not already taken place. A failure to snapshot will stop the job from processing.
2. Next, it updates all the counted quantity and dates on all the items from the file information. A failure in this step stops the job from processing.
3. It then attempts to perform the completion of each child count without the stock count. Completing the count does business processing on the counted information and moves the status of each completed child to the authorize phase. Any failures that occur are logged and the processing is halted.
4. If auto-authorize was not selected, the processing halts as the files are loaded and count phase completed.
5. If auto-authorize was selected, the processing releases all current user activity locks on the stock count, so it is not being used during further processing.
6. If the stock count was for all items, the automated processing will attempt to find and correct any errors within the rejected items, such as items found but not ranged at the store. This part of the processing will then attempt to range the items.
7. The stock count is then marked ready to approve and so that it can begin final authorization.
8. The process approves each stock count child individually. The batch error log keeps track of each authorization failure. If any authorization failed among the children record the process halts.
9. If the stock count is unit and amount and authorization succeeded, the process attempts to create an export file.

Third Party Recovery

1. Import Failure - If this occurs before or during loading the import file fails, you can begin the entire import process again.
2. Authorization Failure - If the import succeeds, but the authorization fails, you can run authorization recovery. Access the **Admin/Technical Maintenance/Job Admin** dialog and execute the **Stock Count Authorize Recovery** batch job.

D

Appendix: Unit and Amount Stock Counts Export

Unit and Amount Stock count authorization generates export file which can be uploaded to external inventory system. The stock count authorization process can be started by user through stock count authorization screen or be invoked by third party stock count batch for an auto-authorized unit amount stock count. The export files can be uploaded to merchandising system (for example, RMS) to update merchandising inventory with the actual physical stock count.

Export File Layout

See the Stock Count Results Export File Specification for file layout details. The generated file will be zipped into an archive with same file naming standard followed for the file generation. A complete file is added once the generated file has been zipped.

Export File Location

Export file directory is created by application installer. Integration admin will need to move the export data files from the application server export directory to a shared upload network location.

Export File Name

STK_<store id>_<schedule id>_<date in YYYYMMDDHH24MISS format>.dat

STK_<store id>_<schedule id>_<date in YYYYMMDDHH24MISS format>.zip

STK_<store id>_<schedule id>_<date in YYYYMMDDHH24MISS format>.zip.complete

E

Appendix: UPC Barcode

UPC-E items compress a normal 12-digit UPC-A item into six digits. The application has the ability to decompress UPC-E barcodes to UPC-A. A seventh digit acts as a check digit for the UPC-E number. When the user scans the UPC-E barcode, the application finds the UPC-A barcode and displays the item ID associated with it.

Differences between UPC-A and UPC-E

UPC-E is also called zero suppressed UPC because UPC-E compresses a normal twelve-digit UPC-A number into a six-digit code by suppressing the number system digit, trailing zeros in the manufacturers code and leading zeros in the product identification part of the bar code message. A seventh check digit is encoded into a parity pattern for the six main digits. UPC-E can thus be uncompressed back into a standard UPC-A twelve-digit number.

> **Note:**
>
> Most bar code readers can be configured to automatically convert six-digit UPC-E numbers to twelve-digit UPC-A numbers before they are transmitted to a host computer.

The main difference between a UPC-A symbol and a UPC-E symbol is the size. The following image presents a UPC-A bar code (left) and the same data encoded as a UPC-E bar code (right):

Figure E-1 UPC-A and UPC-E Differences

To convert between UPC-A and UPC-E bar code numbers, you can use the following table or try online UPC-E converter program. In the following, the number 0 and each of the letters (a, b, c, d and e) represent individual digits in the bar code message. The letter X represents the UPC check digit.

