

Oracle® Retail Merchandising System
Operations Guide, Volume 3 - Back End Configuration and
Operations
Release 16.0.3
F29570-01

March 2020

F29570-01

Copyright © 2020, Oracle. All rights reserved.

Primary Author:

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

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

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

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

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

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

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

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

Value-Added Reseller (VAR) Language

Oracle Retail VAR Applications

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

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

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

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

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

You acknowledge and confirm that Oracle grants you use of only the object code of the VAR Applications. Oracle will not deliver source code to the VAR Applications to you. Notwithstanding any other term or condition of the agreement and this ordering document, you shall not cause or permit alteration of any VAR Applications. For purposes of this section, "alteration" refers to all alterations, translations, upgrades, enhancements, customizations or modifications of all or any portion of the VAR Applications including all reconfigurations, reassembly or reverse assembly, re-engineering or reverse engineering and recompilations or reverse compilations of the VAR Applications or any derivatives of the VAR Applications. You acknowledge that it shall be a breach of the agreement to utilize the relationship, and/or confidential information of the VAR Applications for purposes of competitive discovery.

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

Contents

Contents	v
Send Us Your Comments	xi
Preface	xiii
Audience	xiii
Documentation Accessibility.....	xiii
Customer Support.....	xiii
Review Patch Documentation.....	xiii
Improved Process for Oracle Retail Documentation Corrections	xiv
Oracle Retail Documentation on the Oracle Technology Network.....	xiv
Conventions.....	xiv
1 Introduction	1
Contents Covered in this Volume.....	1
2 Technical Architecture	3
Overview	3
Oracle Application Development Framework (ADF).....	3
Retail Fusion Platform.....	5
RMS Legacy Services.....	5
Data Access Patterns	6
Data Storage.....	6
3 Backend System Administration and Configuration	7
Managing the Notifications Feature	7
Overview of Notifications in Retail Applications	7
Purging the Notifications Table	8
Notification ReST Services.....	9
Enabling Mobile Push Notifications.....	9
Managing Application Navigator.....	10
Managing Functional Security	10
Introduction to Retail Roles.....	10
Retail Role Hierarchy	12
Default Security Reference Implementation	12
Extend the Default Security Reference Implementation	61
Disabling Content	62
Safe Mode.....	62
Disabling Links in the Sidebar Menu.....	62
Managing Oracle Metadata Services (MDS)	62
Overview of Oracle Metadata Services (MDS)	63
Using the System MBean Browser and the MDSAAppRuntime MBean	63
Exporting All MDS Customizations.....	66
Exporting MDS Customization for a Specific User	67

Deleting All MDS Customizations for a User	68
Deleting a Customization for a Specific Page for All the Users	68
Deleting a Customization for a Specific Page for a Particular User	69
Importing all MDS Customizations	70
Importing a Specific Page Customization for a User	71
Creating Metadata Labels	71
Promoting Metadata Labels	72
Listing Metadata Labels	72
Deleting Metadata Labels	72
4 Security in Retail Applications	73
SSO Setup for RAF Applications	73
Displaying External Application Contents in Non-SSO Environments	73
5 Web Services in Retail Applications	75
Common Characteristics of Retail Application ReSTful Web Services	75
Deployment	75
Security	75
Standard Request and Response Headers	76
Standard Error Response	76
List of ReSTful Web Services	76
Platform ReSTful Web Services	76
6 Customization of Retail Applications	83
Prerequisite Concepts	83
Understanding the Deployment of Retail Applications	83
Understanding the Retail application User Interface	84
Supported Customization Scenarios	86
Adding a Custom Shared Library	86
Creating New ADF Contents	96
Adding or Modifying an Item in the Reports Menu	97
Dashboard Customization Scenarios	98
Adding Contextual Reports	122
Enabling Dynamic Task Items in the Retail Application	162
In-Context Launchable Task Flows	183
Limitations of In-Context Launch via URLs	183
List of In-Context Launchable Task Flows	183
7 Pro*C Restart and Recovery	185
Table of Description and Definition	185
RESTART_CONTROL	186
RESTART_PROGRAM_STATUS	187
RESTART_PROGRAM_HISTORY	188
RESTART_BOOKMARK	189
V_RESTART_<X>	190
Restart and Recovery Data Model Design	190

Physical Set-Up.....	190
Table and File-Based Restart/Recovery.....	191
API Functional Descriptions.....	193
RESTART_INIT	193
RESTART_FILE_INIT.....	193
RESTART_COMMIT	194
RESTART_FILE_COMMIT.....	194
RESTART_CLOSE	194
PARSE_ARRAY_ARGS.....	195
RESTART_FILE_WRITE	195
RESTART_CAT.....	195
Restart Headers and Libraries.....	195
RESTART.H.....	195
STD_REST.H.....	195
Updating Restart Headers and Libraries.....	196
Query-Based Commit Thresholds	199
8 Pro*C Multi-Threading.....	201
Threading Description.....	201
Threading Function for Query-Based	201
Restarting View for Query-Based.....	202
Thread Scheme Maintenance.....	203
File-Based.....	203
Query-Based	204
Batch Maintenance.....	204
Scheduling and Initialization of Restart Batch.....	204
Pre-Processing and Post-Processing.....	205
ksh Driven Batch Programs	205
9 Pro*C Array Processing	207
10 Pro*C Input and Output Formats.....	209
General Interface Discussion.....	209
Standard File Layouts	209
Detail-Only Files	209
Master and Detail Files.....	210
11 RETL Program Overview for the RMS-RPAS Interface.....	213
Oracle Retail ETL Architecture	213
RETL Program Overview	214
Configuration	214
Program Return Code	217
Program Status Control Files.....	217
File Naming Conventions.....	217
Restart and Recovery.....	217
Message Logging	218

Daily Log File	218
Format	218
Program Error File	218
RMSE Reject Files	219
Schema Files Overview	219
Command Line Parameters	219
rmse_rpas_config.env	220
RMSE I/O File Names.....	221
Typical Run and Debugging Situations	222
RPAS/AIP Configuration	222
RETL Program Overview for the RMS-Time-Phased Inventory Planning Tool Interface.....	223
Configuration	223
Program Return Code	224
Program Status Control Files	224
Message Logging	226
RMSE and Transformation Reject Files.....	227
Schema Files Overview	227
Command Line Parameters.....	227
Typical Run and Debugging Situations.....	228
12 Internationalization	231
Translation	231
Translation Support for Multiple Entities.....	232
RMS User Interface Language Display Settings	232
Maintenance of Translation Values through the User Interface.....	232
Maintenance of Translation Values through Spreadsheet Uploads.....	233
13 Integrating RMS with Store Inventory Management	235
Supplier	236
Merchandise Hierarchy	236
Warehouse	236
SIM Store	236
Inventory Adjustment Reason	236
Adding Inventory Adjustment Reason in SIM	237
Mapping Inventory Adjustment Reason in RMS	237
Diff ID	237
Item With/Without UIN and Item Locations	237
SIM GUI Item Look up.....	238
Transactions.....	238
Purchase Order.....	238
Transfers.....	239
Return to Warehouse.....	241
Return to Vendor	243

Store Orders.....	244
Inventory Adjustment.....	244
Stock Count.....	245
Price Change.....	246
14 Integrating RMS with Oracle E-Business Suite Financials Using Oracle Retail	
Financial Integration.....	249
Participating Applications.....	249
Assumptions and Dependencies.....	249
Data Setup.....	250
RMS Data Set up and Configuration.....	250
ReIM Data Setup and Configuration.....	254
ReIM Transactional Maintenance.....	257
Calculation of TRANS_AMOUNT.....	257
Generation of Outgoing Data.....	258
Validation of Accounts When Posting Financial Entries.....	258
Validation of Accounts When Prepaying a Merchandise Invoice.....	259
Maintenance of Valid Accounts.....	260
15 Integrating RMS with PeopleSoft Financials Using Oracle Retail Financial	
Integration.....	261
Participating Applications.....	261
Assumptions and Dependencies.....	261
Data Constraints.....	262
Data Setup.....	262
RMS Data Setup and Configuration.....	262
Configuring Drill Back and Forward Web Services.....	266
ReIM Data Setup and Configuration.....	267
CURRENCY_PRECISION.....	267
Chart of Accounts Setup.....	267
ReIM Transactional Maintenance.....	271
Calculation of TRANS_AMOUNT.....	271
Generation of Outgoing Data.....	271
Validation of Accounts When Posting Financial Entries.....	272
Maintenance of Valid Accounts.....	272
Building and Posting Reference IDs.....	273
Drilling Back.....	273
Drilling Back to RMS and ReSA from PeopleSoft Enterprise Financials.....	273
Drilling Back to ReIM from PeopleSoft Enterprise Financials.....	274
Drilling Forward.....	275
Drilling Forward From RMS/ReSA to PeopleSoft Enterprise Financials.....	275
Drilling Forward From ReIM to PeopleSoft Enterprise Financials.....	275
16 Understanding Data Access Schema.....	277
DAS Views.....	278

17 Using Oracle Wallet 279

Send Us Your Comments

Oracle Retail Merchandising System Operations Guide, Volume 3 - Back End Configuration and Operations, Release 16.0.3

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

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

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

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

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

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

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

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

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

Preface

This *Retail Merchandising System Operations Guide, Volume 3 - Back End Configuration and Operations* provides critical information about the processing and operating details of Oracle Retail Merchandising System (RMS), including the following:

- System configuration settings
- Technical architecture
- Functional integration dataflow across the enterprise
- Batch processing

Audience

This guide is for:

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

Documentation Accessibility

For information about Oracle's commitment to accessibility, visit the Oracle Accessibility Program website at

<http://www.oracle.com/pls/topic/lookup?ctx=acc&id=docacc>.

Access to Oracle Support

Oracle customers that have purchased support have access to electronic support through My Oracle Support. For information, visit

<http://www.oracle.com/pls/topic/lookup?ctx=acc&id=info> or visit

<http://www.oracle.com/pls/topic/lookup?ctx=acc&id=trs> if you are hearing impaired.

Customer Support

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

<https://support.oracle.com>

When contacting Customer Support, please provide the following:

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

Review Patch Documentation

When you install the application for the first time, you install either a base release (for example, 16.0) or a later patch release (for example, 16.0.3). If you are installing the base release or additional patch releases, read the documentation for all releases that have occurred since the base release before you begin installation. Documentation for patch

releases can contain critical information related to the base release, as well as information about code changes since the base release.

Improved Process for Oracle Retail Documentation Corrections

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

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

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

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

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

Oracle Retail Documentation on the Oracle Technology Network

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

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

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

Conventions

Navigate: This is a navigate statement. It tells you how to get to the start of the procedure and ends with a screen shot of the starting point and the statement “the Window Name window opens”.

This is a code sample

It is used to display examples of code

Introduction

Welcome to the Oracle Retail Merchandising System Operations Guide, Volume 3 - Back End Configuration and Operations. The guide is designed to inform you about the 'backend' of RMS: data inputs, processes, and outputs. As a member of the Oracle Retail family, RMS provides many benefits of Enterprise Application Integration (EAI).

Contents Covered in this Volume

This volume describes the important features that are necessary to run the Pro*C programs and the RETL programs associated with RMS. Additional RMS configuration, RMS Integration, Operations, and Customization information is also included in this volume.

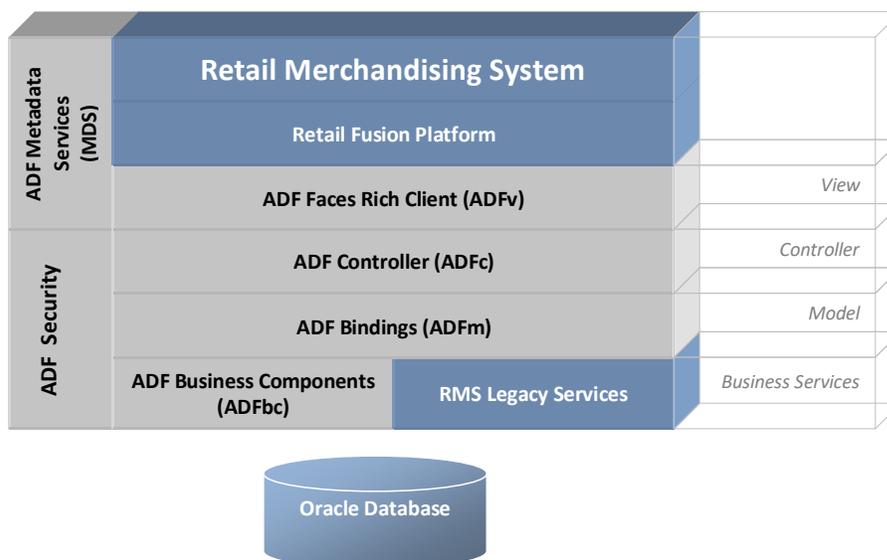
Note: For setting up Business Intelligence Publisher, see *Oracle Retail Merchandising System Installation Guide*.

Technical Architecture

Overview

Retail Applications are based on the Oracle Application Development Framework (ADF). The following diagram shows the key components that make up the architecture of Retail Applications.

Navigate: File – New. The New Document pane opens.



Oracle Application Development Framework (ADF)

The Oracle Application Development Framework (ADF) supports organizations in building cutting-edge rich enterprise business applications that can be customized and personalized in all dimensions. Customizations are global changes, visible to all users that are performed by an administrator. Personalization is user-made changes that are only visible to the person making the change. ADF is based on the Java Enterprise Edition platform.

Model View Controller (MVC) Architectural Pattern

Applications built using ADF follow a Model-View-Controller (MVC) architectural pattern. The goal of the MVC pattern is to clearly separate the application's functionality into a set of cooperating components.

ADF provide a set of components that realize the goals of each part of MVC pattern.

- Model is realized by the ADF Bindings Layer
- Controller is realized by the ADF Controller Layer
- View is realized by the ADF Faces Layer

- ADF Business components and other backend components that sit below the model layer are called Business Services.

ADF Security

- Standards-based (Oracle Platform Security Services (OPSS)) security framework with default roles and permissions.
- Tools to generate the file-based identity store (for both Oracle Internet Directory and AD) based on the framework.
- Tools to migrate the file-based security store in to the database for QA and production environments.
- Reference implementation for clients to manage the security based on their business needs.
- OPSS-based batch security framework (RAF).
- Tools/documentation to implement centralized logout in SSO (Oracle Access Management (OAM)) environments.

ADF View (ADFv)

The View layer provides the user interface to the application. The view layer uses HTML, rich Java components or XML and its variations to render the user interface. JSF based tag libraries are used for displaying the UI.

ADF Controller (ADFc)

The ADF Controller layer controls the application's flow. Web based applications are composed of multiple web pages with dynamic content. The controller layer manages the flow between these pages. Different models can be used when building this later. The most prominent architecture for Java-based web applications relies on a servlet that acts as the controller. The Apache Jakarta Struts controller, an open source framework controller, is the de facto standard for Java-based web systems. Oracle ADF uses the Struts controller to manage the flow of web applications.

ADF Business Components (ADFbc)

The business service layer manages the interaction with a data persistence layer. It provides services as data persistence, object/relational mapping, transaction management and business logic execution.

Business Components easily map the database object and extend it with business logic, validation and so on.

The idea behind Business Components is to abstract the data layer from the view layer. This is a key concept in the MVC pattern. Business Components will expose the interface to the view layer by using an application module that contains the View Object. Those view objects contain a specific usage of the data layer.

ADF Business Components implements the business service through the following set of cooperating components:

- Entity object – An entity object represents a row in a database table and simplifies modifying its data by handling all data manipulation language (DML) operations for you. It can encapsulate business logic for the row to ensure that your business rules are consistently enforced. You associate an entity object with others to reflect relationships in the underlying database schema to create a layer of business domain objects to reuse in multiple applications.

- View object – A view object represents an SQL query. You use the full power of the familiar SQL language to join, filter, sort, and aggregate data into the exact shape required by the end-user task. This includes the ability to link a view object with others to create master-detail hierarchies of any complexity. When end users modify data in the user interface, view objects collaborate with entity objects to consistently validate and save the changes.
- Application module – An application module is the transactional component that UI clients use to work with application data. It defines an updatable data model and top-level procedures and functions (called service methods) related to a logical unit of work related to an end-user task.

ADF Model (ADFm)

This component acts as the connector between the view and business logic layers.

The Model layer connects the Business Services to the objects that use them in the other layers. Oracle ADF provides a Model layer implementation that sits on top of Business Services, providing a single interface that can be used to access any type of Business Services.

Developers get the same development experience when binding any type of Business Service layer implementation to the view and Controller layers. The Model layer in Oracle ADF served as the basis for JSR 227, A Standard Data binding & Data Access Facility for J2EE.

Oracle Metadata Services (MDS)

The ability of an application to adapt to changes is a necessity that needs to be considered in the application design and that should drive the selection of the development platform and architecture. Flexible business applications must be able to adapt to organizational changes, different end user preferences and changes in the supported business are required.

MDS is the customization and personalization framework integral to Oracle Fusion Middleware and a key differentiator of the Oracle development platform. MDS provides a repository for storing metadata for applications, such as customizations and persisted personalization files and configurations.

Retail Applications allow the following through MDS:

- Personalization of saved searches through MDS.
- Implicit personalization of few ADF UI attributes.

Retail Fusion Platform

The Retail Fusion Platform (commonly referred to as Platform) is a collection of common, reusable software components that serve as foundation for building Oracle Retail's next generation ADF-based applications. The Platform imposes standards and patterns along with a consistent look and feel for Oracle Retail's ADF applications.

RMS Legacy Services

RMS Legacy services are implemented using PL/SQL packages. The invocation of these backend components are encapsulated in ADF business components typically as Java wrappers that internally make use of JDBC to invoke the PL/SQL package calls

Data Access Patterns

Database interaction between the middle tier and the database is done using the industry standard Java Database Connectivity Protocol (JDBC). JDBC facilitates the communication between a Java application and a relational database.

Database Access Using ADFbc

JDBC is ingrained within Oracle ADF Business Components as the primary mechanism for its interaction between the middle tier and the database. SQL is realized within ADF business components to facilitate create, read, update and delete (CRUD) actions.

Connection Pooling

When the application 'disconnects' a connection, the connection is saved into a pool instead of being actually disconnected. A standard connection pooling technique, this saved connection enables Retail Applications to reuse the existing connection from a pool. In other words, the application does not have to complete the connection process for each subsequent connection.

Data Storage

The Oracle Database realizes the database tier in a Retail Application's architecture. It is the application's storage platform, containing the physical data (user and system) used throughout the application. The database tier is only intended to handle the storage and retrieval of information and is not involved in the manipulation or in the delivery of the data. This tier responds to queries; it does not initiate them.

Backend System Administration and Configuration

Managing the Notifications Feature

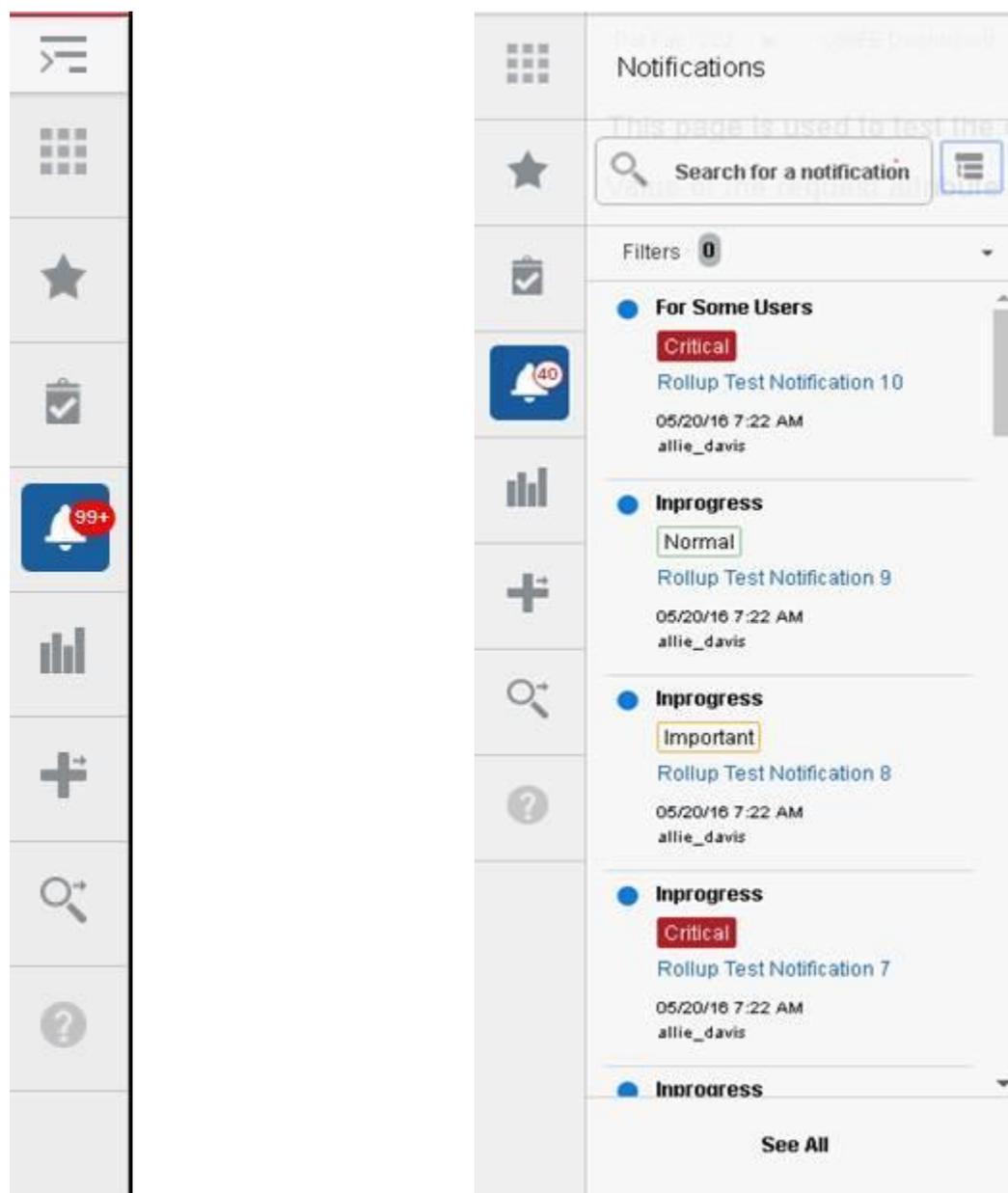
Overview of Notifications in Retail Applications

Retail Applications provide the ability to notify authenticated users in the application when business events occur. An example of a business event is when a background asynchronous task such as an approval of an allocation or a validation of a price change has completed. Typically, the user is expected to take action on these notifications.

The notification icon on the Retail Application's navigation pane displays the count of unread notifications for the user. When a notification is generated, this count is incremented immediately and thereafter refreshed at regular configurable intervals.

The sliding sidebar menu expands when the user clicks on the notification icon. The expanded sidebar menu will show the latest notifications. Clicking the "See All" option shows all the notifications in a new tab.

Clicking on the link for each launchable notification opens the specific UI flow that allows the user to address any pending action for the notification. Illustrations follow.



Sliding Sidebar showing the Notification Count and the Notifications Sidebar Menu

Purging the Notifications Table

Notifications are represented as rows in the table RAF_NOTIFICATION. Over a period of time, depending on how notifications are being generated by the application, the size of this table can grow continually, potentially degrading performance of the notifications feature.

It is recommended that retailers purge this table periodically. You can do this via the Purge feature available in the Retail Application Administration Console (RAAC). For more details on the Manage Notifications feature in RAAC, see the latest *Merchandising Implementation Guide*.

You could also setup a batch job that runs periodically, that invokes the PL-SQL as shown below:

```
declare
    retval number(10);
begin
    -- remove Notifications from the Allocation application that have
    exceeded . -- their retention duration.
    retval := raf_notification_task_pkg.DEL_NOTIF_PAST_RETENTION('ALC');
end;
```

Notification ReST Services

ReST Endpoints have been exposed by the Notifications framework to ease the integration concerns with disparate applications.

Enabling Mobile Push Notifications

Users of the mobile version of the Retail Application can receive notifications in their mobile devices using push notifications.

Push notifications for Mobile Retail Applications need to be enabled by the Retailer. This section describes the specific steps to enable this feature.

Note: Push notification is a native feature in modern mobile devices so the steps to enable this service on different platforms will be different.

Prerequisite Apple Certificate and Provisioning Profile

Using push notifications on iOS requires retailers to acquire an Apple Push Notification (APN) Service key, certificate and Provisioning profile. These files are specific to the application and are linked to the developer account that Apple generated them from. Refer to Apple's developer site for instructions on how to create an account and generate these credentials.

Prerequisite: Android Google Cloud Messaging Registration

Using push notifications on Android requires retailers to register the mobile Retail Application with Google Cloud Messaging (GCM). An API key and sender ID will be generated from the account created during registration. Refer to Google's developer site for instructions on how to create an account and generate these credentials.

Server Configuration

Once credentials to the specific platforms are available, retailers need to add these to the server.

For Apple Push Notifications, retailers must import the key and certificate from Apple into weblogic. In order to do this, a keystore (JKS) file must be created using the Java keystore utility. The key and certificate can then be imported into the created keystore file. Retailers can refer to Oracle Java SE documentation for details on the keystore utility.

Once created, the keystore can be uploaded into the weblogic system app stripe. Retailers can use the weblogic scripting tool to run a Python script similar to the one shown below.

The example shows that the keystore is named RetailAppsPushServices.

```

user = "admin username"
password = "admin password"
url = "[weblogic server]:[port]"
keypass = "keystore password"
keypath = "keystore filepath"
apnsAlias = "key alias"
apnsPass = "key password"
connect(user, password, url)
svc = getOpssService(name='KeyStoreService')
svc.importKeyStore(appStripe='system', name='RetailAppsPushServices', type='JKS',
filepath=keypath, password=keypass, permission=true, aliases=apnsAlias,
keypasswords=apnsPass)
exit('y')

```

For more details on how to import the created java keystores into weblogic, refer to Oracle WebLogic Server and Oracle Fusion Middleware documentation.

For Android notifications, retailers will need to set up the acquired API key as a generic credential in WebLogic. The GCM API key must be set in a map named "RetailAppsPushServices" under the key "gcmApiKey". Refer to Oracle WebLogic Server and Oracle Fusion Middleware documentation for instructions on creating and setting generic credentials

Lastly, if the server environment has proxy settings, the RetailAppsPushServices deployed on the weblogic server has to be modified. Retrieve the RetailAppsPushServices application enterprise archive (EAR) from the server, and modify the proxy.properties within. Once completed, the modified EAR file has to be redeployed to the server.

Managing Application Navigator

Retail Applications provide an ability to switch between applications using the Application Navigator facility. These applications are configured using the Manage Application Navigator screens on the Retail Application Administration Console (RAAC). For more details on Application Navigator in RAAC, see the applicable chapter in the latest *Merchandising Implementation Guide*.

Managing Functional Security

This chapter discusses the Functional Security for Retail Applications and the components used to implement it. Functional security is based on OPSS. For more details on OPSS, see the Oracle Fusion Middleware Application Security Guide.

Introduction to Retail Roles

Users are not assigned to permissions directly; rather access is assigned to roles. Roles group particular permissions required to accomplish a task; instead of assigning individual permissions, roles match users with the permissions required to complete their particular task.

There are two main types of roles, enterprise and application.

The Identity Store contains enterprise roles that are available across applications. These are created as groups in LDAP, making them available across applications.

Applicable Retail Applications security provides four types of roles: abstract, job, duty, and privilege.

Applicable Retail Applications will record job, abstract roles as enterprise roles and duty, privilege roles as application roles.

Security Policy Stripe

Application roles are stored in the application-specific policy store. These roles and role mappings are described in the jazn-data.xml file under the policy stripe 'RMS'.

Abstract Roles

Abstract roles are associated with a user, irrespective of their job or job function. These roles are not associated with a job or duty. These roles are normally assigned by the system (based on user attributes), but can be provisioned to a user on request.

Naming Convention: All the Retail Abstract role names end with' _ABSTRACT'

Example: APPLICATION_ADMIN_ABSTRACT

Job Roles

Job roles are associated with the job of a user. A user with this job can have many job functions or job duties.

Note: These roles are called Job roles because the role names closely map to the jobs commonly found in most organizations.

Naming Convention: All the Retail Job role names end with' _JOB'

Example: ALLOCATOR_JOB.

Duty Roles

Job duties are tasks one must do on a job. A person is hired into a job role. These are the responsibilities one has for a job.

Duty roles are roles that are associated with a specific duty or a logical grouping of tasks. Generally, the list of duties for a job is a good indicator of what duty roles should be defined.

Duty roles should:

- Read as a job description at a job posting site
- Duties that we create should be self-contained and pluggable into any existing or new job or abstract role

Naming Convention: All the Retail duty role names end with' _DUTY'

Example: ALC_ALLOC_POLICY_MAINTENANCE_MANAGEMENT_DUTY

Privilege Roles

Privilege is the logical collection of permissions. A privilege can be associated with any number of UI components. Privileges are expressed as application roles.

Naming Convention: All the Retail Privilege role names end with' _PRIV'

For example: SEARCH_PURCHASE_ORDERS_PRIV

Privilege roles carry security grants.

Example:

```
<grant>
  <grantee>
    <principals>
      <principal>
        <class>oracle.security.jps.service.policystore.
          ApplicationRole</class>
        <name>SEARCH_PURCHASE_ORDERS_PRIV </name>
```

```

        </principal>
    </principals>
</grantee>
<permissions>
  <permission>
<class>oracle.adf.controller.security.TaskFlowPermission</class>
<name>//WEB-
INF/oracle/retail/apps/rms/procurement/view/ordersetup/search/flow/SearchOrderFlow
.xml#SearchOrderFlow</name>
  <actions>view</actions>
</permission>
</permissions>
</grant>

```

Retail Role Hierarchy

Retail role hierarchies are structured to reflect the retail business process model.

Job roles inherit duty roles. For example, the Buyer Job role inherits the RMS_PURCHASE_ORDER_INQUIRY_DUTY roles.

```

<app-role>
  <name>RMS_PURCHASE_ORDER_INQUIRY_DUTY</name>
  <class>oracle.security.jps.service.policystore.ApplicationRole</class>
  <members>
    <member>
      <class>oracle.security.jps.internal.core.principals.
        JpsXmlEnterpriseRoleImpl</class>
      <name>BUYER_JOB</name>
    </member>
  </members>
</app-role>

```

Duty roles inherit Privilege roles. Duty roles can inherit one or more other Duty roles.

Example: RMS_PURCHASE_ORDER_MANAGEMENT_DUTY inherits RMS_PURCHASE_ORDER_INQUIRY_DUTY role.

```

<app-role>
  <name> RMS_PURCHASE_ORDER_INQUIRY_DUTY</name>
  <class>oracle.security.jps.service.policystore.ApplicationRole</class>
  <members>
    <member>
      <class>oracle.security.jps.internal.core.principals.
        JpsXmlEnterpriseRoleImpl</class>
      <name>BUYER_JOB</name>
    </member>
    <member>
      <class>oracle.security.jps.service.policystore.ApplicationRole</class>
      <name> RMS_PURCHASE_ORDER_MANAGEMENT_DUTY</name>
    </member>
  </members>
</app-role>

```

For example: RMS_PURCHASE_ORDER_MANAGEMENT_DUTY role inherits the RMS_PURCHASE_ORDER_INQUIRY_DUTY

Default Security Reference Implementation

Retail Applications ship with a default security reference implementations. The source of truth for default reference implementation is jazn-data.xml.

Privileges

Privilege	Privilege Description
View Application Administration Priv	This privilege provides view access to the System Options screen, the Download Blank Template and the Foundation Data Download screens. Within the Foundation Data Download screen, this privilege provides access to download the following to a spreadsheet: Codes and Descriptions, Custom Flex Attribute Foundation, Languages, Error Messages, Security Users, Security Groups, Associate Users to Groups, Associate Locations to Groups, Filter Groups, and/or Role and Privileges.
Maintain Application Administration Priv	This privilege provides create/edit access to the System Options, the Foundation Data Download and the Foundation Data Upload screens. Foundation Data that can be created and maintained via spreadsheets with this privilege includes Codes and Descriptions, Custom Flex Attribute Foundation, Languages, Error Messages, Security Users, Security Groups, Associate Users to Groups, Associate Locations to Groups, Filter Groups, and Role and Privileges.
View Report Options Priv	A privilege for viewing Report Options and department level Report Option Exceptions.
Maintain Report Options Priv	A privilege for creating and maintaining Report Options and department level Report Option Exceptions.
View Translations Priv	This privilege provides view access to the centralized data 'Translations' screen.
Maintain Translations Priv	This privilege provides edit access to the centralized data 'Translations' screen.
View Asynchronous Jobs Priv	This privilege provides view access to the Asynchronous Job Log and the Data Loading Status screens.
Maintain Asynchronous Jobs Priv	This privilege provides edit access to the Data Loading Status, the Cost Event Search, and the Cost Event screens.
Maintain Custom Flex Attributes Priv	A privilege for creating and maintaining Custom Flex Attribute definitions.
View Custom Flex Attributes Priv	A privilege for viewing Custom Flex Attribute definitions.
View Foundation Data via Spreadsheet Priv	Gives users ability to download supporting data for Inventory, Price and Cost, Items, and Import.
Maintain Foundation Data via Spreadsheet Priv	Gives users ability to upload supporting data for Inventory, Price and Cost, Items, and Import.

Privilege	Privilege Description
View Competitive Shop Priv	A privilege for viewing Competitive Shop information including shop requests competitive price history.
Maintain Competitive Shop Priv	A privilege for creating and editing competitive shop requests.
Approve Contracts Priv	A privilege for approving contracts
Search Contracts Priv	A privilege for searching for contracts
View Contracts Priv	A privilege for viewing contracts and contract cost history.
Maintain Contracts Priv	A privilege for creating and maintaining contracts. Including creating orders from contracts.
Submit Contracts Priv	A privilege for submitting contracts
Approve Cost Changes Priv	A privilege for approving cost changes
Search Cost Changes Priv	A privilege for searching for cost changes
View Cost Changes Priv	A privilege for viewing cost changes
Maintain Cost Changes Priv	A privilege for creating and maintaining cost changes.
Submit Cost Changes Priv	A privilege for submitting cost changes
View Cost Zones Priv	A privilege for viewing Cost Zone Groups and Cost Zones
Maintain Cost Zones Priv	A privilege for creating and maintaining Cost Zone Groups and Cost Zones
View Buyer Dashboard Priv	This privilege gives the user access to the Buyer Dashboard.
View Finance Analyst Dashboard Priv	This privilege gives the user access to the Finance Analyst Dashboard.
View Inventory Analyst Dashboard Priv	This privilege gives the user access to the Inventory Analyst Dashboard.
View Inventory Control Dashboard Priv	This privilege gives the user access to the Inventory Control Dashboard.
View Data Steward Dashboard Priv	This privilege gives the user access to the Data Steward Dashboard.
Approve Deals Priv	A privilege for approving deals
Search Deals Priv	A privilege for searching for deals
View Deals Priv	A privilege for viewing deals
Maintain Deals Priv	A privilege for creating and maintaining deals
Submit Deals Priv	A privilege for submitting deals
View Expense Profiles Priv	A privilege for viewing Expense Profiles for Suppliers, Partners, and Countries.

Privilege	Privilege Description
Maintain Expense Profiles Priv	A privilege for creating and editing Expense Profiles for Suppliers, Partners, and Countries.
View Up Charges Priv	A privilege for viewing the association of Up Charges with Departments.
Maintain Up Charges Priv	A privilege for creating and editing the association of Up Charges with Departments.
Maintain Average Cost Priv	A privilege for creating average cost adjustments and viewing transaction data.
View Budgets Priv	A privilege for the ability to download Monthly Budget and Open to Buy information as well as viewing the Open to Buy dashboard report.
Maintain Budgets Priv	Gives users ability to upload Monthly Budget and Open to Buy information.
View Finance Admin Priv	Gives users ability to download supporting data for GL Cross Reference, VAT, Terms, Currency Exchange Type Mapping, Currency, and Non-Merchandise Codes.
Maintain Finance Admin Priv	Gives users ability to upload supporting data for GL Cross Reference, VAT, Terms, Currency Exchange Type Mapping, Currency, and Non-Merchandise Codes.
View Financial Control Priv	Gives users ability to download Half Data Budget.
Maintain Financial Control Priv	Gives users ability to upload Half Data Budget.
View GL Drill Back Priv	A privilege for viewing Fixed Deal Transaction Data and GL drill back information.
View Transaction Data Priv	A privilege for viewing Transaction Data and Late Posted Transactions dashboard report.
View Franchise Cost Priv	A privilege for viewing franchise cost templates and cost relationships.
Maintain Franchise Cost Priv	A privilege for creating and maintaining franchise cost templates and cost relationships.
View Franchise Foundation Priv	Gives users ability to download supporting data for Deal Pass Through and Franchise Customers.
Maintain Franchise Foundation Priv	Gives users ability to upload supporting data for Deal Pass Through and Franchise Customers.
Approve Franchise Orders and Returns Priv	A privilege for approving franchise orders and returns.
View Franchise Orders and Returns Priv	A privilege for viewing franchise orders and returns.
Maintain Franchise Orders and Returns Priv	A privilege for creating, maintaining, and submitting franchise orders and returns.

Privilege	Privilege Description
Search Actual Landed Cost Priv	A privilege for searching for actual landed cost.
View Actual Landed Cost Priv	A privilege for viewing actual landed cost.
Finalize Actual Landed Cost Priv	A privilege for finalizing actual landed cost.
Confirm Customs Entries Priv	A privilege for confirming customs entries.
Search Customs Entries Priv	A privilege for searching for customs entries.
View Customs Entries Priv	A privilege for viewing customs entries.
Maintain Customs Entries Priv	A privilege for creating and maintaining customs entries.
Submit Customs Entries Priv	A privilege for submitting/sending customs entries.
View Import Foundation Priv	Gives users ability to download supporting data for the import process, including Freight Sizes, Freight Types, HTS Headings, HTS Heading Restraints, Quota Categories, HTS Import Country Setup, Tariff Treatments, Origin Country Tariff Treatments, OGAs, SCACs and timelines. Also included is the ability to download HTS definitions which include HTS, HTS Tariff Treatments, HTS Tariff Treatment Zones, HTS Fees, HTS Fee Zones, HTS Taxes, HTS Tax Zones, HTS Anti-Dumping, HTS Countervailing Duty, HTS Reference and HTS Other Government Agencies.
Maintain Import Foundation Priv	Gives users ability to download and upload supporting data for the import process, including Freight Sizes, Freight Types, HTS Headings, HTS Heading Restraints, Quota Categories, HTS Import Country Setup, Tariff Treatments, Origin Country Tariff Treatments, OGAs, SCACs and timelines. Also included is the ability to download HTS definitions which include HTS, HTS Tariff Treatments, HTS Tariff Treatment Zones, HTS Fees, HTS Fee Zones, HTS Taxes, HTS Tax Zones, HTS Anti-Dumping, HTS Countervailing Duty, HTS Reference and HTS Other Government Agencies.
View Import Items Priv	A privilege for viewing import items, including, Item Suppliers, Item Supplier Countries, Item HTS, Item Import Attributes and Item Eligible Tariff Treatments.
Maintain Import Items Priv	A privilege for maintaining import items, including, Item Suppliers, Item Supplier Countries, Item HTS, Item Import Attributes, Item Eligible Tariff Treatments.
Approve Letters of Credit Priv	A privilege for approving letters of credit.
Confirm Letters of Credit Priv	A privilege for confirming letters of credit.

Privilege	Privilege Description
Search Letters of Credit Priv	A privilege for searching for letters of credit.
View Letters of Credit Priv	A privilege for viewing letters of credit.
Maintain Letters of Credit Priv	A privilege for creating and maintaining letters of credit.
Submit Letters of Credit Priv	A privilege for submitting letters of credit.
Approve Obligations Priv	A privilege for approving obligations.
Search Obligations Priv	A privilege for searching for obligations.
View Obligations Priv	A privilege for viewing obligations.
Maintain Obligations Priv	A privilege for creating and maintaining obligations.
Search Transportation Priv	A privilege for searching for transportation records.
View Transportation Priv	A privilege for viewing transportation records.
Finalize Transportation Priv	A privilege for finalizing transportation records, generating customs entries.
Maintain Transportation Priv	A privilege for creating and maintaining transportation records.
Maintain Inventory Adjustments Priv	A privilege for creating inventory adjustments by item or by location, breaking sellable packs, and creating and maintaining product transformations.
View Item Inventory and Sales Priv	A privilege for view item inventory and sales information via the Item Location Inventory, Store Sales/Issues by Location, and Sellable Pack Inquiry screens, as well as the Daily Sales, Inventory Variance to Forecast, Inventory by Store, and Inventory by Warehouse, Unexpected Inventory, and Negative Inventory dashboard reports.
Approve Intercompany Transfers Priv	A privilege for approving intercompany transfers. It is expected that a user with this privilege will also have the ability to view and maintain both intercompany and non-intercompany transfers.
View Intercompany Transfers Priv	A privilege for viewing intercompany transfers. It is expected that a user with this privilege will also have the View Transfers Priv.
Maintain Intercompany Transfers Priv	A privilege for creating and maintaining intercompany transfers. It is expected that a user with this privilege will also have the ability to view, create and maintain non-intercompany transfers.

Privilege	Privilege Description
Submit Intercompany Transfers Priv	A privilege for submitting intercompany transfers. It is expected that a user with this privilege will also have the ability to view, create and maintain non-intercompany transfers.
Approve MRTs Priv	A privilege for approving mass return transfers (MRTs).
Search MRTs Priv	A privilege for searching for mass return transfers (MRTs).
View MRTs Priv	A privilege for viewing mass return transfers (MRTs).
Maintain MRTs Priv	A privilege for creating and maintaining mass return transfers (MRTs).
Submit MRTs Priv	A privilege for submitting mass return transfers (MRTs).
Approve RTVs Priv	A privilege for approving returns to vendors (RTVs).
Search RTVs Priv	A privilege for searching for returns to vendors (RTVs).
View RTVs Priv	A privilege for viewing returns to vendors (RTVs) and Overdue RTVs dashboard report.
Maintain RTVs Priv	A privilege for creating and maintaining returns to vendors (RTVs).
View Shipments and Receipts Priv	A privilege for viewing shipment and receipt information via the Transportation Shipments, BOL Search, BOL, Shipment Search, Shipment, and Appointments screens, as well as the Early/Late Shipments and Stock Orders Pending Close dashboard reports.
Maintain Shipments and Receipts Priv	<p>A privilege for creating and maintaining shipment and receipt information via the BOL Search, BOL, Receive by Carton, Receive by Item, Adjust Receipts, Shipment Search, Shipment, Order Shipment, MRT, Transfer, RTV, Stock Order Reconciliation, screens, as well as the Stock Orders Pending Close dashboard report.</p> <p>A user with this privilege will be able to close mass return transfers (MRTs), assuming they have the Maintain MRTs Priv as well. A user with this privilege will be able to ship returns to vendors (RTVs), assuming they have the Maintain RTVs Priv as well. A user with this privilege will be able to ship and close transfers, assuming they have the Maintain Transfers Priv.</p>

Privilege	Privilege Description
Approve Transfers Priv	A privilege for approving transfers via the Transfer screen or the Transfers Pending Approval dashboard report.
View Transfers Priv	A privilege for viewing transfers and appointments via the Transfer Search and Appointments screens, as well as the Overdue Transfers and Transfers Pending Approval dashboard reports.
Search Transfers Priv	A privilege for searching for transfers via the Transfer Search screen.
Maintain Transfers Priv	A privilege for creating and maintaining transfers via the Transfer Search and Transfer screens, as well as the Unexpected Inventory, Overdue Transfers and Transfers Pending Approval dashboard reports.
Submit Transfers Priv	A privilege for submitting transfers via the Transfer screen.
Maintain Transfers with Finishing Priv	This priv allows users to create multi-leg transfers with finishers. Without this priv users can only create single leg transfers.
View Diffs Priv	A privilege for viewing Diffs including Diff Groups, Diff Ranges, and Pack Templates
Maintain Diffs Priv	A privilege for creating and maintaining Diffs including Diff Groups, Diff Ranges, and Pack Templates
Use Diffs Priv	The privilege provides access to the Diff Matrix, Diff Distribution, Re-distribution by Diff, and the Order, Transfer and Contract Parent/Diff Summary contextual reports. This privilege also provides access to the Differentiators container in the Item screen.
Search Diff Ratios Priv	A privilege for searching for Diff Ratios.
View Diff Ratios Priv	A privilege for viewing Diff Ratios.
Maintain Diff Ratios Priv	A privilege for creating and maintaining Diff Ratios.
Maintain/View Grocery Attributes Priv	A privilege for access to grocery attributes for grocery items.
Approve Items Priv	A privilege for approving items.
Search Items Priv	A privilege for searching for Items.