Table E-1 UPC Conversion Table

UPC-A Number	Equivalent UPC-E	Notes
0ab00000cdeX	abcde0X	Manufacturer code must have two leading digits with three trailing zeros and the item number is limited to three digits (000 to 999).
0ab10000cdeX	abcde1X	Manufacturer code must have three leading digits ending with 1 and two trailing zeros. The item number is limited to three digits.
0ab20000cdeX	abcde2X	Manufacturer code must have three leading digits ending with 2 and two trailing zeros. The item number is limited to three digits.
0abc00000deX	abcde3X	Manufacturer code must have three leading digits and two trailing zeros. The item number is limited to two digits (00 to 99).
0abcd00000eX	abcde4X	Manufacturer code must have four leading digits with one trailing zero and the item number is limited to one digit (0 to 9).
0abcde00005X	abcde5X	Manufacturer code has all five digits. The item number is limited to a single digit consisting of either 5, 6, 7, 8 or 9.
0abcde00006X	abcde6X	
0abcde00007X	abcde7X	
0abcde00008X	abcde8X	
0abcde00009X	abcde9X	

Conversion between UPC-A and UPC-E

Not all UPC-A numbers can be compressed to UPC-E. These codes with a corresponding UPC-E code must have at least four zeros. The requirements are:

1. If the manufacturer code ends with 000, 100, or 200, the UPC-E code consists of the first two characters of the manufacturer code, the last three characters of the product code, followed by the third character of the manufacturer code. In this case, the product code must be 00000 and 00999.
2. If the manufacturer code ends with 00 but does not meet the first requirement, the UPC-E code consists of the first three characters of the manufacturer code, the last two characters of the product code, followed by digit 3. The product code can only contain two digits (00000 to 00099).
3. If the manufacturer code ends in 0 but none of the previous qualifies, the UPC-E consists of the first four digits of the manufacturer code and the last digit of the product code, followed by the digit 4. The product code in this case can only contain one digit (00000 to 00009).
4. If the manufacturer code ends with non-zero digit, the UPC-E code consists of the manufacturer code and the last digit of the product code. In this case the product case can only be one from 00005 to 00009 because 0 through 4 has been used for the previous four cases.

F

Appendix: EICS Provided URLs

> **Note:**
>
> The <Region Name> and <Customer Subnamespace> part of the URL should be replaced with the one specific to your environment. This will be the same as your cloud service Application URL provided in the Welcome email.

EICS web-client URL

Table F-1 EICS Application URL

	URL
EICS web-client	https://rex.retail.<Region Name>.ocs.oraclecloud.com/<Customer Subnamespace>/siocs-web-client

SOCS (connections config) URL

Table F-2 SOCS Connections URL

	URL
SOCS (Connections Config)	https://rex.retail.<Region Name>.ocs.oraclecloud.com/<Customer Subnamespace>/siocs-client-services/oracle.retail.sim.mobile.client.SimMobile/connections.xml

EICS Web Service URLs

Table F-3 EICS Web Service URLs

SIM-WS	URL
	https://rex.retail.<Region Name>.ocs.oraclecloud.com/<Customer Subnamespace>/ActivityLockBean/ActivityLockService?wsdl
	https://rex.retail.<Region Name>.ocs.oraclecloud.com/<Customer Subnamespace>/FulfillmentOrderDeliveryBean/FulfillmentOrderDeliveryService?wsdl
	https://rex.retail.<Region Name>.ocs.oraclecloud.com/<Customer Subnamespace>/FulfillmentOrderPickBean/FulfillmentOrderPickService?wsdl
	https://rex.retail.<Region Name>.ocs.oraclecloud.com/<Customer Subnamespace>/FulfillmentOrderReversePickBean/FulfillmentOrderReversePickService?wsdl