Privilege	Privilege Description
View Items Priv	A privilege for viewing Items, including Item Zone Retail, Item Seasons, Item Attributes, Item Timelines, Item UDAs, Item VAT, Images, Item Differentiators, Item Children, Mass Change Item Attributes, Simple Pack Setup, Pack Detail, Packs by Component, Item Approval Errors, Ticketing, Orders by Item, Item Transformation, Item Countries, Item Expenses and viewing the Incomplete Items dashboard report.
Search Item Lists Priv	A privilege for searching for item lists
View Item Lists Priv	A privilege for viewing item lists
Maintain Item Lists Priv	A privilege for creating and maintaining item lists
View Item Locations Priv	A privilege for viewing item location information including Item Up Charges and Scheduled Item Maintenance, also to view WAC Variance dashboard report.
Maintain Item Locations Priv	A privilege for maintaining item location information including Item Up Charges, Scheduled Item Maintenance, using Item Lists, and Mass Change Item/Location Attributes
Maintain Items Priv	A privilege for creating and editing Items, including Item Zone Retail, Item Seasons, Item Attributes, Item Timelines, Item UDAs, Item VAT, Images, Item Differentiators, Item Children, Mass Change Item Attributes, Simple Pack Setup, Pack Details, Item Approval Errors, Ticketing, Orders by Item, Item Transformation, Item Countries, Item Expenses and putting items back to Worksheet status.
Submit Items Priv	A privilege for submitting items.
View Item Suppliers Priv	A privilege for viewing item supplier relationships and sourcing information including Item Supplier Country of Manufacture, Item Supplier Country List, Item Supplier UOM, Item Supplier Country of Sourcing, Item Supplier Country Dimensions, Item Bracket Costing, Item Expenses, Item Supplier Country Locations and the Lead Time Days dashboard report
Maintain Item Suppliers Priv	A privilege for viewing item supplier relationships and sourcing information including Item Supplier Country of Manufacture, Item Supplier Country List, Item Supplier UOM, Item Supplier Country of Sourcing, Item Supplier Country Dimensions, Item Bracket Costing, Item Expenses, and Item Supplier Country Locations.

Privilege	Privilege Description
View Price Change History Priv	A privilege for viewing price change history information via the Price Change History screen.
Maintain Ticket Requests Priv	A privilege for creating ticket printing requests via the Ticket Request screen.
View Stores Priv	A privilege for viewing Stores, including Addresses, CFAS, Store Location Traits and Walk Through Stores.
View Warehouses Priv	A privilege for viewing Physical and Virtual Warehouses, including Warehouse Department Parameters, Addresses and CFAS.
Search Location Lists Priv	A privilege for searching for Location Lists.
View Location Lists Priv	A privilege for viewing Location Lists, including Location List Criteria and Location List Details.
Maintain Location Lists Priv	A privilege for creating and editing Location Lists, including Location List Criteria and Location List Details.
Maintain Stores Priv	A privilege for creating and editing Stores, including Addresses, CFAS, Store Location Traits and Walk Through Stores.
Maintain Warehouses Priv	A privilege for creating and editing Physical and Virtual Warehouses, including Warehouse Department Parameters, Addresses and CFAS.
View Supply Chain Network Priv	A privilege for viewing Supply Chain Network information including Priority Groups, Location Activity Schedules, Warehouse Department Parameters, Transit Times, and Supplier and Location Delivery Schedules.
Maintain Supply Chain Network Priv	A privilege for creating and editing Supply Chain Network information including Priority Groups, Location Activity Schedules, Warehouse Department Parameters, Transit Times, and Supplier and Location Delivery Schedules.
Search Merchandise Hierarchy Priv	A privilege for searching for Pending Merchandise Hierarchy changes via the Pending Merchandise Hierarchy Search screen.
View Classes Priv	A privilege for viewing Classes, including CFAS.
View Departments Priv	A privilege for viewing Departments, including CFAS.
View Divisions Priv	A privilege for viewing Divisions.
View Groups Priv	A privilege for viewing Groups.

Privilege	Privilege Description
View Merchandise Hierarchy Defaults Priv	A privilege for viewing Merchandise Hierarchy Defaults and UDA Defaults for departments, classes, and subclasses.
View Merchandise Hierarchy Priv	The View Merchandise Hierarchy Priv provides access to the Pending Merchandise Hierarchy Changes screen to view details of pending changes, and the Merchandise Hierarchy screen to view all levels of the merchandise hierarchy. In order to access further details of entities at each level of the hierarchy, the view privileges specific to each entity are used.
View Subclasses Priv	A privilege for viewing Subclasses, including CFAS.
Maintain Classes Priv	A privilege for creating and editing Classes, including CFAS.
Maintain Departments Priv	A privilege for creating and editing Departments, including CFAS. Also provides access to update budget information from the Cumulative Markon Variance dashboard report.
Maintain Divisions Priv	A privilege for creating and editing Divisions.
Maintain Groups Priv	A privilege for creating and editing Groups.
Maintain Merchandise Hierarchy Defaults Priv	A privilege for creating and editing Merchandise Hierarchy Defaults and UDA Defaults for departments, classes, and subclasses.
Maintain Merchandise Hierarchy Priv	A privilege for editing Pending Merchandise Hierarchy changes.
Maintain Reclassifications Priv	A privilege for maintaining merchandise hierarchy relationships.
Maintain Subclasses Priv	A privilege for creating and editing Subclasses, including CFAS.
View Areas Priv	A privilege for viewing Areas.
View Chains Priv	A privilege for viewing Chains.
View Districts Priv	A privilege for viewing Districts.
View Organizational Hierarchy Priv	The View Organization Hierarchy Priv provides access to the Organizational Hierarchy screen to view all levels of the Organizational Hierarchy. In order to access further details of entities at each level of the hierarchy, the view privileges specific to each entity are used.
View Regions Priv	A privilege for viewing Regions.
Maintain Areas Priv	A privilege for creating and editing Areas.
Maintain Chains Priv	A privilege for creating and editing Chains.
Maintain Districts Priv	A privilege for creating and editing Districts.

Privilege	Privilege Description
Maintain Regions Priv	A privilege for creating and editing Regions.
Maintain DSD Orders Priv	A privilege for creating Direct to Store Delivery (DSD) orders.
Maintain Pre-Issued Order Numbers Priv	A privilege for creating pre-issued order numbers and creating order using pre-issued numbers.
Approve Purchase Orders Priv	A privilege for approving purchase orders and viewing the Orders Pending Approval dashboard report.
Search Purchase Orders Priv	A privilege for searching for purchase orders.
View Purchase Orders Priv	A privilege for viewing purchase orders and viewing the Past Not After Date, Orders to Close, Never Approved, Once Approved, Missing Orders Data, Items Missing Data and Open Orders dashboard reports.
Close Purchase Orders Priv	A privilege for closing purchase orders.
Maintain Purchase Orders Priv	A privilege for creating and maintaining purchase orders
Submit Purchase Orders Priv	A privilege for submitting purchase orders.
Maintain Receipt Adjustments by Cost Priv	A privilege for adjusting receipts by cost and viewing transaction data.
View Coupons Priv	A privilege for viewing coupons.
View Product Restrictions Priv	A privilege for viewing product restrictions.
Maintain Coupons Priv	A privilege for creating and maintaining coupons.
Maintain Product Restrictions Priv	A privilege for creating and maintaining production restrictions.
View Replenishment Attributes Priv	A privilege for viewing replenishment attributes.
Maintain Replenishment Attributes Priv	A privilege for creating and maintaining replenishment attributes.
View Required Documents Priv	A privilege for viewing the association of Required Documents with Suppliers, Partners, HTS Headings/Chapters, Countries, Letters of Credit, Letter of Credit Amendments, Items, Orders, and Order Items.
Maintain Required Documents Priv	A privilege for creating and editing the association of Required Documents with Suppliers, Partners, HTS Headings/Chapters, Countries, Letters of Credit, Letter of Credit Amendments, Items, Orders, and Order Items.
View Stock Count Units Priv	A privilege for viewing stock count results and viewing the Stock Count Unit Variance dashboard report.

Privilege	Privilege Description
View Stock Counts Priv	A privilege for viewing stock count requests and viewing the Stock Counts Missing dashboard report.
Maintain Stock Count Units Priv	A privilege for maintaining stock count results, unit variance and adjustments, and virtual warehouse distribution.
Maintain Stock Counts Priv	A privilege for creating and maintaining stock count requests.
Maintain Stock Count Values Priv	A privilege for maintaining stock count value variance and adjustments.
View Stock Ledger Priv	A privilege for viewing the stock ledger and viewing the Cumulative Markon Variance and Shrinkage Variance dashboard reports.
Maintain Stock Ledger Priv	A privilege for maintaining the stock ledger.
Activate Partners Priv	A privilege for activating a Partner.
Deactivate Partners Priv	A privilege for deactivating a Partner.
Search Partners Priv	A privilege for searching for Partners.
View Partners Priv	A privilege for viewing Partners including Partner Address and Invoice Matching Attributes.
Maintain Partners Priv	A privilege for creating and editing Partners including Partner Address, Invoice Matching Attributes and CFAS via the Address screen.
Activate Suppliers Priv	A privilege for activating a supplier.
Deactivate Suppliers Priv	A privilege for deactivating a supplier.
Search Suppliers Priv	A privilege for searching for Suppliers.
View Suppliers Priv	A privilege for viewing Suppliers, including Supplier Address, EDI, Returns, Traits, Import Attributes, Invoice Matching Attributes, Importer/Exporter, Routing Location, CFAS, and Org Unit information.
View Supplier Inventory Management Priv	A privilege for creating and editing Supplier Inventory Management information including Supplier Brackets.
Maintain Supplier Inventory Management Priv	A privilege for creating and editing Supplier Inventory Management information including Supplier Brackets.
Maintain Suppliers Priv	A privilege for creating and editing Suppliers, including Supplier Address, EDI, Returns, Traits, Import Attributes, Invoice Matching Attributes, Importer/Exporter, Routing Location, CFAS, and Org Unit information.
View VAT Priv	A privilege for viewing the association of VAT information with Departments.

Privilege	Privilege Description
Maintain VAT Priv	A privilege for creating and editing the association of VAT information with Departments.

Duties

Duty	Duty Description	Privileges
Application Admin Inquiry Duty	A duty for viewing application administrative information including system options.	View Application Administration Priv
Application Admin Management Duty	A duty for maintaining application administration information, including system options. This duty is an extension of the Application Admin Inquiry Duty.	All privileges in the Application Admin Inquiry Duty Maintain Application Administration Priv
Asynchronous Job Inquiry Duty	A duty for viewing the asynchronous job log.	View Asynchronous Jobs Priv
Asynchronous Job Management Duty	A duty for managing asynchronous jobs. This duty is an extension of the Application Admin Inquiry Duty.	All privileges included in the Asynchronous Job Inquiry Duty. Maintain Asynchronous Jobs Priv
Data Security Inquiry Duty	A duty for viewing data security definitions.	View Data Security Priv
Data Security Management Duty	A duty for maintaining data security definitions. This duty is an extension of the Data Security Inquiry Duty.	All privileges included in the Data Security Inquiry Duty. Maintain Data Security Priv
Foundation Data via Spreadsheet Inquiry Duty	A duty for viewing supporting data for Inventory, Price and Cost, Items, and Import, via downloaded spreadsheets.	View Foundation Data via Spreadsheet Priv
Foundation Data via Spreadsheet Management Duty	A duty for maintaining supporting data for Inventory, Price and Cost, Items, and Import, via spreadsheet downloads and uploads. This duty is an extension of the Foundation Data via Spreadsheet Inquiry Duty.	All privileges included in the Foundation Data Admin Inquiry Duty. Maintain Foundation Data via Spreadsheet Priv
Custom Flex Attributes Inquiry Duty	A duty for viewing custom flex attribute (CFAS) definitions	View Custom Flex Attributes Priv
Custom Flex Attributes Management Duty	A duty for maintaining custom flex attribute (CFAS) definitions. This duty is an extension of the Custom Flex Attributes Inquiry Duty.	All privileges included in the Custom Flex Attributes Inquiry Duty. Maintain Custom Flex Attributes Priv

Duty	Duty Description	Privileges
Report Options Inquiry Duty	A duty for viewing dashboard and contextual BI report configuration options.	View Report Options Priv
Report Options Management Duty	A duty for maintaining dashboard and contextual BI report configuration options. This duty is an extension of the Report Options Inquiry Duty.	All privileges included in the Report Options Inquiry Duty Maintain Report Options Priv
Centralized Data Translation Inquiry Duty	A duty for viewing data translations via a centralized online screen.	View Translations Priv
Centralized Data Translation Management Duty	A duty for maintaining data translations via a centralized online screen. This duty is an extension of the Centralized Data Translation Inquiry Duty.	All privileges included in the Centralized Data Translation Inquiry Duty Maintain Translations Priv
Competitor Inquiry Duty	A duty for viewing competitive shop requests and competitive price history.	View Competitive Shop Priv
Competitor Management Duty	A duty for maintaining competitive shop requests. This duty is an extension of the Competitive Shop Inquiry Duty.	All privileges included in the Competitor Inquiry Duty. Maintain Competitive Shop Priv
Contract Inquiry Duty	A duty for searching for and viewing Contracts.	Search Contracts Priv View Contracts Priv
Contract Management Duty	A duty for maintaining and submitting contracts. This duty is an extension of the Contract Inquiry Duty.	All privileges included in the Contract Inquiry Duty. Maintain Contracts Priv Submit Contracts Priv
Contract Approval Duty	A duty for approving contracts. This duty is an extension of the Contract Management Duty.	All privileges included in the Contract Management Duty. Approve Contracts Priv
Cost Change Inquiry Duty	A duty for viewing item cost changes.	View Cost Changes Priv
Cost Change Management Duty	A duty for maintaining and submitting item cost changes. This duty is an extension of the Cost Change Inquiry Duty.	All privileges included in the Cost Change Inquiry Duty. Maintain Cost Changes Priv Submit Cost Changes Priv
Cost Change Approval Duty	A duty for approving item cost changes. This duty is an extension of the Cost Change Management Duty.	All privileges included in the Cost Change Management Duty. Approve Cost Changes Priv
Cost Zone Inquiry Duty	A duty for viewing cost zones.	View Cost Zones Priv

Duty	Duty Description	Privileges
Cost Zone Management Duty	A duty for maintaining cost zones. This duty is an extension of the Cost Zone Inquiry Duty.	All privileges included in the Cost Zone Inquiry Duty. Maintain Cost Zones Priv
Deal Inquiry Duty	A duty for searching for and viewing deals.	Search Deals Priv View Deals Priv
Deal Management Duty	A duty for maintaining and submitting deals. This duty is an extension of the Deal Inquiry Duty.	All privileges included in the Deal Inquiry Duty. Maintain Deals Priv Submit Deals Priv
Deal Approval Duty	A duty for approving deals. This duty is an extension of the Deal Management Duty.	All privileges included in the Deal Management Duty. Approve Deals Priv
Expense Inquiry Duty	A duty for viewing expense profiles used to build up an item's ELC.	View Expense Profiles Priv
Expense Management Duty	A duty for maintaining expense profiles used to build up an item's ELC. This duty is an extension of the Expense Inquiry Duty.	All privileges included in the Expense Inquiry Duty. Maintain Expense Profiles Priv
Up Charge Inquiry Duty	A duty for viewing department up charges.	View Up Charges Priv
Up Charge Management Duty	A duty for maintaining department up charges. This duty is an extension of the Up Charge Inquiry Duty.	All privileges included in the Up Charge Inquiry Duty. Maintain Up Charges Priv
Average Cost Management Duty	A duty for maintaining average cost values.	Maintain Average Cost Priv
GL Drill Back Inquiry Duty	A duty for viewing general ledger drill back.	View GL Drill Back Priv
Tran Data Inquiry Duty	A duty for viewing transaction data.	View Transaction Data Priv
Franchise Cost Inquiry Duty	A duty for viewing franchise cost information.	View Franchise Cost Priv
Franchise Cost Management Duty	A duty for maintaining franchise cost. This duty is an extension of the Franchise Cost Inquiry Duty.	All privileges included in the Franchise Cost Inquiry Duty. Maintain Franchise Cost Priv
Franchise Orders and Returns Inquiry Duty	A duty for viewing franchise orders and returns.	View Franchise Orders and Returns Priv
Franchise Orders and Returns Management Duty	A duty for maintaining franchise orders and returns. This duty is an extension of the Franchise Orders and Returns Inquiry Duty.	All privileges included in the Franchise Orders and Returns Inquiry Duty. Maintain Franchise Orders and Returns Priv

Duty	Duty Description	Privileges
Franchise Orders and Returns Approval Duty	A duty for approving franchise orders and returns. This is an extension of the Franchise Orders and Returns Management Duty.	All privileges included in the Franchise Orders and Returns Management Duty. Approve Franchise Orders and Returns Priv
ALC Inquiry Duty	A duty for searching for and viewing actual landed cost (ALC).	Search Actual Landed Cost Priv View Actual Landed Cost Priv
ALC Management Duty	A duty for finalizing actual landed cost (ALC). This duty is an extension of the ALC Inquiry Duty.	All privileges included in the ALC Inquiry Duty. Finalize Actual Landed Cost Priv
Customs Entry Inquiry Duty	A duty for searching for and viewing customs entries.	Search Customs Entries Priv View Customs Entries Priv
Customs Entry Management Duty	A duty for maintaining and submitting customs entries. This is an extension of the Customs Entry Inquiry Duty.	All privileges included in the Customs Entry Inquiry Duty. Maintain Customs Entries Priv Submit Customs Entries Priv
Customs Entry Confirmation Duty	A duty for confirming customs entries. This is an extension of the Customs Entry Management Duty.	All privileges included in the Customs Entry Management Duty. Confirm Customs Entries Priv
Import Item Inquiry Duty	A duty for viewing item data related to imports.	View Import Items Priv
Import Item Management Duty	A duty for maintaining item data related to imports. This is an extension of the Import Item Inquiry Duty.	All privileges included in the Import Item Management Duty. Maintain Import Items Priv
Letter of Credit Inquiry Duty	A duty for searching for and viewing letters of credit.	Search Letters of Credit Priv View Letters of Credit Priv
Letter of Credit Management Duty	A duty for maintaining and submitting letters of credit. This is an extension of the Letter of Credit Inquiry Duty.	All privileges included in the Letter of Credit Inquiry Duty. Maintain Letters of Credit Priv Submit Letters of Credit Priv
Letter of Credit Approval Duty	A duty for approving and confirming letters of credit. This is an extension of the Letter of Credit Management Duty.	All privileges included in the Letter of Credit Management Duty. Approve Letters of Credit Priv Confirm Letters of Credit Priv
Obligation Inquiry Duty	A duty for searching for and viewing obligations.	Search Obligations Priv View Obligations Priv
Obligation Management Duty	A duty for maintaining obligations. This is an extension of the Obligation Inquiry Duty.	All privileges included in the Obligation Inquiry Duty. Maintain Obligations Priv

Duty	Duty Description	Privileges
Obligation Approval Duty	A duty for approving obligations. This is an extension of the Obligation Management Duty.	All privileges included in the Obligation Management Duty. Approve Obligations Priv
Transportation Inquiry Duty	A duty for searching for and viewing transportation information.	Search Transportation Priv View Transportation Priv
Transportation Management Duty	A duty for maintaining transportation information. This duty is an extension of the Transportation Inquiry Duty.	All privileges included in the Transportation Inquiry Duty. Maintain Transportation Priv Finalize Transportation Priv
Inventory Adjustment Management Duty	A duty for creating inventory adjustments.	Maintain Inventory Adjustments Priv
Inventory and Sales Inquiry Duty	A duty for viewing item related inventory and sales information.	View Item Inventory and Sales Priv
Transfer Inquiry Duty	A duty for searching for and viewing transfers.	Search Transfers Priv View Transfers Priv
Transfer Management Duty	A duty for maintaining and submitting transfers. This duty is an extension of the Transfer Inquiry Duty.	All privileges included in the Transfer Inquiry Duty. Maintain Transfers Priv Submit Transfers Priv
Transfer Approval Duty	A duty for approving transfers. This duty is an extension of the Transfer Management Duty.	All privileges included in the Transfer Management Duty. Approve Transfers Priv
Transfer with Finishing Management Duty	A duty for maintaining transfers with finishing. This duty is an extension of the Transfer Management Duty.	All privileges included in the Transfer Management Duty. Maintain Transfers with Finishing Priv
Intercompany Transfer Inquiry Duty	A duty for viewing intercompany transfers. This is an extension of the Transfer Inquiry Duty.	All privileges included in the Transfer Inquiry Duty. View Intercompany Transfers Priv
Intercompany Transfer Management Duty	A duty for maintaining and submitting intercompany transfers. This is an extension of the Transfer Management Duty and the Intercompany Transfer Inquiry Duty.	All privileges included in the Transfer Management Duty. All privileges included in the Intercompany Transfer Inquiry Duty. Maintain Intercompany Transfers Priv Submit Intercompany Transfers Priv
Intercompany Transfer Approval Duty	A duty for approving intercompany transfers. This is an extension of the Transfer Management Duty and the Intercompany Transfer Management Duty.	All privileges included in the Transfer Approval Duty. All privileges included in the Intercompany Transfer Management Duty. Approve Intercompany Transfers Priv

Duty	Duty Description	Privileges
MRT Inquiry Duty	A duty for searching for and viewing mass return transfers (MRTs).	Search MRTs Priv View MRTs Priv
MRT Management Duty	A duty for maintaining and submitting mass return transfers (MRTs). This duty is an extension of the MRT Inquiry Duty.	All privileges included in the MRT Inquiry Duty. Maintain MRTs Priv Submit MRTs Priv
MRT Approval Duty	A duty for the approval of mass return transfers (MRTs). This duty is an extension of the MRT Management Duty.	All privileges included in the MRT Management Duty. Approve MRTs Priv
RTV Inquiry Duty	A duty for searching for and viewing returns to vendors (RTVs).	Search RTVs Priv View RTVs Priv
RTV Management Duty	A duty for maintaining returns to vendors (RTVs). This duty is an extension of the RTV Inquiry Duty.	All privileges included in the RTV Inquiry Duty. Maintain RTVs Priv
RTV Approval Duty	A duty for approving returns to vendors (RTVs). This duty is an extension of the RTV Management Duty.	All privileges included in the RTV Management Duty. Approve RTVs Priv
Shipment and Receipts Inquiry Duty	A duty for viewing shipments and receipts.	View Shipments and Receipts Priv
Shipment and Receipts Management Duty	A duty for maintaining shipments and receipts. This is an extension of the Shipment and Receipts Inquiry Duty.	All privileges included in the Shipment and Receipts Inquiry Duty. Maintain Shipments and Receipts Priv
Diff Inquiry Duty	A duty for viewing differentiators.	View Diffs Priv
Diff Management Duty	A duty for maintaining differentiators. This duty is an extension of the Diff Inquiry Duty.	All privileges included in Diff Inquiry Duty. Maintain Diffs Priv
Diff Ratio Inquiry Duty	A duty for searching for and viewing differentiator ratios.	Search Diff Ratios Priv View Diff Ratios Priv
Diff Ratio Management Duty	A duty for maintaining differentiator ratios. This duty is an extension of the Diff Ratio Inquiry Duty.	All privileges included in the Diff Ratio Inquiry Duty. Maintain Diff Ratios Priv
Grocery Attribute Management Duty	A duty for viewing and maintaining grocery item related attributes.	Maintain/View Grocery Attributes Priv
Item Inquiry Duty	A duty for searching for and viewing items.	Search Items Priv View Items Priv

Duty	Duty Description	Privileges
Item Management Duty	A duty for maintaining and submitting items. This duty is an extension of the Item Inquiry Duty.	All privileges included in the Item Inquiry Duty. Maintain Items Priv Submit Items Priv
Item Approval Duty	A duty for approving items. This duty is an extension of the Item Management Duty.	All privileges included in the Item Management Duty. Approve Items Priv
Item List Inquiry Duty	A duty for searching for and viewing item lists.	Search Item Lists Priv View Item Lists Priv
Item List Management Duty	A duty for maintaining item lists. This duty is an extension of the Item List Inquiry Duty.	All privileges included in the Item List Inquiry Duty. Maintain Item Lists Priv
Item Supplier Inquiry Duty	A duty for viewing item supplier sourcing information. This is an extension of the Item Inquiry Duty.	All privileges included in the Item Inquiry Duty. View Item Suppliers Priv
Item Supplier Management Duty	A duty for maintaining item supplier sourcing information. This duty is an extension of the Item Supplier Inquiry Duty and the Item Inquiry Duty.	All privileges included in the Item Supplier Inquiry Duty. Maintain Item Suppliers Priv
Item Location Inquiry Duty	A duty for viewing item location information. This is an extension of the Item Inquiry Duty.	All privileges included in the Item Inquiry Duty. View Item Locations Priv
Item Location Management Duty	A duty for maintaining item location information. This duty is an extension of the Item Location Inquiry Duty and the Item Inquiry Duty.	All privileges included in the Item Location Inquiry Duty. Maintain Item Locations Priv
Price Change Inquiry Duty	A duty for viewing price change history.	View Price Change History Priv
Ticket Management Duty	A duty for maintaining ticket requests.	Maintain Ticket Requests Priv
Location Inquiry Duty	A duty for searching for and viewing stores and warehouses.	View Organizational Hierarchy Priv View Stores Priv View Warehouses Priv
Location Management Duty	A duty for maintaining stores and warehouses. This duty is an extension of the Location Inquiry and Organizational Hierarchy Inquiry Duties.	All privileges included in the Location Inquiry Duty. Maintain Stores Priv Maintain Warehouses Priv
Location List Inquiry Duty	A duty for searching for and viewing location lists.	Search Location Lists Priv View Location Lists Priv

Duty	Duty Description	Privileges
Location List Management Duty	A duty for maintaining location lists. This duty is an extension of the Location List Inquiry Duty.	All privileges included in the Location List Inquiry Duty. Maintain Location Lists Priv
Supply Chain Network Inquiry Duty	A duty for viewing supplier chain network information.	View Supply Chain Network Priv
Supply Chain Network Management Duty	A duty for maintaining supply chain network information. This duty is an extension of the Supply Chain Network Inquiry Duty.	All privileges included in the Supply Chain Network Inquiry Duty. Maintain Supply Chain Network Priv
Merchandise Hierarchy Inquiry Duty	A duty for searching for and viewing the merchandise hierarchy, viewing defaults defined at various levels of the merchandise hierarchy, viewing pending merchandise hierarchy changes. This also includes viewing all grouping levels of the hierarchy, Division, Group, Department, Class, and Subclass, and viewing department level up charges and VAT information.	Search Merchandise Hierarchy Priv View Merchandise Hierarchy Priv View Divisions Priv View Groups Priv View Departments Priv View Classes Priv View Subclasses Priv View Merchandise Hierarchy Defaults Priv View Up Charges Priv View VAT Priv
Merchandise Hierarchy Management Duty	A duty for creating and maintaining the grouping levels of the merchandise hierarchy, creating and maintaining defaults at various levels of the hierarchy, creating and maintaining item reclassifications and maintaining pending merchandise hierarchy changes. This also includes maintaining the details of all grouping levels of the hierarchy, Division, Group, Department, Class and Subclass, and maintaining department level up charges and VAT information. This duty is an extension of the Merchandise Hierarchy Inquiry Duty.	All privileges included in the Merchandise Hierarchy Inquiry Duty. Maintain Merchandise Hierarchy Priv Maintain Divisions Priv Maintain Groups Priv Maintain Departments Priv Maintain Classes Priv Maintain Subclasses Priv Maintain Merchandise Hierarchy Defaults Priv Maintain Reclassifications Priv Maintain Up Charges Priv Maintain VAT Priv
Organizational Hierarchy Inquiry Duty	A duty for searching for and viewing the Organization Hierarchy. This also includes viewing all grouping levels of the hierarchy, Chain, Area, Region, and District.	View Organizational Hierarchy Priv View Chains Priv View Areas Priv View Regions Priv View Districts Priv

Duty	Duty Description	Privileges
Organizational Hierarchy Management Duty	A duty for creating and maintaining the grouping levels of the Organization Hierarchy. This includes all grouping levels of the hierarchy, Chain, Area, Region, and District. This duty is an extension of the Organizational Hierarchy Inquiry Duty.	All privileges included in the Organizational Hierarchy Inquiry Duty. Maintain Chains Priv Maintain Areas Priv Maintain Regions Priv Maintain Districts Priv
DSD Management Duty	A duty for maintaining DSD purchase orders.	Maintain DSD Orders Priv
Pre-issued Order Numbers Management Duty	A duty for generating pre-issue order numbers and creating orders using pre-issued numbers.	Maintain Pre-Issued Order Numbers Priv
Purchase Order Inquiry Duty	A duty for searching for and viewing purchase orders.	Search Purchase Orders Priv View Purchase Orders Priv
Purchase Order Management Duty	A duty for maintaining and submitting purchase orders. This duty is an extension of the Purchase Order Inquiry Duty.	All privileges included in the Purchase Order Inquiry Duty. Maintain Purchase Orders Priv Submit Purchase Orders Priv
Purchase Order Approval Duty	A duty for approving purchase orders. This duty is an extension of the Purchase Order Management Duty.	All privileges included in the Purchase Order Management Duty. Approve Purchase Orders Priv
Receipt Adjustments by Cost Management Duty	A duty for creating receipt adjustments by cost.	Maintain Receipt Adjustments by Cost Priv
POS Configuration Inquiry Duty	A duty for searching for and viewing coupons and product restrictions.	View Coupons Priv View Product Restrictions Priv
POS Configuration Management Duty	A duty for maintaining coupons and product restrictions. This duty is an extension of the POS Configuration Inquiry Duty.	All privileges included in the POS Configuration Inquiry Duty. Maintain Coupons Priv Maintain Product Restrictions Priv
Replenishment Inquiry Duty	A duty for searching for and viewing replenishment attributes.	View Replenishment Attributes Priv
Replenishment Management Duty	A duty for maintaining replenishment attributes. This duty is an extension of the Replenishment Inquiry Duty.	All privileges included in the Replenishment Inquiry Duty. Maintain Replenishment Attributes Priv
Required Document Inquiry Duty	A duty for viewing required documents.	View Required Documents Priv

Duty	Duty Description	Privileges
Required Document Management Duty	A duty for maintaining required documents. This duty is an extension of the Required Documents Inquiry Duty.	All privileges included in the Required Documents Inquiry Duty. Maintain Required Documents Priv
Stock Count Inquiry Duty	A duty for searching for and viewing stock count requests and stock count unit results.	View Stock Counts Priv View Stock Count Units Priv
Stock Count Management Duty	A duty for maintaining stock count requests and stock count unit results. This duty is an extension of the Stock Count Inquiry Duty.	All privileges included in the Stock Count Inquiry Duty. Maintain Stock Counts Priv Maintain Stock Count Units Priv
Stock Count Value Management Duty	A duty for maintaining stock count value variances and adjustments. This duty is an extension of the Stock Count Inquiry Duty.	All privileges included in the Stock Count Inquiry Duty. Maintain Stock Count Values Priv
Stock Ledger Inquiry Duty	A duty for viewing the stock ledger.	View Stock Ledger Priv
Stock Ledger Management Duty	A duty for maintaining the stock ledger. This duty is an extension of the Stock Ledger Inquiry Duty.	All privileges included in the Stock Ledger Inquiry Duty. Maintain Stock Ledger Priv
Buyer Dashboard Inquiry Duty	A duty for viewing the Buyer Dashboard.	View Buyer Dashboard Priv
Finance Analyst Inquiry Duty	A duty for viewing the Finance Analyst Dashboard.	View Finance Analyst Dashboard Priv
Inventory Analyst Inquiry Duty	A duty for viewing the Inventory Analyst Dashboard.	View Inventory Analyst Dashboard Priv
Inventory Control Inquiry Duty	A duty for viewing the Inventory Control Dashboard.	View Inventory Control Dashboard Priv
Data Steward Inquiry Duty	A duty for viewing the Data Steward Dashboard.	View Data Steward Dashboard Priv
Partner Inquiry Duty	A duty for searching for and viewing partners.	Search Partners Priv View Partners Priv
Partner Management Duty	A duty for maintaining partners. This duty is an extension of the Partner Inquiry Duty.	All privileges included in the Partner Inquiry Duty. Maintain Partners Priv
Partner Activation Duty	A duty for activating or deactivating a partner. This duty is an extension of the Partner Management Duty.	All privileges included in the Partner Management Duty. Activate Partners Priv Deactivate Partners Priv
Supplier Inquiry Duty	A duty for searching for and viewing suppliers and supplier sites.	Search Suppliers Priv View Suppliers Priv

Duty	Duty Description	Privileges
Supplier Management Duty	A duty for maintaining suppliers and supplier sites. This duty is an extension of the Supplier Inquiry Duty.	All privileges included in the Supplier Inquiry Duty. Maintain Suppliers Priv
Supplier Activation Duty	A duty for activating or deactivating a supplier or supplier site. This duty is an extension of the Supplier Management Duty.	All privileges included in the Supplier Management Duty. Activate Suppliers Priv Deactivate Suppliers Priv
Supplier Inventory Inquiry Duty	A duty for viewing supplier inventory information.	View Supplier Inventory Management Priv
Supplier Inventory Management Duty	A duty for maintaining supplier inventory information. This duty is an extension of the Supplier Inventory Inquiry Duty.	All privileges included in the Supplier Inventory Inquiry Duty. Maintain Supplier Inventory Management Priv
VAT Inquiry Duty	A duty for viewing department VAT information.	View VAT Priv
VAT Management Duty	A duty for maintaining department VAT information. This duty is an extension of the VAT Inquiry Duty.	All privileges included in the VAT Inquiry Duty. Maintain VAT Priv

Role Mapping

Roles												
	Persona	Roger	Stuart	Benny	Ivan	Ian	Connor	Isaac	Sarah	Fatima	Shawn	Fran
Duty	Privilege	RMS/RTM Application Administrator	Data Steward	Buyer	Inventory Analyst	Inventory Manager	Corporate Inventory Control Analyst	Inventory Control Manger	Sourcing Analyst	Finance Analyst	Supply Chain Analyst	Finance Manager
Application Admin Inquiry Duty	View Application Administration Priv	x	x									
Application Admin Management Duty	Maintain Application Administration Priv	x										
Report Options Inquiry Duty	View Report Options Priv	x	x									
Report Options Management Duty	Maintain Report Options Priv	x	x									
Centralized Data Translation Inquiry Duty	View Translations Priv	x										
Centralized Data Translation Management Duty	Maintain Translations Priv	x										
Asynchronous Job Inquiry Duty	View Asynchronous Jobs Priv	x	x	x	x	x	x	x	x	x	x	
Asynchronous Job Management Duty	Maintain Asynchronous Jobs Priv	x	x	x	x	x	x	x	x	x	x	

Roles												
	Persona	Roger	Stuart	Benny	Ivan	Ian	Connor	Isaac	Sarah	Fatima	Shawn	Fran
Duty	Privilege	RMS/RTM Application Administrator	Data Steward	Buyer	Inventory Analyst	Inventory Manager	Corporate Inventory Control Analyst	Inventory Control Manger	Sourcing Analyst	Finance Analyst	Supply Chain Analyst	Finance Manager
Custom Flex Attributes Management Duty	Maintain Custom Flex Attributes Priv	x										
Custom Flex Attributes Inquiry Duty	View Custom Flex Attributes Priv	x										
Foundation Data via Spreadsheet Inquiry Duty	View Foundation Data via Spreadsheet Priv	x	x	x	x	x	x	x	x	x	x	x
Foundation Data via Spreadsheet Management Duty	Maintain Foundation Data via Spreadsheet Priv	x	x									
Competitor Inquiry Duty	View Competitive Shop Priv	x	x	x						x		
Competitor Management Duty	Maintain Competitive Shop Priv	x	x									
Contract Approval Duty	Approve Contracts Priv	x		x		x						

Roles												
	Persona	Roger	Stuart	Benny	Ivan	Ian	Connor	Isaac	Sarah	Fatima	Shawn	Fran
Duty	Privilege	RMS/RTM Application Administrator	Data Steward	Buyer	Inventory Analyst	Inventory Manager	Corporate Inventory Control Analyst	Inventory Control Manger	Sourcing Analyst	Finance Analyst	Supply Chain Analyst	Finance Manager
Contract Inquiry Duty	Search Contracts Priv	x		x	x	x						
Contract Inquiry Duty	View Contracts Priv	x		x	x	x						
Contract Management Duty	Maintain Contracts Priv	x		x	x	x						
Contract Management Duty	Submit Contracts Priv	x		x	x	x						
Cost Change Approval Duty	Approve Cost Changes Priv	x		x								
Cost Change Inquiry Duty	Search Cost Changes Priv											
Cost Change Inquiry Duty	View Cost Changes Priv	x	x	x					x			
Cost Change Management Duty	Maintain Cost Changes Priv	x	x	x								
Cost Change Management Duty	Submit Cost Changes Priv	x	x	x								
Cost Zone Inquiry Duty	View Cost Zones Priv	x	x						x		x	

Roles												
	Persona	Roger	Stuart	Benny	Ivan	Ian	Connor	Isaac	Sarah	Fatima	Shawn	Fran
Duty	Privilege	RMS/RTM Application Administrator	Data Steward	Buyer	Inventory Analyst	Inventory Manager	Corporate Inventory Control Analyst	Inventory Control Manger	Sourcing Analyst	Finance Analyst	Supply Chain Analyst	Finance Manager
Cost Zone Management Duty	Maintain Cost Zones Priv	x	x						x		x	
Buyer Dashboard Inquiry Duty	View Buyer Dashboard Priv	x		x								
Finance Analyst Inquiry Duty	View Finance Analyst Dashboard Priv	x								x		X
Inventory Analyst Inquiry Duty	View Inventory Analyst Dashboard Priv	x			x	x						
Inventory Control Inquiry Duty	View Inventory Control Dashboard Priv	x					x	x				
Data Steward Inquiry Duty	View Data Steward Dashboard Priv	x	x									
Deal Approval Duty	Approve Deals Priv	x		x						x		
Deal Inquiry Duty	Search Deals Priv	x		x						x		

Roles												
	Persona	Roger	Stuart	Benny	Ivan	Ian	Connor	Isaac	Sarah	Fatima	Shawn	Fran
Duty	Privilege	RMS/RTM Application Administrator	Data Steward	Buyer	Inventory Analyst	Inventory Manager	Corporate Inventory Control Analyst	Inventory Control Manger	Sourcing Analyst	Finance Analyst	Supply Chain Analyst	Finance Manager
Deal Inquiry Duty	View Deals Priv	x		x						x		
Deal Management Duty	Maintain Deals Priv	x		x						x		
Deal Management Duty	Submit Deals Priv	x		x						x		
Expense Inquiry Duty	View Expense Profiles Priv	x							x		x	x
Expense Management Duty	Maintain Expense Profiles Priv	x							x		x	x
Up Charge Inquiry Duty	View Up Charges Priv	x									x	x
Up Charge Management Duty	Maintain Up Charges Priv	x									x	x
Average Cost Management Duty	Maintain Average Cost Priv	x										x
Budgets Inquiry Duty	View Budgets Priv	x		x	x	x				x		x
Budgets Management Duty	Maintain Budgets Priv	x		x								

Roles												
	Persona	Roger	Stuart	Benny	Ivan	Ian	Connor	Isaac	Sarah	Fatima	Shawn	Fran
Duty	Privilege	RMS/RTM Application Administrator	Data Steward	Buyer	Inventory Analyst	Inventory Manager	Corporate Inventory Control Analyst	Inventory Control Manger	Sourcing Analyst	Finance Analyst	Supply Chain Analyst	Finance Manager
Finance Admin Inquiry Duty	View Finance Admin Priv	x	x							x		x
Finance Admin Management Duty	Maintain Finance Admin Priv	x	x							x		
Financial Control Inquiry Duty	View Financial Control Priv	x								x		x
Financial Control Management Duty	Maintain Financial Control Priv	x								x		X
GL Drill Back Inquiry Duty	View GL Drill Back Priv	x	x	x								
Tran Data Inquiry Duty	View Transaction Data Priv	x	x	x								
Franchise Cost Inquiry Duty	View Franchise Cost Priv	x		x								
Franchise Cost Management Duty	Maintain Franchise Cost Priv	x		x								
Franchise Foundation Inquiry Duty	View Franchise Foundation Priv	x	x	x								

Roles												
	Persona	Roger	Stuart	Benny	Ivan	Ian	Connor	Isaac	Sarah	Fatima	Shawn	Fran
Duty	Privilege	RMS/RTM Application Administrator	Data Steward	Buyer	Inventory Analyst	Inventory Manager	Corporate Inventory Control Analyst	Inventory Control Manger	Sourcing Analyst	Finance Analyst	Supply Chain Analyst	Finance Manager
Franchise Foundation Management Duty	Maintain Franchise Foundation Priv	x	x	x								
Franchise Orders and Returns Approval Duty	Approve Franchise Orders and Returns Priv	x		x	x	x						
Franchise Orders and Returns Inquiry Duty	View Franchise Orders and Returns Priv	x		x	x	x						
Franchise Orders and Returns Management Duty	Maintain Franchise Orders and Returns Priv	x		x	x	x						
ALC Inquiry Duty	Search Actual Landed Cost Priv	x							x			x
ALC Inquiry Duty	View Actual Landed Cost Priv	x							x			x
ALC Management Duty	Finalize Actual Landed Cost Priv	x							x			x