Table F-3 (Cont.) EICS Web Service URLs

SIM-WS	URL
	https://rex.retail.<Region Name>.ocs.oraclecloud.com/<Customer Subnamespace>/InventoryAdjustmentBean/InventoryAdjustmentService?wsdl
	https://rex.retail.<Region Name>.ocs.oraclecloud.com/<Customer Subnamespace>/ItemBasketBean/ItemBasketService?wsdl
	https://rex.retail.<Region Name>.ocs.oraclecloud.com/<Customer Subnamespace>/OrderRequestBean/OrderRequestService?wsdl
	https://rex.retail.<Region Name>.ocs.oraclecloud.com/<Customer Subnamespace>/POSTransactionBean/POSTransactionService?wsdl
	https://rex.retail.<Region Name>.ocs.oraclecloud.com/<Customer Subnamespace>/ProductGroupScheduleBean/ProductGroupScheduleService?wsdl
	https://rex.retail.<Region Name>.ocs.oraclecloud.com/<Customer Subnamespace>/ProductGroupBean/ProductGroupService?wsdl
	https://rex.retail.<Region Name>.ocs.oraclecloud.com/<Customer Subnamespace>/ReplenishmentGapBean/ReplenishmentGapService?wsdl
	https://rex.retail.<Region Name>.ocs.oraclecloud.com/<Customer Subnamespace>/RfidInventoryBean/RfidInventoryService?wsdl
	https://rex.retail.<Region Name>.ocs.oraclecloud.com/<Customer Subnamespace>/ShelfAdjustmentBean/ShelfAdjustmentService?wsdl
	https://rex.retail.<Region Name>.ocs.oraclecloud.com/<Customer Subnamespace>/ShelfReplenishmentBean/ShelfReplenishmentService?wsd
	https://rex.retail.<Region Name>.ocs.oraclecloud.com/<Customer Subnamespace>/StockCountBean/StockCountService?wsdl
	https://rex.retail.<Region Name>.ocs.oraclecloud.com/<Customer Subnamespace>/StoreBean/StoreService?wsdl
	https://rex.retail.<Region Name>.ocs.oraclecloud.com/<Customer Subnamespace>/StoreFulfillmentOrderBean/StoreFulfillmentOrderService?wsdl
	https://rex.retail.<Region Name>.ocs.oraclecloud.com/<Customer Subnamespace>/StoreInventoryBean/StoreInventoryService?wsdl
	https://rex.retail.<Region Name>.ocs.oraclecloud.com/<Customer Subnamespace>/StoreInventoryIsnBean/StoreInventoryIsnService?wsdl
	https://rex.retail.<Region Name>.ocs.oraclecloud.com/<Customer Subnamespace>/StoreItemPriceBean/StoreItemPriceService?wsdl
	https://rex.retail.<Region Name>.ocs.oraclecloud.com/<Customer Subnamespace>/StoreNotificationBean/StoreNotificationService?wsdl
	https://rex.retail.<Region Name>.ocs.oraclecloud.com/<Customer Subnamespace>/StoreShipmentManifestBean/StoreShipmentManifestService?wsdl
	https://rex.retail.<Region Name>.ocs.oraclecloud.com/<Customer Subnamespace>/StoreShipmentReasonBean/StoreShipmentReasonService?wsdl
	https://rex.retail.<Region Name>.ocs.oraclecloud.com/<Customer Subnamespace>/StoreTicketBean/StoreTicketService?wsdl
	https://rex.retail.<Region Name>.ocs.oraclecloud.com/<Customer Subnamespace>/StoreTransferBean/StoreTransferService?wsdl
	https://rex.retail.<Region Name>.ocs.oraclecloud.com/<Customer Subnamespace>/TransferDeliveryBean/TransferDeliveryService?wsdl

Table F-3 (Cont.) EICS Web Service URLs

SIM-WS	URL
	https://rex.retail.<Region Name>.ocs.oraclecloud.com/<Customer Subnamespace>/TransferShipmentBean/TransferShipmentService?wsdl
	https://<eics_external_load_balancer_address>/<CUST_ENV>/VendorDeliveryBean/VendorDeliveryService?wsdl
	https://<eics_external_load_balancer_address>/<CUST_ENV>/VendorReturnBean/VendorReturnService?wsdl
	https://<eics_external_load_balancer_address>/<CUST_ENV>/VendorShipmentBean/VendorShipmentService?wsdl

EICS-RICS Integration URLs

EICS-RICS Message Publishing

Publisher Web Service URL

The message publisher service (SIM -> RIB) is hosted by RIB-SIM. EICS installation takes in the message publisher service WSDL URLs and updates the DB system configuration table.

The WSDL URL can also be updated via EICS System Configuration UI post install.

Table F-4 RIB Message Publishing

Database CONFIG_SYSTEM Name	System Configuration UI Display Name
integration.rib.publisher.wsdl.url	Integration Publisher Web Service URL

Web Service User Management

Oracle RICS RIB manages the RIB publisher user.