Roles												
	Persona	Roger	Stuart	Benny	Ivan	Ian	Connor	Isaac	Sarah	Fatima	Shawn	Fran
Duty	Privilege	RMS/RTM Application Administrator	Data Steward	Buyer	Inventory Analyst	Inventory Manager	Corporate Inventory Control Analyst	Inventory Control Manger	Sourcing Analyst	Finance Analyst	Supply Chain Analyst	Finance Manager
Customs Entry Confirmation Duty	Confirm Customs Entries Priv	x							x			
Customs Entry Inquiry Duty	Search Customs Entries Priv	x			x	x	x	x	x	x		x
Customs Entry Inquiry Duty	View Customs Entries Priv	x			x	x	x	x	x	x		x
Customs Entry Management Duty	Maintain Customs Entries Priv	x							x			
Customs Entry Management Duty	Submit Customs Entries Priv	x							x			
Import Foundation Inquiry Duty	View Import Foundation Priv	x	x	x								
Import Foundation Management Duty	Maintain Import Foundation Priv	x	x	x								
Import Item Inquiry Duty	View Import Items Priv	x	x						x			
Import Item Management Duty	Maintain Import Items Priv	x	x						x			

Roles												
	Persona	Roger	Stuart	Benny	Ivan	Ian	Connor	Isaac	Sarah	Fatima	Shawn	Fran
Duty	Privilege	RMS/RTM Application Administrator	Data Steward	Buyer	Inventory Analyst	Inventory Manager	Corporate Inventory Control Analyst	Inventory Control Manger	Sourcing Analyst	Finance Analyst	Supply Chain Analyst	Finance Manager
Letter of Credit Approval Duty	Approve Letters of Credit Priv	x							x			x
Letter of Credit Approval Duty	Confirm Letters of Credit Priv	x							x			x
Letter of Credit Inquiry Duty	Search Letters of Credit Priv	x							x			x
Letter of Credit Inquiry Duty	View Letters of Credit Priv	x							x			x
Letter of Credit Management Duty	Maintain Letters of Credit Priv	x							x			x
Letter of Credit Management Duty	Submit Letters of Credit Priv	x							x			x
Obligation Approval Duty	Approve Obligations Priv	x							x			x
Obligation Inquiry Duty	Search Obligations Priv	x							x	x		x
Obligation Inquiry Duty	View Obligations Priv	x							x	x		x

Roles												
	Persona	Roger	Stuart	Benny	Ivan	Ian	Connor	Isaac	Sarah	Fatima	Shawn	Fran
Duty	Privilege	RMS/RTM Application Administrator	Data Steward	Buyer	Inventory Analyst	Inventory Manager	Corporate Inventory Control Analyst	Inventory Control Manger	Sourcing Analyst	Finance Analyst	Supply Chain Analyst	Finance Manager
Obligation Management Duty	Maintain Obligations Priv	x							x			x
Transportation Inquiry Duty	Search Transportation Priv	x			x	x	x	x	x	x	x	x
Transportation Inquiry Duty	View Transportation Priv	x			x	x	x	x	x	x	x	x
Transportation Management Duty	Finalize Transportation Priv	x							x		x	
Transportation Management Duty	Maintain Transportation Priv	x							x		x	
Inventory Adjustment Management Duty	Maintain Inventory Adjustments Priv	x					x	x				
Inventory and Sales Inquiry Duty	View Item Inventory and Sales Priv	x		x	x	x	x	x		x		
Intercompany Transfer Approval Duty	Approve Intercompany Transfers Priv	x			x	x		x				

Roles												
	Persona	Roger	Stuart	Benny	Ivan	Ian	Connor	Isaac	Sarah	Fatima	Shawn	Fran
Duty	Privilege	RMS/RTM Application Administrator	Data Steward	Buyer	Inventory Analyst	Inventory Manager	Corporate Inventory Control Analyst	Inventory Control Manger	Sourcing Analyst	Finance Analyst	Supply Chain Analyst	Finance Manager
Intercompany Transfer Inquiry Duty	View Intercompany Transfers Priv	x			x	x	x	x		x		x
Intercompany Transfer Management Duty	Maintain Intercompany Transfers Priv	x			x	x	x	x		x		x
Intercompany Transfer Management Duty	Submit Intercompany Transfers Priv	x			x	x	x	x		x		x
MRT Approval Duty	Approve MRTs Priv	x			x	x						
MRT Inquiry Duty	Search MRTs Priv	x			x	x	x	x		x		x
MRT Inquiry Duty	View MRTs Priv	x			x	x	x	x		x		x
MRT Management Duty	Maintain MRTs Priv	x			x	x	x	x				
MRT Management Duty	Submit MRTs Priv	x			x	x	x	x				
RTV Approval Duty	Approve RTVs Priv	x			x	x						

Roles												
	Persona	Roger	Stuart	Benny	Ivan	Ian	Connor	Isaac	Sarah	Fatima	Shawn	Fran
Duty	Privilege	RMS/RTM Application Administrator	Data Steward	Buyer	Inventory Analyst	Inventory Manager	Corporate Inventory Control Analyst	Inventory Control Manger	Sourcing Analyst	Finance Analyst	Supply Chain Analyst	Finance Manager
RTV Inquiry Duty	Search RTVs Priv	x		x	x	x	x	x		x		x
RTV Inquiry Duty	View RTVs Priv	x		x	x	x	x	x		x		x
RTV Management Duty	Maintain RTVs Priv	x			x	x						
Shipment and Receipts Inquiry Duty	View Shipments and Receipts Priv	x					x	x		x		x
Shipment and Receipts Management Duty	Maintain Shipments and Receipts Priv	x					x	x				
Transfer Approval Duty	Approve Transfers Priv	x			x	x		x				
Transfer Inquiry Duty	View Transfers Priv	x			x	x	x	x		x		x
Transfer Inquiry Duty AND Intercompany Transfer Inquiry Duty	Search Transfers Priv	x			x	x	x	x		x		x

Roles												
	Persona	Roger	Stuart	Benny	Ivan	Ian	Connor	Isaac	Sarah	Fatima	Shawn	Fran
Duty	Privilege	RMS/RTM Application Administrator	Data Steward	Buyer	Inventory Analyst	Inventory Manager	Corporate Inventory Control Analyst	Inventory Control Manger	Sourcing Analyst	Finance Analyst	Supply Chain Analyst	Finance Manager
Transfer Management Duty	Maintain Transfers Priv	x			x	x	x	x		x		x
Transfer Management Duty	Submit Transfers Priv	x			x	x	x	x				
Transfer with Finishing Management Duty	Maintain Transfers with Finishing Priv	x			x	x	x	x				
Diff Inquiry Duty	View Diffs Priv	x	x									
Diff Management Duty	Maintain Diffs Priv	x	x									
Diff Management Duty	Use Diffs Priv	x	x	x	x	x	x	x	x	x	x	x
Diff Ratio Inquiry Duty	Search Diff Ratios Priv	x	x	x	x	x						
Diff Ratio Inquiry Duty	View Diff Ratios Priv	x	x	x	x	x						
Diff Ratio Management Duty	Maintain Diff Ratios Priv	x	x	x	x	x						

Roles												
	Persona	Roger	Stuart	Benny	Ivan	Ian	Connor	Isaac	Sarah	Fatima	Shawn	Fran
Duty	Privilege	RMS/RTM Application Administrator	Data Steward	Buyer	Inventory Analyst	Inventory Manager	Corporate Inventory Control Analyst	Inventory Control Manger	Sourcing Analyst	Finance Analyst	Supply Chain Analyst	Finance Manager
Grocery Attribute Management Duty	Maintain/View Grocery Attributes Priv	x	x	x	x	x						
Item Approval Duty	Approve Items Priv	x	x	x	x	x						
Item Inquiry Duty	Search Items Priv	x	x	x	x	x	x	x	x	x	x	x
Item Inquiry Duty	View Items Priv	x	x	x	x	x	x	x	x	x	x	x
Item List Inquiry Duty	Search Item Lists Priv	x	x	x	x	x				x	x	
Item List Inquiry Duty	View Item Lists Priv	x	x	x	x	x				x	x	
Item List Management Duty	Maintain Item Lists Priv	x	x	x	x	x				x	x	
Item Location Inquiry Duty	View Item Locations Priv	x	x	x	x	x	x	x	x	x	x	X
Item Location Management Duty	Maintain Item Locations Priv	x	x		x	x					x	
Item Management Duty	Maintain Items Priv	x	x	x	x	x						

Roles												
	Persona	Roger	Stuart	Benny	Ivan	Ian	Connor	Isaac	Sarah	Fatima	Shawn	Fran
Duty	Privilege	RMS/RTM Application Administrator	Data Steward	Buyer	Inventory Analyst	Inventory Manager	Corporate Inventory Control Analyst	Inventory Control Manger	Sourcing Analyst	Finance Analyst	Supply Chain Analyst	Finance Manager
Item Management Duty	Submit Items Priv	x	x	x	x	x						
Item Supplier Inquiry Duty	View Item Suppliers Priv	x	x	x	x	x	x	x	x	x	x	x
Item Supplier Management Duty	Maintain Item Suppliers Priv	x	x		x	x			x		x	
Price Change Inquiry Duty	View Price Change History Priv	x	x	x						x		x
Ticket Management Duty	Maintain Ticket Requests Priv	x	x		x	x						
Location Inquiry Duty	View Stores Priv	x	x								x	
Location Inquiry Duty	View Warehouses Priv	x	x								x	
Location List Inquiry Duty	Search Location Lists Priv	x	x	x	x	x				x	x	
Location List Inquiry Duty	View Location Lists Priv	x	x	x	x	x				x	x	

Roles												
	Persona	Roger	Stuart	Benny	Ivan	Ian	Connor	Isaac	Sarah	Fatima	Shawn	Fran
Duty	Privilege	RMS/RTM Application Administrator	Data Steward	Buyer	Inventory Analyst	Inventory Manager	Corporate Inventory Control Analyst	Inventory Control Manger	Sourcing Analyst	Finance Analyst	Supply Chain Analyst	Finance Manager
Location List Management Duty	Maintain Location Lists Priv	x	x	x	x	x				x	x	
Location Management Duty	Maintain Stores Priv	x	x									
Location Management Duty	Maintain Warehouses Priv	x	x									
Supply Chain Network Inquiry Duty	View Supply Chain Network Priv	x	x								x	
Supply Chain Network Management Duty	Maintain Supply Chain Network Priv	x	x									x
Merchandise Hierarchy Inquiry Duty	Search Merchandise Hierarchy Priv	x	x									
Merchandise Hierarchy Inquiry Duty	View Classes Priv	x	x									
Merchandise Hierarchy Inquiry Duty	View Departments Priv	x	x									

Roles												
	Persona	Roger	Stuart	Benny	Ivan	Ian	Connor	Isaac	Sarah	Fatima	Shawn	Fran
Duty	Privilege	RMS/RTM Application Administrator	Data Steward	Buyer	Inventory Analyst	Inventory Manager	Corporate Inventory Control Analyst	Inventory Control Manger	Sourcing Analyst	Finance Analyst	Supply Chain Analyst	Finance Manager
Merchandise Hierarchy Inquiry Duty	View Divisions Priv	x	x									
Merchandise Hierarchy Inquiry Duty	View Groups Priv	x	x									
Merchandise Hierarchy Inquiry Duty	View Merchandise Hierarchy Defaults Priv	x	x									
Merchandise Hierarchy Inquiry Duty	View Merchandise Hierarchy Priv	x	x									
Merchandise Hierarchy Inquiry Duty	View Subclasses Priv	x	x									
Merchandise Hierarchy Management Duty	Maintain Classes Priv	x	x									
Merchandise Hierarchy Management Duty	Maintain Departments Priv	x	x									

Roles												
	Persona	Roger	Stuart	Benny	Ivan	Ian	Connor	Isaac	Sarah	Fatima	Shawn	Fran
Duty	Privilege	RMS/RTM Application Administrator	Data Steward	Buyer	Inventory Analyst	Inventory Manager	Corporate Inventory Control Analyst	Inventory Control Manger	Sourcing Analyst	Finance Analyst	Supply Chain Analyst	Finance Manager
Merchandise Hierarchy Management Duty	Maintain Divisions Priv	x	x									
Merchandise Hierarchy Management Duty	Maintain Groups Priv	x	x									
Merchandise Hierarchy Management Duty	Maintain Merchandise Hierarchy Defaults Priv	x	x									
Merchandise Hierarchy Management Duty	Maintain Merchandise Hierarchy Priv	x	x									
Merchandise Hierarchy Management Duty	Maintain Reclassifications Priv	x	x									
Merchandise Hierarchy Management Duty	Maintain Subclasses Priv	x	x									
Organizational Hierarchy Inquiry Duty	View Areas Priv	x	x									

Roles												
	Persona	Roger	Stuart	Benny	Ivan	Ian	Connor	Isaac	Sarah	Fatima	Shawn	Fran
Duty	Privilege	RMS/RTM Application Administrator	Data Steward	Buyer	Inventory Analyst	Inventory Manager	Corporate Inventory Control Analyst	Inventory Control Manger	Sourcing Analyst	Finance Analyst	Supply Chain Analyst	Finance Manager
Organizational Hierarchy Inquiry Duty	View Chains Priv	x	x									
Organizational Hierarchy Inquiry Duty	View Districts Priv	x	x									
Organizational Hierarchy Inquiry Duty	View Organizational Hierarchy Priv	x	x									
Organizational Hierarchy Inquiry Duty	View Regions Priv	x	x									
Organizational Hierarchy Management Duty	Maintain Areas Priv	x	x									
Organizational Hierarchy Management Duty	Maintain Chains Priv	x	x									
Organizational Hierarchy Management Duty	Maintain Districts Priv	x	x									

Roles												
	Persona	Roger	Stuart	Benny	Ivan	Ian	Connor	Isaac	Sarah	Fatima	Shawn	Fran
Duty	Privilege	RMS/RTM Application Administrator	Data Steward	Buyer	Inventory Analyst	Inventory Manager	Corporate Inventory Control Analyst	Inventory Control Manger	Sourcing Analyst	Finance Analyst	Supply Chain Analyst	Finance Manager
Organizational Hierarchy Management Duty	Maintain Regions Priv	x	x									
DSD Management Duty	Maintain DSD Orders Priv	x										x
Pre-issued Order Numbers Management Duty	Maintain Pre-Issued Order Numbers Priv	x		x	x	x						
Purchase Order Approval Duty	Approve Purchase Orders Priv	x		x	x							
Purchase Order Inquiry Duty	Search Purchase Orders Priv	x		x	x	x	x	x	x	x	x	x
Purchase Order Inquiry Duty	View Purchase Orders Priv	x		x	x	x	x	x	x	x	x	x
Purchase Order Management Duty	Close Purchase Orders Priv	x		x	x	x	x	x	x			
Purchase Order Management Duty	Maintain Purchase Orders Priv	x		x	x	x	x	x	x			

Roles												
	Persona	Roger	Stuart	Benny	Ivan	Ian	Connor	Isaac	Sarah	Fatima	Shawn	Fran
Duty	Privilege	RMS/RTM Application Administrator	Data Steward	Buyer	Inventory Analyst	Inventory Manager	Corporate Inventory Control Analyst	Inventory Control Manger	Sourcing Analyst	Finance Analyst	Supply Chain Analyst	Finance Manager
Purchase Order Management Duty	Submit Purchase Orders Priv	x		x	x	x	x	x	x			
Receipt Adjustments by Cost Management Duty	Maintain Receipt Adjustments by Cost Priv	x										x
POS Configuration Inquiry Duty	View Coupons Priv	x	x									
POS Configuration Inquiry Duty	View Product Restrictions Priv	x	x									
POS Configuration Management Duty	Maintain Coupons Priv	x	x									
POS Configuration Management Duty	Maintain Product Restrictions Priv	x	x									
Replenishment Inquiry Duty	View Replenishment Attributes Priv	x			x	x					x	

Roles												
	Persona	Roger	Stuart	Benny	Ivan	Ian	Connor	Isaac	Sarah	Fatima	Shawn	Fran
Duty	Privilege	RMS/RTM Application Administrator	Data Steward	Buyer	Inventory Analyst	Inventory Manager	Corporate Inventory Control Analyst	Inventory Control Manger	Sourcing Analyst	Finance Analyst	Supply Chain Analyst	Finance Manager
Replenishment Management Duty	Maintain Replenishment Attributes Priv	x			x	x						
Required Document Inquiry Duty	View Required Documents Priv	x							x			
Required Document Management Duty	Maintain Required Documents Priv	x							x			
Stock Count Inquiry Duty	View Stock Count Units Priv	x					x	x		x		x
Stock Count Inquiry Duty	View Stock Counts Priv	x					x	x		x		x
Stock Count Management Duty	Maintain Stock Count Units Priv	x					x	x				
Stock Count Management Duty	Maintain Stock Counts Priv	x					x	x				
Stock Count Value Management Duty	Maintain Stock Count Values Priv	x								x		x

Roles												
	Persona	Roger	Stuart	Benny	Ivan	Ian	Connor	Isaac	Sarah	Fatima	Shawn	Fran
Duty	Privilege	RMS/RTM Application Administrator	Data Steward	Buyer	Inventory Analyst	Inventory Manager	Corporate Inventory Control Analyst	Inventory Control Manger	Sourcing Analyst	Finance Analyst	Supply Chain Analyst	Finance Manager
Stock Ledger Inquiry Duty	View Stock Ledger Priv	x								x		x
Stock Ledger Management Duty	Maintain Stock Ledger Priv	x										x
Partner Activation Duty	Activate Partners Priv	x	x							x		x
Partner Activation Duty	Deactivate Partners Priv	x	x							x		x
Partner Inquiry Duty	Search Partners Priv	x	x						x	x		x
Partner Inquiry Duty	View Partners Priv	x	x						x	x		x
Partner Management Duty	Maintain Partners Priv	x	x						x			x
Supplier Activation Duty	Activate Suppliers Priv	x								x		x
Supplier Activation Duty	Deactivate Suppliers Priv	x								x		x
Supplier Inquiry Duty	Search Suppliers Priv	x	x						x	x	x	x

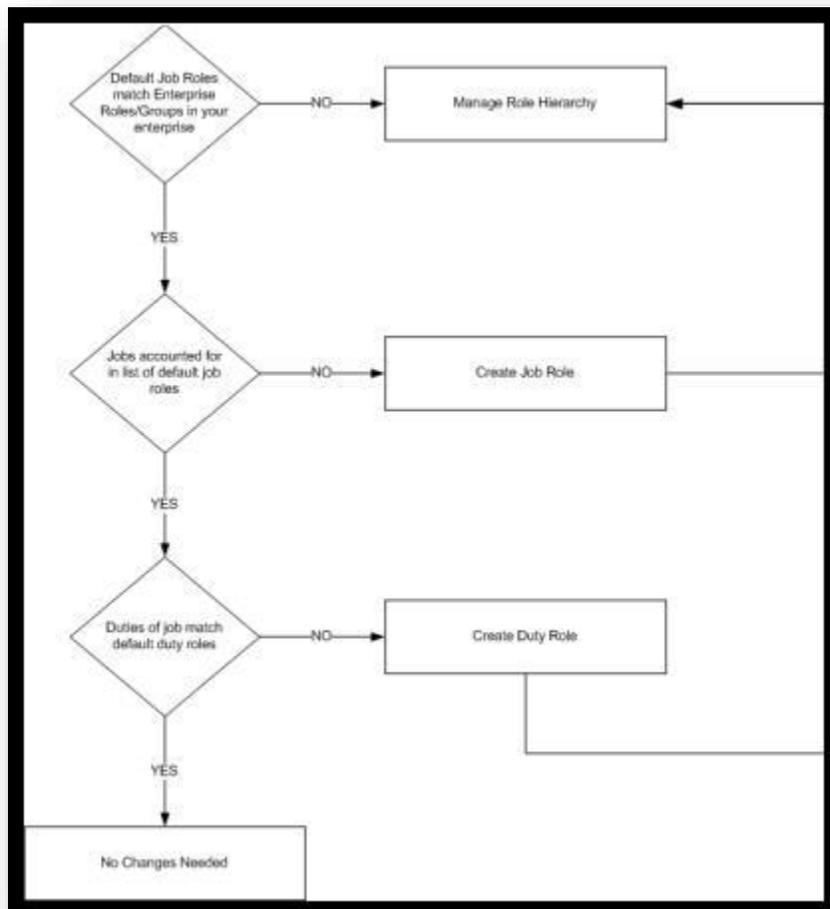
Roles												
	Persona	Roger	Stuart	Benny	Ivan	Ian	Connor	Isaac	Sarah	Fatima	Shawn	Fran
Duty	Privilege	RMS/RTM Application Administrator	Data Steward	Buyer	Inventory Analyst	Inventory Manager	Corporate Inventory Control Analyst	Inventory Control Manger	Sourcing Analyst	Finance Analyst	Supply Chain Analyst	Finance Manager
Supplier Inquiry Duty	View Suppliers Priv	x	x						x	x	x	x
Supplier Inventory Inquiry Duty	View Supplier Inventory Management Priv	x	x							x	x	x
Supplier Inventory Management Duty	Maintain Supplier Inventory Management Priv	x	x							x	x	
Supplier Management Duty	Maintain Suppliers Priv	x	x						x	x	x	x
VAT Inquiry Duty	View VAT Priv	x	x							x		x
VAT Management Duty	Maintain VAT Priv	x	x									x

Extend the Default Security Reference Implementation

Note: Make sure that the policy store is loaded with the default security configuration. For more information, see the latest *RMS Installation Guide*.

The common decisions made to match your enterprise to the default security reference implementation include the following:

- Do the default job roles match the equivalent job roles in your enterprise?
- Do the jobs in your enterprise exist in the security reference implementation?
- Do the duties performed by the jobs in your enterprise match the duties in the security reference implementation?



Note: It is important when constructing a role hierarchy that circular dependencies are not introduced. A best practice is to leave the default security configuration in place and first incorporate your customized application roles in a test environment.

Managing Roles in Retail Application Administration Console

Retail Applications provide a way in which retailers can modify the default roles to map to their security groups through the Retail Application Administration Console (RAAC).

RAAC is installed along with the Retail Application. Users with proper security privileges to access RAAC can launch RAAC by clicking on a link from the Retail Application's global menu.

For more details about using RAAC, see the latest *Merchandising Implementation Guide*.

Disabling Content

There are situations where administrators need to disable certain links or the default content such as Dashboards due to unavailability or other reasons. Retail Applications provide the flexibility to disable such content so that the application remains largely unaffected.

Safe Mode

Applications can choose to serve certain content such as dashboards to users upon launching the application. This is referred to as "Default Content". However sometimes this default content may cause delays in application launch after logging-in or worse it may render the application unusable.

To handle such scenarios Retail Applications provide a feature for Administrators called "Safe Mode" which allows the user to log in without serving up any default content. Once this mode is turned on, no default content is shown to any user when the application is launched.

To turn on this mode the property "uishell.load.safe.mode" must be set to true in the RetailAppsViewController.properties file.

Disabling Links in the Sidebar Menu

Administrators may occasionally need to disable content launchable from links in the sidebar menu. Retail Applications provide the ability to disable such links.

To disable a link the Administrator must first find the "id" of that link as specified in the SidebarNavigationModel.xml file. This value must then be provided to the property "uishell.sidebar.invalid.item.ids" within the RetailAppsViewController.properties file.

To disable more than one link, pass in multiple IDs separated by a comma.

Managing Oracle Metadata Services (MDS)

Retail Applications are built using ADF and one of the features within ADF is the Oracle Metadata Services (MDS) framework which provides a facility for retailers to customize the applications.

Refer to the document, Oracle® Fusion Middleware Developing Fusion Web Applications with Oracle Application Development Framework (<https://docs.oracle.com/middleware/1221/adf/develop/GUID-33CB77F6-6ECD-4B5D-9B95-5874AEDA1A78.htm#ADFFD2077>), for more information about MDS.

Overview of Oracle Metadata Services (MDS)

Oracle Metadata Services (MDS) is a key infrastructure component in Oracle Fusion Middleware. It is the layer through which metadata is loaded, saved, cached, stored, managed, and customized both by various middleware components and by the applications built on Fusion Middleware.

The use of MDS in ADF applications, for example, can allow applications to remember how users like to work, and therefore not require them to set up the application for every session. This may include, for example, saving of common searches and screen layouts for every user. This allows making use of the application easier and more intuitive for the users. MDS provides a foundation that can be leveraged by Oracle Application Development Framework (ADF) applications to provide such persistent personalization.

Oracle Metadata Services (MDS) makes use of metadata repositories or partitions. A Metadata repository or partition contains metadata for Oracle Fusion Middleware components. It can also contain metadata about the configuration of Oracle Fusion Middleware and metadata for applications. Oracle Metadata Services (MDS) stores the customizations in a metadata repository and retrieves them at runtime to merge the customizations with the base metadata to reveal the customized application.

A common issue when a patch is installed for a Retail Application is that certain screens would fail to load or UI elements fail to display data properly.

The cause of this issue is commonly attributed to user personalization on screen elements that are now removed in the patch.

For example, prior to patching the application, users may have saved search criterias on certain screens as a way to conveniently recall their desired search results whenever they use the application. Those saved search criterias are persisted by ADF in the MDS repository. If the patch involves the removal of one of the attributes used in the search criterias, applying the patch will cause the screens that have those search criterias fail to load.

The MDS repository is configured in the WebLogic server where the Retail Application is deployed. The repository is database-based and it is organized or subdivided into partitions. Retail Applications are deployed with their own partition within the server's MDS repository.

It is recommended to not delete the MDS partition during the upgrade of the Retail Application; instead use the functions described in this document to resolve any issues related to MDS.

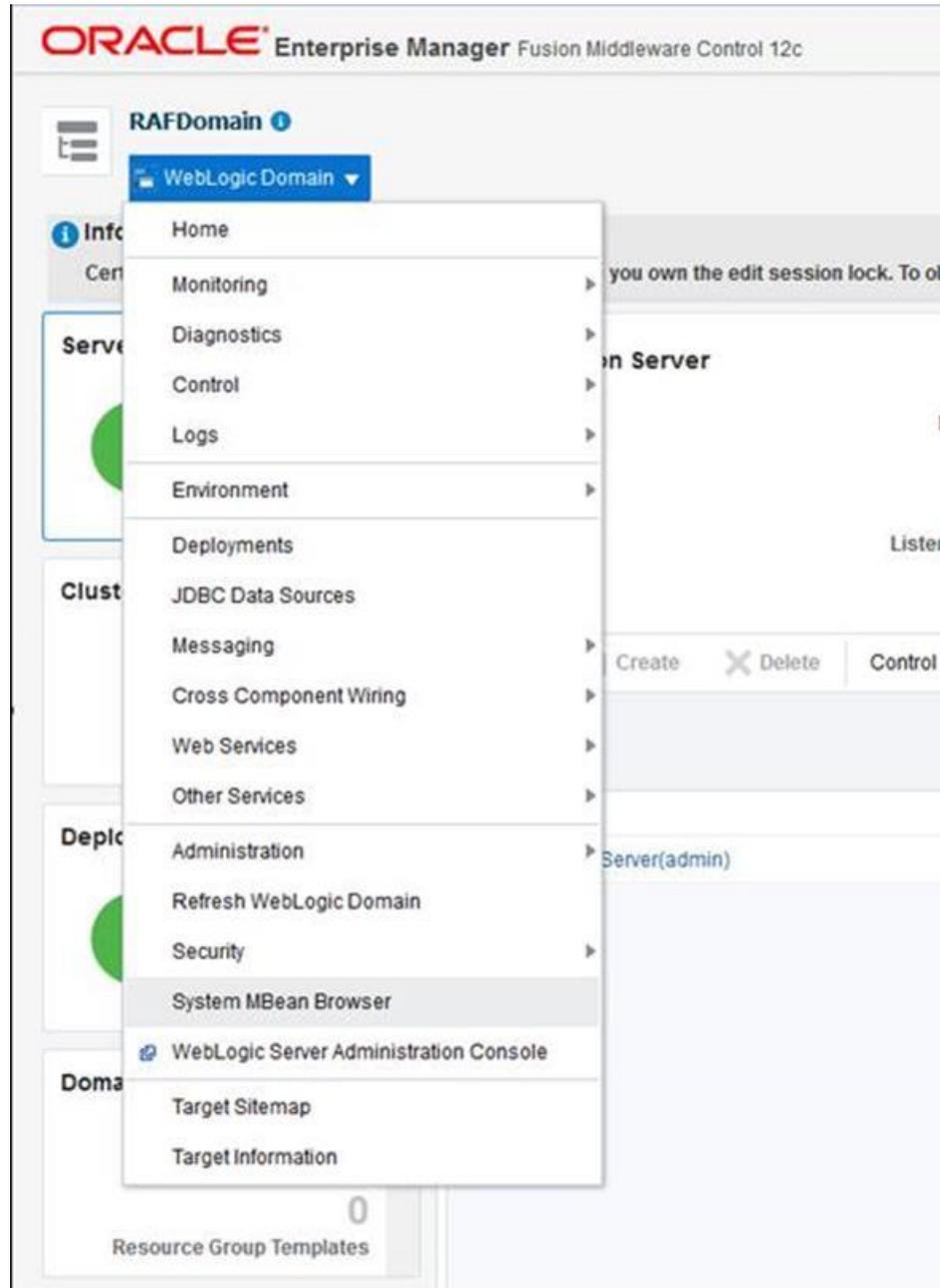
Using the System MBean Browser and the MDSAppRuntime MBean

For managing MDS Customizations in Retail Fusion Applications, use the Oracle Enterprise Manager to perform common metadata service tasks such as exporting, deleting, and importing of MDS Customizations. This can be performed in the Oracle Enterprise Manager using the System MBean Browser and the MDSAppRuntime MBean.

Perform the following steps to access the MDSAppRuntime MBean.

1. Log in to the Oracle Enterprise Manager by navigating to the URL in the following format:
`http://<host>:<port>/em/`

- Click the WebLogic Domain dropdown under the domain name. Navigate to System MBean Browser menu item and click System MBean Browser. The screenshot below shows the RAFDomain as the domain name.

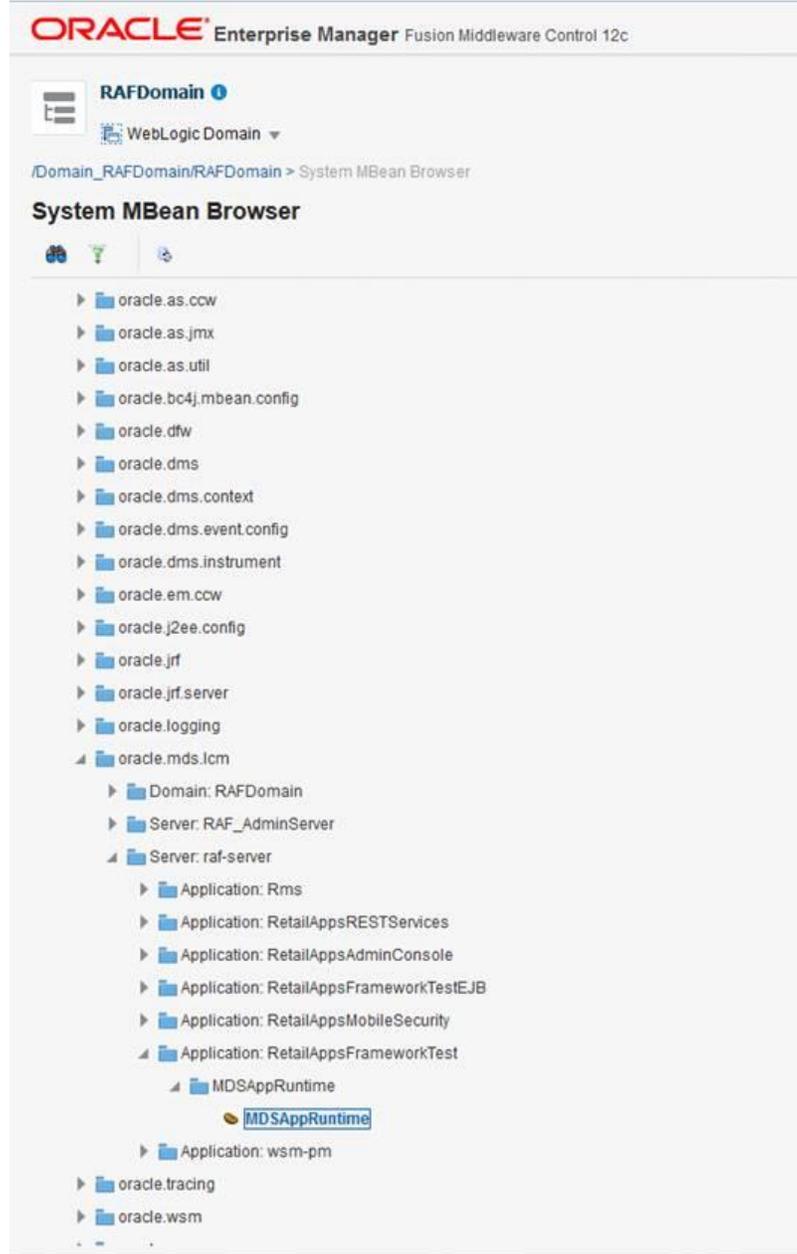


- Under Application Defined MBeans, locate MDSAppRuntime which can be found under the following folder structure:

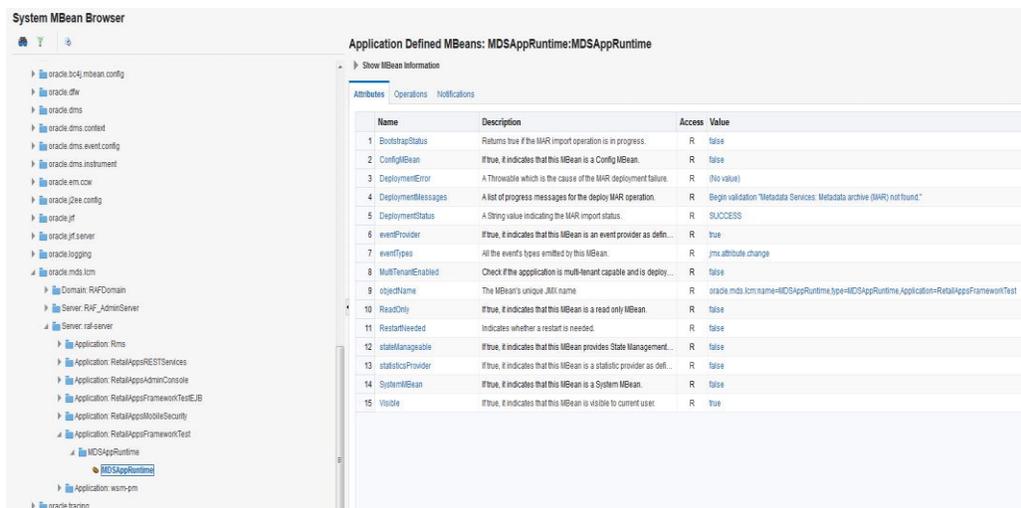
```

> oracle.mds.lcm
  > Server: AppServer
    > Application: Application Name
      > MDSAppRuntime
    
```

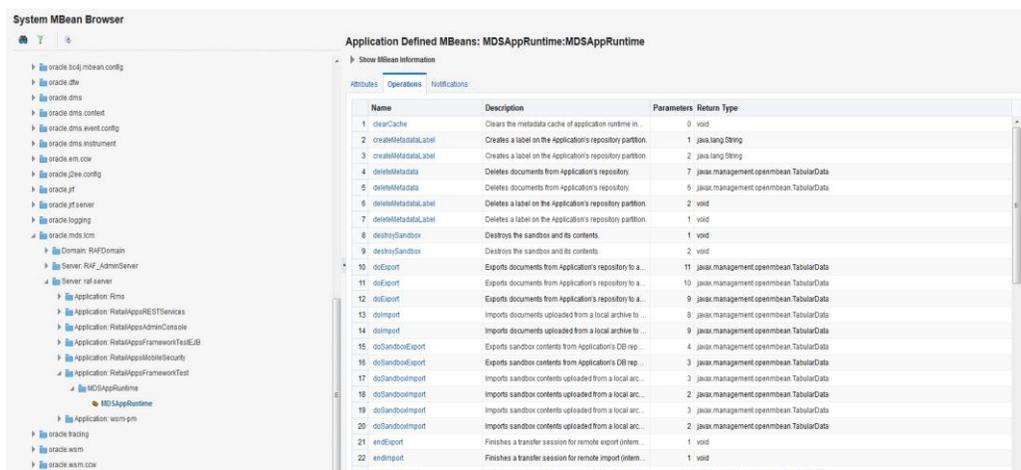
Choose the correct server and the Retail Application name based on your installation. The screenshot below displays RAFServer and RetailAppsFrameworkTest application.



4. Click on MDSAppRuntime management bean (MBean).



5. Select the Operations tab to view the operations available for the MDSAppRuntime MBean.



These are the operations which can be used in managing MDS Customizations. The exportMetadata, deleteMetadata, importMetadata, createMetadataLabel, listMetadataLabels, deleteMetadataLabel, and promoteMetadataLabel; will be briefly discussed in the following sections.

Exporting All MDS Customizations

1. Navigate to the MDSAppRuntime management bean (MBean), as described in the section "Using the System MBean Browser and the MDSAppRuntime MBean."
2. Export metadata by selecting the exportMetadata operation available from the Operations tab.
3. For toLocation, provide a valid absolute path to a directory or archive in the file system to which the selected documents will be exported. This location must be accessible from the machine where the application is running. If it does not exist, a directory will be created except that when the name ends with ".jar", ".JAR", ".zip", or ".ZIP", an archive will be created. Exporting metadata to an existing archive will overwrite the existing file.
Example: /tempDir/downloads/mdsExport/RetailAppsFrameworkTestMDS.zip

4. Click Invoke (located at the upper-right corner of the page) to proceed with the export operation.
5. Click Return to return to the list of operations.

Exporting MDS Customization for a Specific User

1. Navigate to the MDSAppRuntime management bean (MBean), as described in the section "Using the System MBean Browser and the MDSAppRuntime MBean".
2. Export metadata by selecting the **exportMetadata** operation available from the Operations tab.
3. For **toLocation**, provide a valid absolute path to a directory or archive in the file system to which the selected documents will be exported. This location must be accessible from the machine where the application is running. If it does not exist, a directory will be created except that when the name ends with ".jar", ".JAR", ".zip", or ".ZIP", an archive will be created. Exporting metadata to an existing archive will overwrite the existing file.

Example: /tempDir/downloads/mdsExport/RetailAppsFrameworkTestMDS.zip

4. For **docs**, click the pencil icon. On the Edit Parameter popup, click Add, and in the Element box, enter a list of comma-separated, fully qualified document names or document name patterns to export.
To export customizations for a specific page, simply enter the fully qualified base document name in the Element box.

Example:

/oracle/retail/apps/framework/uishell/skin/page/TestTablesAndTrees.jsff

You can provide the path to multiple documents to export, by clicking the Add button. Do not provide any docs in case you want to export all customizations for the user.

5. For **restrictCustTo**, click the pencil icon. On the Edit Parameter popup, click Add, and in the Element box, enter the list of customization layer names. This can be a list of comma-separated customization layer names used to restrict the operation to only customization documents that match the specified customization layers.
Each customization layer name can contain, within a pair of brackets, optional customization layer values and value patterns separated by commas. Wildcards (%) may also be used for restricting the operation.

For example:

Value	Description
user[buyer]	restricts operation to the user 'buyer'
user[buyer, betty_buyer]	restricts operation to the users 'buyer', and 'betty_buyer'
user[bu%]	restricts operation to user names that start with 'bu' (i.e. buyer)
user[be%]	restricts operation to user names that start with 'be' (i.e. betty_buyer)
user[%bu%]	restricts operation to users with 'bu' in the name (i.e. buyer and betty_buyer)

6. Click the Invoke button, located at the upper-right corner of the page, to proceed with the export operation.

7. A confirmation message will display in the page. Click Return to return to the list of operations.

Deleting All MDS Customizations for a User

1. Navigate to the MDSAppRuntime management bean (MBean), as described in the section "Using the System MBean Browser and the MDSAppRuntime MBean".
2. Delete metadata by selecting the **deleteMetadata** operation available from the Operations tab.
3. For **docs**, click the pencil icon. On the Edit Parameter popup, click Add, and in the Element box, enter a list of comma-separated, fully qualified document names or document name patterns

To delete all customizations for a user, enter `"/**"` (without the quotes). This will recursively delete all customizations under `"/` including any other customizations located in the folder(s) under it. Click OK.

4. For **restrictCustTo**, click the pencil icon. On the Edit Parameter popup, click Add, and in the Element box, enter the list of customization layer names. This can be used to restrict the operation to only customization documents that match the specified customization layers.

Each customization layer name can contain, within a pair of brackets, optional customization layer values and value patterns separated by commas. Wildcards (%) may also be used for restricting the operation.

For example

Value	Description
user[buyer]	restricts operation to the user 'buyer'
user[buyer, betty_buyer]	restricts operation to the users 'buyer', and 'betty_buyer'
user[bu%]	restricts operation to user names that start with 'bu' (i.e. buyer)
user[be%]	restricts operation to user names that start with 'be' (i.e. betty_buyer)
user[%bu%]	restricts operation to users with 'bu' in the name (i.e. buyer and betty_buyer)

5. Click the Invoke button, located at the upper-right corner of the page, to proceed with the delete operation.
6. A confirmation message will display in the page. Click Return to return to the list of operations.

Deleting a Customization for a Specific Page for All the Users

1. Navigate to the MDSAppRuntime management bean (MBean), as described in the section "Using the System MBean Browser and the MDSAppRuntime MBean".
2. Delete metadata by selecting the **deleteMetadata** operation available from the Operations tab.
3. For **excludeBaseDocs**, select true. This is a Boolean value indicating whether to exclude base metadata documents from being deleted.

4. For **docs**, click the pencil icon. On the Edit Parameter popup, click Add, and in the Element box, enter a list of comma-separated, fully qualified document names or document name patterns.
5. To delete customizations for a specific page, simply enter the fully qualified base document name in the Element box.
Example:
`/oracle/retail/apps/framework/uishell/skin/page/TestTablesAndTrees.jsff`
6. You can provide the path to multiple documents to be deleted, by clicking the Add button. When done, click on the OK button.
7. For **restrictCustTo**, click the pencil icon. On the Edit Parameter popup, click Add, and in the Element box, enter the list of customization layer names. This can be used to restrict the operation to only customization documents that match the specified customization layers.
Each customization layer name can contain, within a pair of brackets, optional customization layer values and value patterns separated by commas. Wildcards (%) may also be used for restricting the operation.
To delete customizations for all users, you may either:
 - a. remove any values for restrictCustTo or
 - b. enter "user[%]" in the Element box.

For Example:

Value	Description
user[buyer]	restricts operation to the user 'buyer'
user[buyer, betty_buyer]	restricts operation to the users 'buyer', and 'betty_buyer'
user[bu%]	restricts operation to user names that start with 'bu' (i.e. buyer)
user[be%]	restricts operation to user names that start with 'be' (i.e. betty_buyer)
user[%bu%]	restricts operation to users with 'bu' in the name (i.e. buyer and betty_buyer)

8. Click on the Invoke button, located at the upper-right corner of the page, to proceed with the delete operation.
9. A confirmation message will display in the page. Click Return to return to the list of operations.

Deleting a Customization for a Specific Page for a Particular User

1. Navigate to the MDSAppRuntime management bean (MBean), as described in the section "Using the System MBean Browser and the MDSAppRuntime MBean".
2. Delete metadata by selecting the **deleteMetadata** operation available from the Operations tab.
3. For **excludeBaseDocs**, select true. A Boolean value indicating whether to exclude base metadata documents from being deleted.
4. For docs, click the pencil icon. On the Edit Parameter popup, click Add, and in the Element box, enter a list of comma-separated, fully qualified document names or document name patterns.