The publisher user (for example ribadmin) is created as part of the RIB/RTG install.

EICS chef scripts takes the input to add to EICS credential stores.

EICS Install Properties

- input.sim.integration.rib.user.alias=rib-user
- This property is for SIM calling the ApplicationMessagePublishingService. The username and password for the alias should match the user which is created as part of RIB-SIM and belongs to ribAdminGroup.
- The alias name should be unique within the domain, and should not conflict with database data source user alias (for example, in some environment, the SIMRIBAlias is used for db user sim01_RIB, if that is case, choose a different alias for sim -> rib publishing user)

Setting Up RIB Message Injector

Injector Web Service URL

RIB Message Injector Service is hosted in EICS application server.

The WSDL URL is:

https://rex.retail.<Region Name>.ocs.oraclecloud.com/<Customer Subnamespace>/ApplicationMessageInjector-Bean/InjectorService?WSDL

Web Service User Management

The injector user (for example, sim_int) must belong to integration_users IDCS or OCI IAM Application Role, the injector user needs to be created as part of EICS provisioning process.

EICS Installation Properties

- input.sim.integration.rib.inject.user.alias
- This property is used for RIB-SIM -> SIM (RIB-SIM calling SIM's injector service), the user is hosted in SIM app and assign integration role (integration_users IDCS or OCI IAM Application Role), the user is created as part of SIM install (for example, sim_int user)

  See *Oracle® Retail Integration Bus Implementation Guide* Provisioning InjectorSrvice URL

BI Related URL

Table F-5 BI Related URL

	URL
xmlpserver	https://<GBUA-URL>/<TENANT_ID>/xmlpserver

ORDS (Apex Data Viewer) URL

Table F-6 ORDS (Apex Data Viewer) URL

	URL
ORDS (Apex Data Viewer)	https://rex.retail.<Region Name>.ocs.oraclecloud.com/<Customer Subnamespace>/ords

G

Appendix: MPS Message Types

DCS Message Types

Incoming messages type from Data Collection System

Table G-1 DCS Message Types

MPS Family	MPS Message Type	Message
DcsAllocation	DcsAllocation	Allocation
	DcsAllocationImport	Allocation
DcsAsn	DcsAsnAdjustment	ASN Adjustment (adjusts delivery)
	DcsAsnShipment	ASN Shipment (creates delivery)
	DcsAsnReceipt	ASN Receipt (receives delivery)
	DcsTransferDelv	Transfer Delivery
	DcsTransferDelvDelete	Transfer Delivery Delete
	DcsTransferDelvImport	Transfer Delivery
	DcsTransferDelvUinImport	Transfer Delivery UIN
	DcsTransferShipImport	Transfer Shipment
	DcsTransferShipUinImport	Transfer Shipment
	DcsVendorDelv	Vendor Delivery
	DcsVendorDelvCancel	Vendor Delivery Cancel
	DcsVendorDelvImport	Vendor Delivery
	DcsVendorDelvUinImport	Vendor Delivery UIN
	DcsVendorShipUinImport	Vendor Shipment UIN
	DcsVendorReturnShipmentImport	Vendor Shipment
DcsDiff	DcsDiff	Differentiator
	DcsDiffType	Differentiator Type
DcsFiscalDocument	DcsFiscalDocument	Fiscal Document
DcsHierarchy	DcsItemHierarchy	Item Hierarchy (department, class, or subclass)
DcsItem	DcsItem	Item
	DcsItemApprove	Item Approved
	DcsItemImage	Item Image
	DcsItemRelated	Item Relationship
	DcsItemTicketType	Item Ticket Type
	DcsItemUda	Item User Defined Attributes