5. To delete customizations for a specific page, simply enter the fully qualified base document name in the Element box.
Example:
/oracle/retail/apps/framework/uishell/skin/page/TestTablesAndTrees.jsff
6. You can provide the path to multiple documents to be deleted, by clicking the Add button. When done, click on the OK button.
7. For **restrictCustTo**, click the pencil icon. On the Edit Parameter popup, click Add, and in the Element box, enter the list of customization layer names. This can be used to restrict the operation to only customization documents that match the specified customization layers.
Each customization layer name can contain, within a pair of brackets, optional customization layer values and value patterns separated by commas. Wildcards (%) may also be used for restricting the operation.

To delete customizations for a particular user, enter "user[<username>]" in the Element box. See the table below for other examples.

For Example:

Value	Description
user[buyer]	restricts operation to the user 'buyer'
user[buyer, betty_buyer]	restricts operation to the users 'buyer', and 'betty_buyer'
user[bu%]	restricts operation to user names that start with 'bu' (i.e. buyer)
user[be%]	restricts operation to user names that start with 'be' (i.e. betty_buyer)
user[%bu%]	restricts operation to users with 'bu' in the name (i.e. buyer and betty_buyer)

8. Click on the Invoke button, located at the upper-right corner of the page, to proceed with the delete operation.
9. A confirmation message will display in the page. Click Return to return to the list of operations.

Importing all MDS Customizations

1. Navigate to the MDSAppRuntime management bean (MBean), as described in the section "Using the System MBean Browser and the MDSAppRuntime MBean".
2. Import metadata by selecting the importMetadata operation available from the Operations tab.
3. For fromLocation, enter the path of the directory or archive from which the documents will be imported.
Example: /tempDir/downloads/mdsExport/RetailAppsFrameworkTestMDS.zip
4. Click on the Invoke button, located at the upper-right corner of the page, to proceed with the import operation.
5. A confirmation message will display in the page. Click Return to return to the list of operations.

Importing a Specific Page Customization for a User

1. Navigate to the MDSAppRuntime management bean (MBean), as described in the section "Using the System MBean Browser and the MDSAppRuntime MBean".
2. Import metadata by selecting the **importMetadata** operation available from the Operations tab.
3. For **fromLocation**, enter the path of the directory or archive from which the documents will be imported.
Example: /tempDir/downloads/mdsExport/RetailAppsFrameworkTestMDS.zip
4. For **excludeBaseDocs**, select true. A Boolean value indicating whether to exclude base metadata documents from being imported.
5. For docs, click the pencil icon. On the Edit Parameter popup, click Add, and in the Element box, enter a list of comma-separated, fully qualified document names or document name patterns.
To import customizations for a specific page, simply enter the fully qualified base document name in the Element box.

Example:

```
/oracle/retail/apps/framework/uishell/skin/page/TestTablesAndTrees.jsff
```

6. You can provide the path to multiple documents to be imported, by clicking the Add button. When done, click on the OK button.
7. For restrictCustTo, click the pencil icon. On the Edit Parameter popup, click Add, and in the Element box, enter the list of customization layer names. This can be used to restrict the operation to only customization documents that match the specified customization layers.
Each customization layer name can contain, within a pair of brackets, optional customization layer values and value patterns separated by commas. Wildcards (%) may also be used for restricting the operation.
To import customizations for a particular user, enter "user[<username>]" in the Element box. See the table below for other examples.
For example:

Value	Description
user[buyer]	restricts operation to the user 'buyer'
user[buyer, betty_buyer]	restricts operation to the users 'buyer', and 'betty_buyer'
user[bu%]	restricts operation to user names that start with 'bu' (i.e. buyer)
user[be%]	restricts operation to user names that start with 'be' (i.e. betty_buyer)
user[%bu%]	restricts operation to users with 'bu' in the name (i.e. buyer and betty_buyer)

8. Click on the Invoke button, located at the upper-right corner of the page, to proceed with the import operation.
9. A confirmation message will display in the page. Click Return to return to the list of operations.

Creating Metadata Labels

1. Navigate to the MDSAppRuntime management bean (MBean), as described in the section "Using the System MBean Browser and the MDSAppRuntime MBean".

2. Create metadata label by selecting the createMetadataLabel operation available from the Operations tab.
3. For label, enter a valid name of the new metadata label to be created.
4. Click on the Invoke button, located at the upper-right corner of the page, to proceed with the operation.
5. A confirmation message will display in the page. Click Return to return to the list of operations.

Promoting Metadata Labels

1. Navigate to the MDSAppRuntime management bean (MBean), as described in the section "Using the System MBean Browser and the MDSAppRuntime MBean".
2. Promote metadata label by selecting the promoteMetadataLabel operation available from the Operations tab.
3. For label, enter a valid name of the new metadata label to be promoted. Promoting metadata labels can be used to roll back to an earlier version of the document, as captured by the label.
4. Click on the Invoke button, located at the upper-right corner of the page, to proceed with the operation.
5. A confirmation message will display in the page. Click Return to return to the list of operations.

Listing Metadata Labels

1. Navigate to the MDSAppRuntime management bean (MBean), as described in the section "Using the System MBean Browser and the MDSAppRuntime MBean".
2. List all metadata labels by selecting the listMetadataLabels operation available from the Operations tab.
3. Click on the Invoke button, located at the upper-right corner of the page, to proceed with the operation.
4. This will list all the available metadata labels previously created. You can use this, for example, to get the latest metadata label that you want to promote.
5. Click Return to return to the list of operations.

Deleting Metadata Labels

1. Navigate to the MDSAppRuntime management bean (MBean), as described in the section "Using the System MBean Browser and the MDSAppRuntime MBean".
2. Delete metadata label by selecting the deleteMetadataLabel operation available from the Operations tab.
3. For label, enter a valid name of the metadata label to be deleted.
4. Click on the Invoke button, located at the upper-right corner of the page, to proceed with the operation.

A confirmation message will display in the page. Click Return to return to the list of operations.

Security in Retail Applications

Retail Applications leverage ADF's security framework that is based on the Oracle Platform Security Services.

This section discusses the various assumptions around security for Retail Applications

SSO Setup for RAF Applications

RAF provides the following applications as enterprise archive (EAR) files to Retail Applications. By default, these applications are installing as part of Retail Applications

1. RetailAppsAdminConsole(RAAC)
2. RetailAppsMobileSecurity
 - a. RetailAppsMobileBasicAuth
 - b. RetailAppsMobileAccessService
3. RetailAppsRESTServices

In an SSO environment, follow the SSO setup procedure for these applications similar to Retail Applications.

Displaying External Application Contents in Non-SSO Environments

Retail Applications allow retailers to display content from external applications. These contents are typically business intelligence reports from a third party application that are configured to display within the Retail Application's dashboard.

Some of these contents might be secured requiring users to login before the contents can be accessed and displayed.

In non-SSO environments, when a user logs out of the Retail Application, they may not be logged out of any secured content they had configured access to. Therefore, it is highly recommended that customers only configure access to external content in a SSO-enabled environments where the application logout manages the logout from any other secured content that was previously accessed.

Web Services in Retail Applications

Retailers can access backend functionality in Retail Applications by calling the applications' ReSTful Web Services.

Refer to <http://www.oracle.com/technetwork/articles/javase/index-137171.html> to learn more about ReST as an architectural style applied to building web services.

Common Characteristics of Retail Application ReSTful Web Services

Deployment

A Retail Application can optionally package its ReST services as a separate Enterprise Archive (EAR) file. Installation of the ReST web services is a deemed choice of the customer and is controlled through an installer prompt.

Security

Services are secured using J2EE-based security model.

- Realm-based User Authentication. This verifies users through an underlying realm. The username and password is passed using Http Basic authentication.
- Role-based Authorization. This assigns users to roles, which in turn are granted or restricted access to resources/services. The authorization of ReSTful web services is static and cannot be reassigned to other rules post installation. The following role(s) is/are associated with ReSTful Web Services and should be added to the Enterprise LDAP:
 - BUYER
 - CORPORATE_INVENTORY_CONTROL_ANALYST
 - FINANCE_ANALYST
 - FINANCE_MANAGER
 - INVENTORY_ANALYST
 - INVENTORY_CONTROL_MANAGER
 - INVENTORY_MANAGER
 - RMS_APPLICATION_ADMINISTRATOR
 - RMS_DATA_STEWARD
 - SOURCING_ANALYST
 - SUPPLY_CHAIN_ANALYST

All enterprise roles defined above are mapped in web.xml and weblogic.xml of the ReST Service webapp.

The communication between the server and client is encrypted using one way SSL. In non-SSL environments the encoding defaults to BASE-64 so it is highly recommended that these ReST services are configured to be used in production environments secured with SSL connections.

Standard Request and Response Headers

Retail Application ReSTful web services have the following standard HTTP headers:

Accept: application/xml or application/JSON

Accept-Version: 15.0 (service version number)

Accept-Language: en-US,en;q=0.8

Accept-Versioning: False

Note: Specify 'Accept-Versioning: False' in the Service Request to bypass 'Accept-Version' validation on services.

Depending on the type of the operation or HTTP method, the corresponding response header is updated in the Http response with the following codes:

- GET/READ : 200
- PUT/CREATE : 201 created
- POST/UPDATE : 204
- DELETE : 204

Standard Error Response

Example response payload in case of service error is depicted below:

```
<?xml version="1.0" encoding="UTF-8" standalone="yes"?>
<messagesRDOes>
  <messagesRDO>
    <message>REST Service Version Mismatch</message>
    <messageType>ERROR</messageType>
    <status>BAD_REQUEST</status>
  </messagesRDO>
</messagesRDOes>
```

- message: The error message - translated.
- messageType: Value of 'ERROR' is returned.
- status : For a bad request or error, the status is BAD_REQUEST.
- The http error code for an error response is 400

List of ReSTful Web Services

Platform ReSTful Web Services

Notification ReST Services

Service endpoints have been enabled for the Notifications Framework. These endpoints are deployed separately and can be accessed by the following url:

http://server:port/RetailAppsReSTServices/services/private/Notifications/*

The endpoints exposed are detailed below.

Description	URL	HTTP Method
Creating Notifications	/Notifications/create	POST
Updating Notifications	/Notifications/update	PUT

Description	URL	HTTP Method
Deleting Notifications	/Notifications/delete/{id}	DELETE
Fetch Notifications	/Notifications/fetch?appCode={appCode}	GET
Get number of unread Notifications	/Notifications/fetch/unreadCount?appCode={appCode}	GET
Search Notifications	/Notifications/search?appCode={appCode}	GET
Filter Notifications – Return list of Notifications	/Notifications/filter/list	POST
Filter Notifications – Return grouped list of Notifications	/Notifications/filter/group	POST
Filter Notifications – Return a summarized list of Notifications	/Notifications/filter/summarize	POST
Count Notifications matching the filter	/Notifications/filter/count	POST
Persist the Criteria	/Notifications/criteria	POST
Fetch the Criteria	/Notifications/fetch/criteria?appCode={appCode}	GET
Fetch Recipients	/Notifications/fetch/recipients/{id}	GET
Fetch Notification Context	/Notifications/fetch/context/{id}	GET
Fetch Notification Time Periods	/Notifications/fetch/timeperiods?appCode={appCode}	GET
Fetch Notification Hierarchy Levels	/Notifications/fetch/hierarchylevels?appCode={appCode}	GET

Description	URL	HTTP Method
Fetch Notification Types	/Notifications/fetch/notificationtypes?appCode={appCode}	GET
Status update for multiple Notifications	/Notifications/update/multiple/status	PUT
Delete multiple Notifications	/Notifications/delete/multiple	POST

Access Control ReST Service

Description	URL	HTTP Method
FetchFilteredRoles	/AccessRoles	POST

Favorite ReST Services

Description	URL	HTTP Method
Fetch all Favorites	/Favorites/fetch	GET
Add a Favorite	/Favorites/add	POST
Delete a Favorite	/Favorites/delete	DELETE
Update a Favorite	/Favorites/update	POST
Replace a Favorite	/Favorites/replace	POST

RMS ReST Services

Description	URL	HTTP Method
RMS Common Services	/Common/	
Fetch VDate	/Common/vDate	GET
Fetch Currencies	/Common/Currencies	GET
Fetch Departments	/Common/departmentSearch?searchString={searchString}&pageSize={pageSize}&pageNumber={pageNumber}	GET
Refresh Departments Attributes	/Common/departmentLoad?departments={departments}	GET
Fetch Procurement System Options	/Common/POSysOps	GET
Fetch Functional Config System Options	/Common/FuncSysOps	GET
Fetch Currencies	/Common/Currencies	GET

Description	URL	HTTP Method
Fetch Inventory Movement System Options	/Common/InvMovSysOps	GET
Transfer Services	/Transfer	
Fetch Next Transfer Number	/Transfer/transferId	GET
Fetch Items for a given Inventory Transfer	/Transfer/item?itemSearchType={itemSearchType}&searchString={searchString}&dept={dept}&fromLocation={fromLocation}&pageSize={pageSize}&pageNumber={pageNumber}	GET
Refresh Items for a given Inventory Transfer	/Transfer/item/load?items={items}&fromLocation={fromLocation}	GET
Fetch From Location	/Transfer/fromLocation?locationType={locationType}&searchString={searchString}&tsfType={tsfType}&toLocation={toLocation}&items={items}&pageSize={pageSize}&pageNumber={pageNumber}	GET
Fetch To Location	/Transfer/toLocation?locationType={locationType}&searchString={searchString}&tsfType={tsfType}&fromLocation={fromLocation}&pageSize={pageSize}&pageNumber={pageNumber}	GET
Refresh Location	/Transfer/loadLocations?fromLocation={fromLocation}&toLocation={toLocation}	GET
Create Transfer	/Transfer	POST
Recent Transfer Services	/Transfer/recent/	
Fetch Transfer Type	/Transfer/recent/transferTypeList	GET
Fetch Transfer Status	/Transfer/recent/transferStatusList	GET
Fetch Location	/Transfer/recent/transferLocSearch?searchString={searchString}&locType={locType}&pageSize={pageSize}&pageNumber={pageNumber}	GET
Refresh Location	/Transfer/recent/transferLocationLoad?locations={locations}	GET
Fetch Transfer User IDs	/Transfer/recent/searchUserIds?searchString={searchString}&pageSize={pageSize}&pageNumber={pageNumber}	

Description	URL	HTTP Method
Transfer Search	/Transfer/recent/transferSearch?status={statuses}&transferTypes={transferTypes}&createIds={createIds}&startCreateDate={startCreateDate}&endCreateDate={endCreateDate}&startDeliveryDate={startDeliveryDate}&endDeliveryDate={endDeliveryDate}&transferNumber={transferNumber}&locations={locations}&departments={departments}&pageSize={pageSize}&pageNumber={pageNumber}	GET
Fetch Transfer Detail	/Transfer/recent/transferDetail?transferNumber={transferNumber}&pageSize={pageSize}&pageNumber={pageNumber}	GET
Update Transfer Status	/Transfer/recent/updateTransferStatus?newStatus={newStatus}&transferNumbers={transferNumbers}	POST
Purchase Orders Services	/PurchaseOrders	
Fetch Locations	/PurchaseOrders/location?locationType={locationType}&searchString={searchString}&supplier={supplier}&orgUnitId={orgUnitId}&pageSize={pageSize}&pageNumber={pageNumber}	GET
Refresh Locations	/PurchaseOrders/location/load?locations={locations}&supplier={supplier}	GET
Fetch Items	/PurchaseOrders/item?itemSearchType={itemSearchType}&searchString={searchString}&dept={dept}&supplier={supplier}&locations={locations}&pageSize={pageSize}&pageNumber={pageNumber}	GET
Refresh Items	/PurchaseOrders/item/load?item={item}&supplier={supplier}&locations={locations}	GET
Fetch Supplier	/PurchaseOrders/supplier?supplierSearchString={supplierSearchString}&locations={locations}&items={items}&pageSize={pageSize}&pageNumber={pageNumber}	GET
Refresh Supplier	/PurchaseOrders/supplier/load?suppliers={suppliers}&locations={locations}&items={items}	GET
Fetch Supplier Terms	/PurchaseOrders/supplier/terms	GET
Fetch Next Order Number	/PurchaseOrders/order/id	GET
Create Purchase Orders	/PurchaseOrders	POST
Recent Purchase Orders	/PurchaseOrders/recent	

Description	URL	HTTP Method
Fetch Origin Code	/PurchaseOrders/recent/originCodeList	GET
Fetch PO Status Codes	/PurchaseOrders/recent/purchaseOrderStatusList	GET
Searching Recent Purchase Orders	/PurchaseOrders/recent/purchaseOrderSearch?statuses={statuses}&createIds={createIds}&startCreateDate={startCreateDate}&endCreateDate={endCreateDate}&orderNumber={orderNumber}&suppliers={suppliers}&originCodes={originCodes}&departments={departments}&pageSize={pageSize}&pageNumber={pageNumber}	GET
Purchase Order Summary	/PurchaseOrders/recent/PurchaseOrderSummary?orderNumber={orderNumber}	GET
Fetch Purchase Order Items	/PurchaseOrders/recent/PurchaseOrderItems?orderNumber={orderNumber}&itemDisplayLevel={itemDisplayLevel}&pageSize={pageSize}&pageNumber={pageNumber}	GET
Fetch Purchase Order Item Locations	/PurchaseOrders/recent/PurchaseOrderItemLocations?orderNumber={orderNumber}&item={item}&itemDisplayLevel={itemDisplayLevel}&diff1={diff1}&diff2={diff2}&diff3={diff3}&diff4={diff4}&pageSize={pageSize}&pageNumber={pageNumber}	GET
Fetch Purchase Orders User IDs	/PurchaseOrders/recent/searchUserIds?searchString={searchString}&pageSize={pageSize}&pageNumber={pageNumber}	GET
Fetch Cancel Reason Codes	/PurchaseOrders/recent/cancelReasonCodeList	GET
Update Purchase Orders Date	/PurchaseOrders/recent/updatePurchaseOrderDate?notBeforeDate={notBeforeDate}¬AfterDate={notAfterDate}&otbEowDate={otbEowDate}orderNumbers={orderNumbers}	POST
Canceling Purchase Orders	/PurchaseOrders/recent/cancelPurchaseOrders?orderNumbers={orderNumbers}&reasonCode={reasonCode}	POST
Rejecting Purchase Orders	/PurchaseOrders/recent/rejectPurchaseOrders?orderNumbers={orderNumbers}	POST
Approving Purchase Orders	/PurchaseOrders/recent/approvePurchaseOrders?orderNumbers={orderNumbers}	POST

Customization of Retail Applications

This section discusses key concepts, guidelines, and supported use cases for customizing a Retail Application.

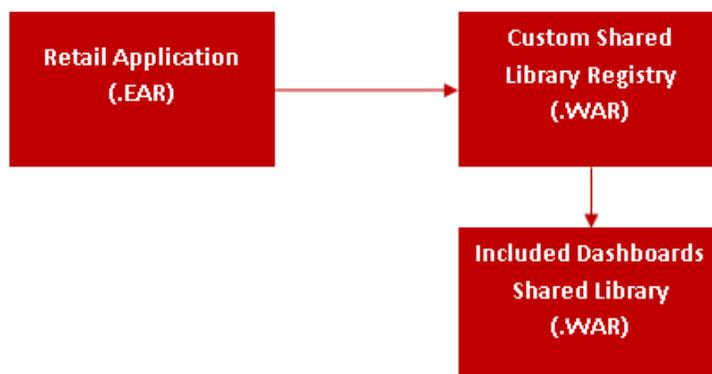
Prerequisite Concepts

This section contains concepts retailers **must** understand before performing any of the supported customization scenarios discussed in this document.

Understanding the Deployment of Retail Applications

In order to customize the Retail Application, retailers must understand the deployment of the Retail Application.

A Retail Application can install one or more application artifacts into a Weblogic server instance. The diagram below shows a typical installation of a Retail Application:



The Retail Application is deployed as a standalone application and deployed using an EAR file.

Two shared libraries are typically installed along with the application EAR:

- The Included Dashboards Shared Library
- The Custom Shared Library Registry

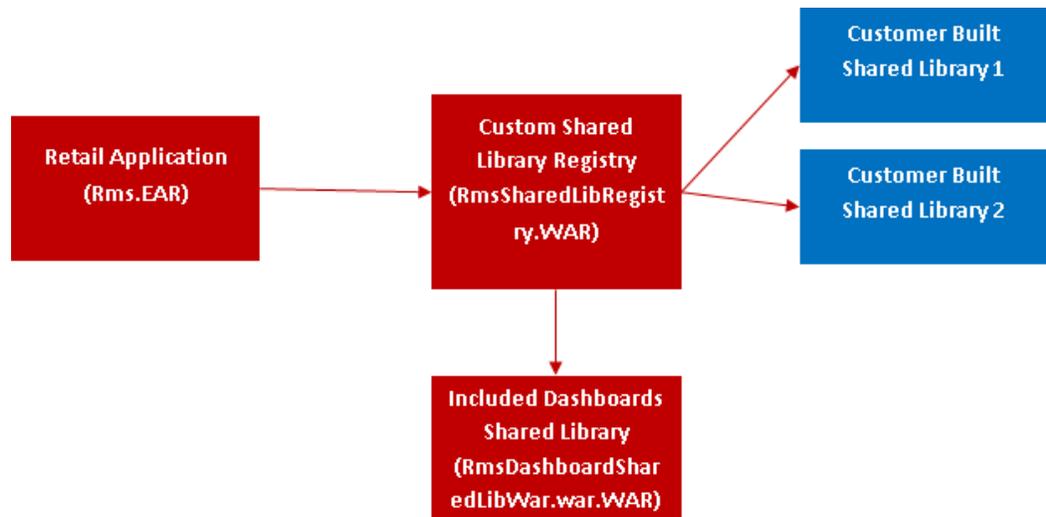
The Included Dashboards Shared Library contains the Retail Application-built dashboards. These dashboards can be customized. Further discussion on Retail

Application Included Dashboards can be found in the section, “Understanding Dashboards in Retail Applications.”