Table G-1 (Cont.) DCS Message Types

	DcsItemUinImport	Item UIN
DcsItemLocation	DcsItemWarehouse	Warehouse Item
	DcsItemWarehouseReq	Warehouse Item (bulk processing request)
	DcsItemFinisher	Finisher Item
	DcsItemFinisherReq	Finisher Item (bulk processing request)
	DcsItemStore	Store Item
	DcsItemStoreReq	Store Item (bulk processing request)
	DcsItemStoreReplenish	Store Item Replenishment
DcsOrder	DcsOrderApprove	Purchase Order (Approved)
	DcsOrderReview	Purchase Order (For Review as Store Order)
	DcsOrderImport	Purchase Order
	DcsStoreOrder	Store Order (third-party integration)
DcsPartner	DcsPartner	Partner (Finisher)
DcsPrice	DcsPrice	Price
DcsRtv	DcsVendorReturn	Return To Vendor Request
	DcsVendorReturnImport	Return To Vendor Request
DcsStore	DcsAddress	Entity Address (store, finisher, supplier, warehouse)
	DcsStore	Store
DcsSupplier	DcsSupplier	Supplier
DcsSupplierItem	DcsSupplierItem	Supplier Item
	DcsSupplierItemCountry	Supplier Item Country
	DcsSupplierItemCountryDim	Supplier Item Country Dimensions
	DcsSupplierItemCountryManu	Supplier Item Country of Manufacture
	DcsSupplierItemUom	Supplier Item Unit Of Measure
DcsTransfer	DcsTransferRequest	Transfer Request
	DcsTransferReview	Transfer Review (For Review as Store Order)
	DcsTransferImport	Transfer Import
DcsUda	DcsUda	User Defined Attribute
DcsWarehouse	DcsWarehouse	Warehouse
	DcsWarehouseAdjust	Warehouse Inventory Adjustment

DCS Message Type Source

Table G-2 DCS Message Type Source

MPS Family	MPS Message Type	ICL Direct DB	REST Service	File Import
DcsAllocation	DcsAllocation	X	X	
	DcsAllocationImport			X
DcsAsn	DcsAsnAdjustment	X	X	
	DcsAsnShipment	X		
	DcsAsnReceipt*	**		
	DcsTransferDelv		X	
	DcsTransferDelvDelete		X	
	DcsTransferDelvImport			X
	DcsTransferDelvUinImport			X
	DcsTransferShipImport			X
	DcsTransferShipUinImport			X
	DcsVendorDelv		X	
	DcsVendorDelvCancel		X	
	DcsVendorDelvImport			X
	DcsVendorDelvUinImport			X
	DcsVendorShipUinImport			X
	DcsVendorReturnShipmentImport			X
DcsDiff	DcsDiff	X	X	
	DcsDiffype	X	X	
DcsFiscalDocument	DcsFiscalDocument	X		
DcsHierarchy	DcsItemHierarchy	X	X	
DcsItem	DcsItem	X	X	
	DcsItemApprove	X		
	DcsItemImage	X	X	
	DcsItemRelated	X	X	
	DcsItemTicketType	X		
	DcsItemUda	X	X	
	DcsItemUinImport			X
DcsItemLocation	DcsItemWarehouse	**		
	DcsItemWarehouseReq	X	X	
	DcsItemFinisher	**	X	
	DcsItemFinisherReq	X	X	
	DcsItemStore	**	X	
	DcsItemStoreReq	X	X	
	DcsItemStoreReplenish			

Table G-2 (Cont.) DCS Message Type Source

DcsOrder	DcsOrderApprove	X	X		
	DcsOrderReview	X			
	DcsOrderImport				X
	DcsStoreOrder		X		
DcsPartner	DcsPartner	X	X		
DcsPrice	DcsPrice	X			X
DcsRtv	DcsVendorReturn	X	X		
	DcsVendorReturnImport				X
DcsStore	DcsAddress	X	X		
	DcsStore	X	X		
DcsSupplier	DcsSupplier	X	X		
DcsSupplierItem	DcsSupplierItem	X	X		
	DcsSupplierItemCountry	X	X		
	DcsSupplierItemCountryDim	X	X		
	DcsSupplierItemCountryManu	X	X		
	DcsSupplierItemUom	X	X		
DcsTransfer	DcsTransferRequest	X	X		
	DcsTransferReview	X			
	DcsTransferImport				X
DcsUda	DcsUda	X	X		
DcsWarehouse	DcsWarehouse	X	X		
	DcsWarehouseAdjust		X		

** Indicates it is a byproduct of mps processing and not directly used by any integration point