A Retail Application that allows for customization will have as part of its installation an intermediary shared library that will serve as a registry to reference the actual retailer-built shared libraries. This is called the Custom Shared Library Registry.

When retailers need to add their own content into the application, metadata to register their content into the application’s UI including the binaries for the content itself (for example, task flows and pages) is expected to be packaged into a Web Archive (WAR) file and deployed as a shared library in the same managed server as the application itself.

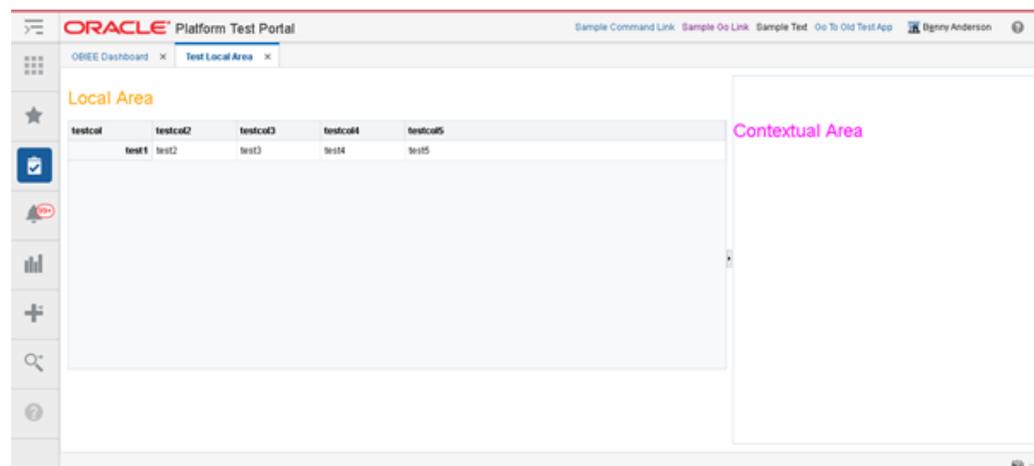
Then, the names of these shared libraries have to be referenced in the Custom Shared Library Registry.



Understanding the Retail application User Interface

Retail Applications allow retailers to add and display custom content such as new workflows and dashboards. These custom contents can be accessed from the Retail Application’s user interface (UI).

The Retail Application user interface organizes the contents into visual containers that fulfill common layout and navigational requirements in a structured, consistent manner.



The high-level containers or areas in the UI are highlighted below:

- Global Area
- Navigation Pane and Sliding Sidebar Menu
- Local Area
- Contextual Area



The Local Area is the main content area of the application. Workflows that allow users to get tasks done in the application are presented in this area. The workflows are launched into this area as a result of actions the user take in the navigation pane, sliding sidebar panel and global area.

The Navigation Pane organizes the different actions a user can take on the application as iconic menu items. The options or actions available to the users for each menu item is presented in the Sliding Sidebar panel next to the Navigation Pane.

There are five (5) standard Navigation Pane menus that are available to users:

- Application Navigator/Switcher-This menu allows users to switch between Retail Applications.
- Favorites-This menu allows users to access their own favorite list of actions or tasks within the application.
- Tasks-This menu presents a hierarchy of tasks or actions the user can access in the application.
- Reports- This menu presents a hierarchy of links to the application reports and dashboards.
- Notification-This menu allows users to access unread and past notifications around business processes that are happening in the application.

The Global Area of the application contains branding information on the left hand side and application wide menu options on the right hand side. Application wide menu options include access to application preferences, login and logout, application help and application information.

The Contextual Area is a collapsible area on the right of the page, which provides space to present information that can assist users in completing their tasks. The Contextual Area is presented per Local Area tab. Each task flow is presented in the local area can have a contextual area

Supported Customization Scenarios

Adding a Custom Shared Library

As discussed in the section, “Understanding the Deployment of Retail Applications,” a Retail Application that allows for customization will have as part of its installation an intermediary shared library that will serve as a registry to reference the actual retailer-built shared libraries. This is called the Custom Shared Library Registry.

This section contains steps retailers can follow to create a custom shared library that will contain new custom made content, reference that shared library from the registry, and deploy it to the server.

Download JDeveloper

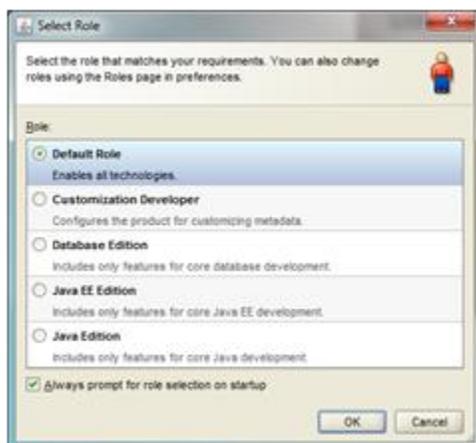
To create the custom shared library, it is recommended that retailers download and install JDeveloper version 12.2.1 by following the link below:

<http://www.oracle.com/technetwork/developer-tools/jdev/downloads/index.html>

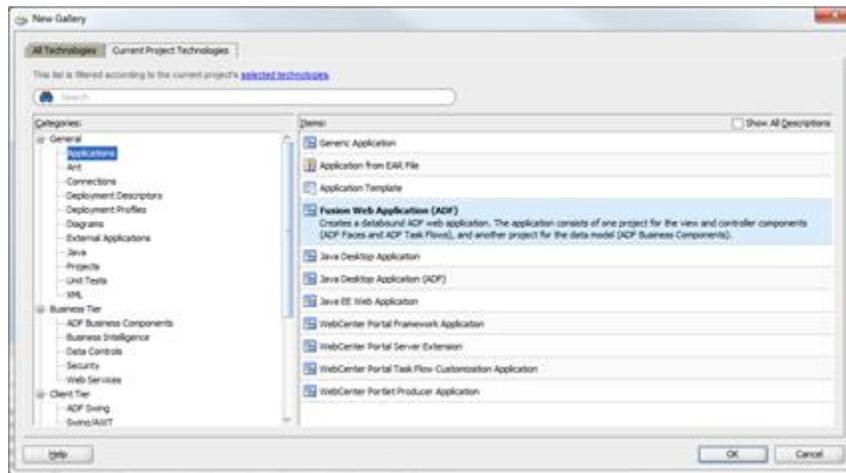
Create the Custom Shared Library Workspace through JDeveloper

This section describes the steps to create and configure the Custom Shared Library Workspace in JDeveloper, which will house the code for any custom content the retailer wants to add into the Retail Application. Through JDeveloper, a shared library web application archive file (.WAR) can be generated which can be deployed in the same managed server as the Retail Application.

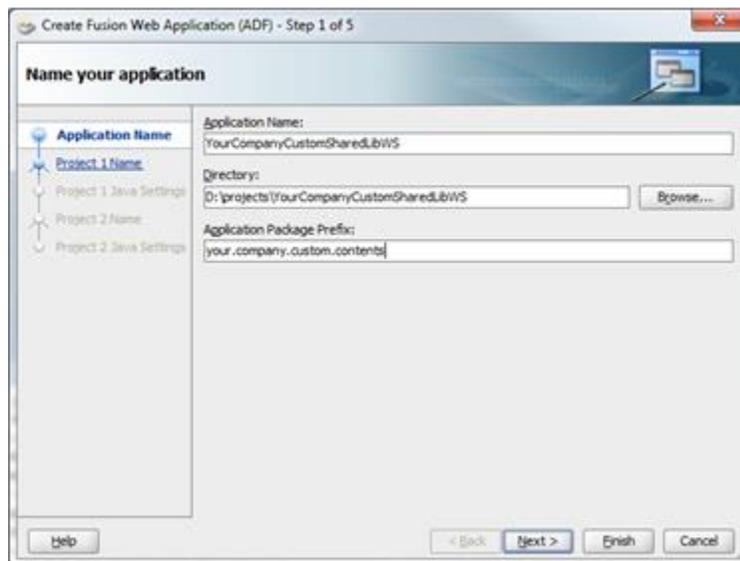
1. Open JDeveloper and choose Developer Role when prompted.



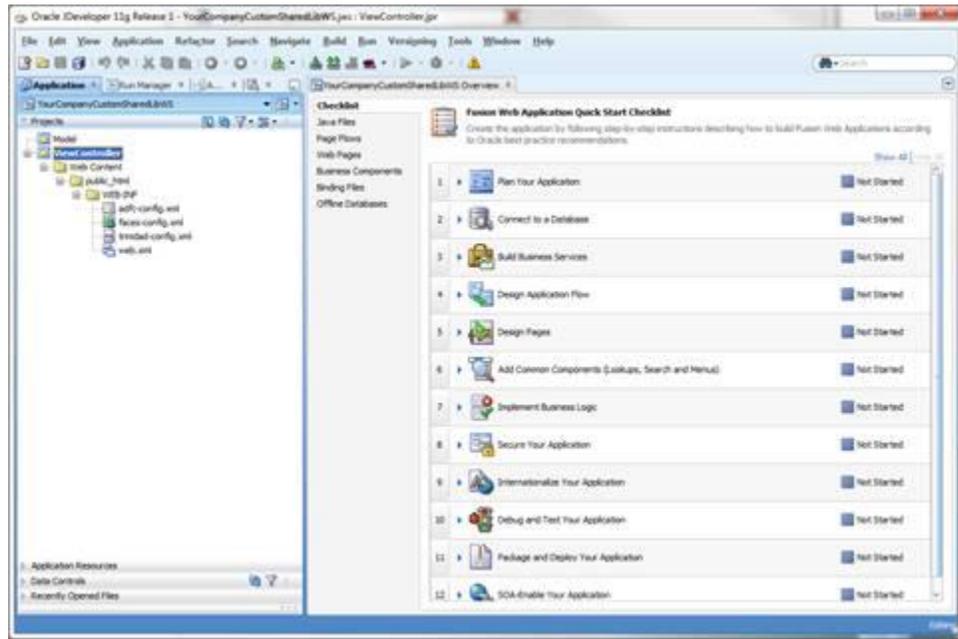
2. Create a new Fusion Web Application JDeveloper workspace.
 - a. Go to File *New* to invoke the *New Gallery* dialog. Choose *Fusion Web Application (ADF)* as the type of application to create and click OK.



- b. Provide a meaningful application name, a directory path to the workspace and an application (Java) package prefix and click Finish

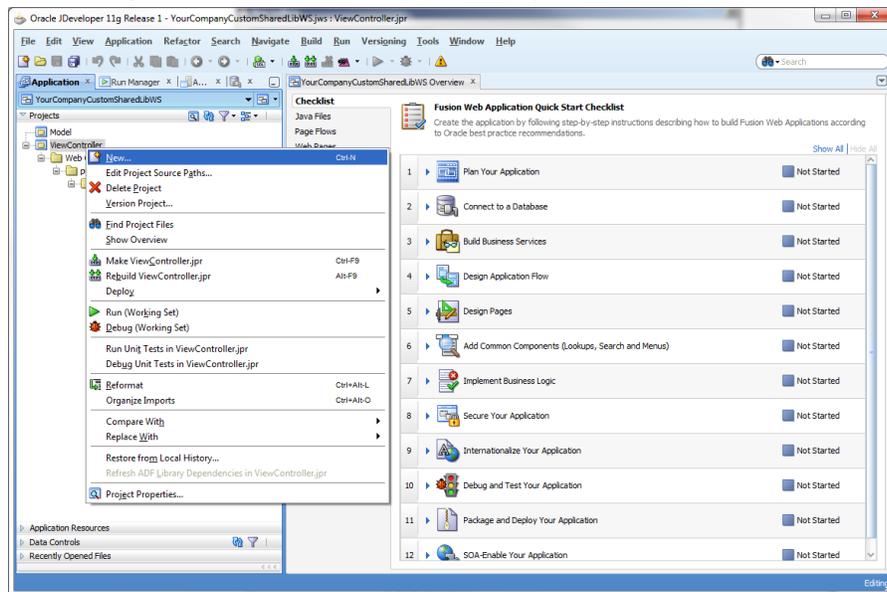


- c. .JDeveloper will generate a new workspace with two projects: *Model* and *View-Controller*.

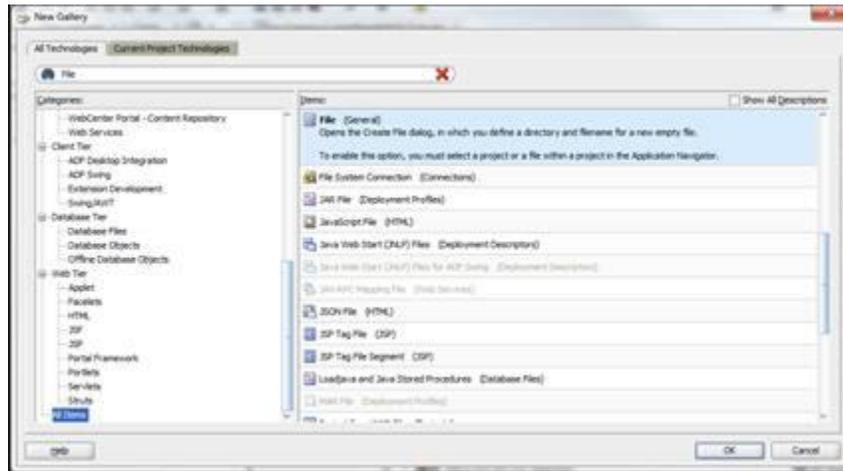


- 3. Add a Manifest file containing the name of the shared library.

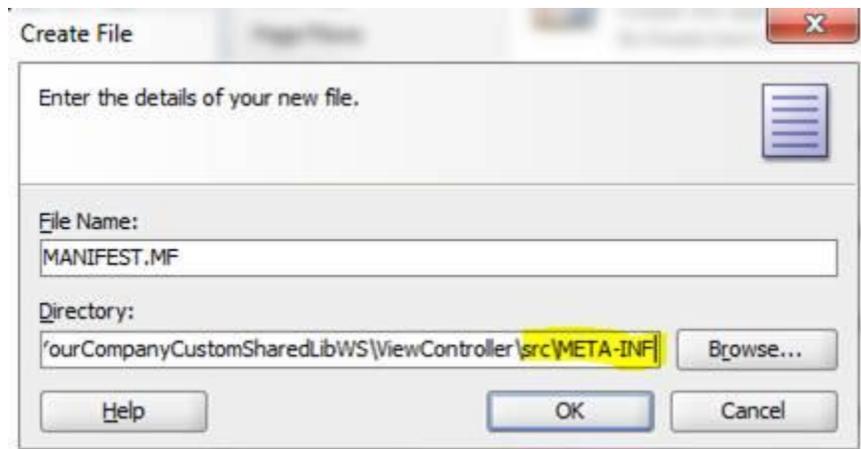
- a. Right-click on the *View-Controller* project and choose *New* from the context menu.



- b. The *New Gallery* dialog appears. Choose the option, *File (General)*, in the *All Technologies* section of the dialog. Click OK.



- c. The *Create File* dialog opens. For the *File Name*, specify `MANIFEST.MF`. For the *Directory*, the new file **must be added under the `src/META-INF` subdirectory** under the *View-Controller* project's directory.



- d. Edit the new `MANIFEST.MF` file and add the following entries:

```
Manifest-Version: 1.0
Implementation-Vendor: companyName
Implementation-Title: Custom Shared Library for companyName
Implementation-Version: 1.0
Extension-Name: companyname.custom.shared.lib
Specification-Version: 1.0
Created-By: companyName
```

Modify the contents such that meaningful and unique values are used for `Implementation-Vendor`, `Implementation-Title`, `Extension-Name` and `Created-By`. Example:

Manifest-Version: 1.0

Implementation-Vendor: Acme Retail

Implementation-Title: Custom Shared Library for Acme Retail

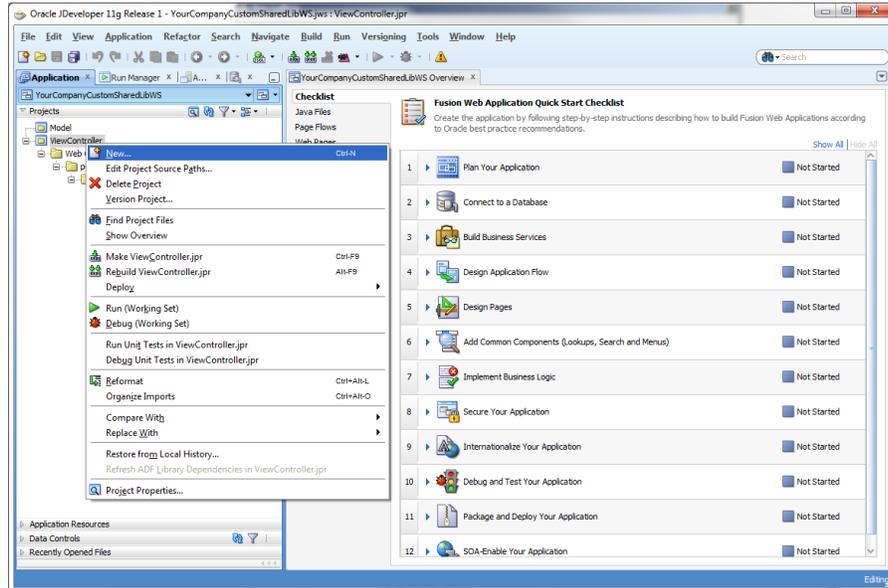
Implementation-Version: 1.0

Extension-Name: acmeretail.custom.shared.lib.procurement

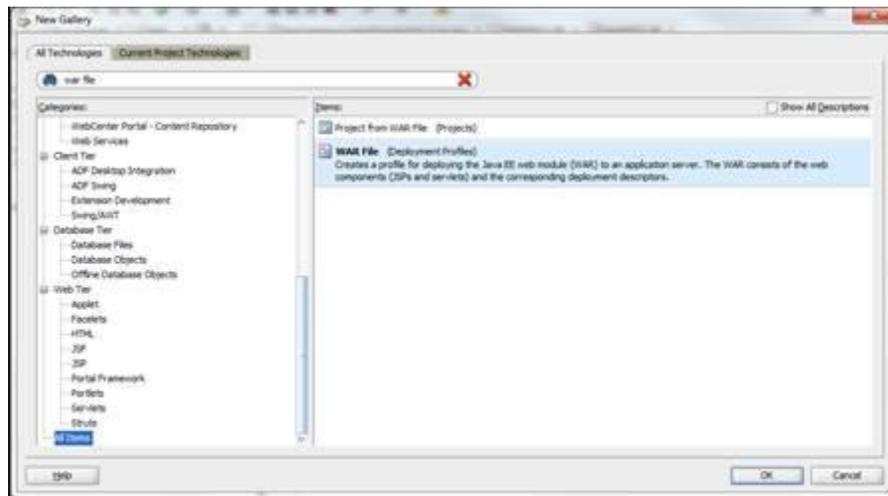
Specification-Version: 1.0

Created-By: AcmeRetail

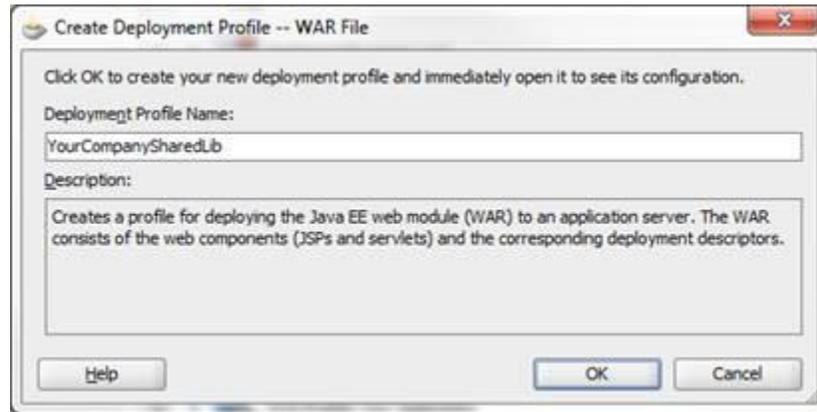
4. Create a deployment profile for the shared library.
 - a. Right-click on the *View-Controller* project and choose *New* from the context menu.



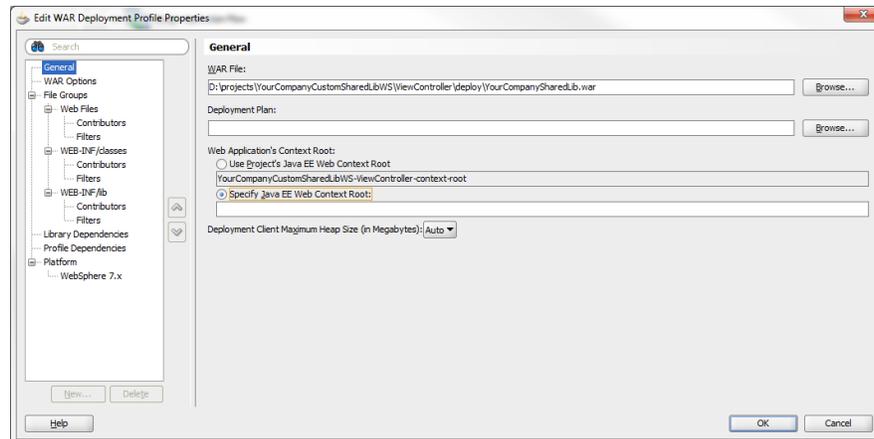
- b. The New Gallery dialog appears. Choose the option, WAR File (Deployment Profiles), in the All Technologies section of the dialog. Click OK



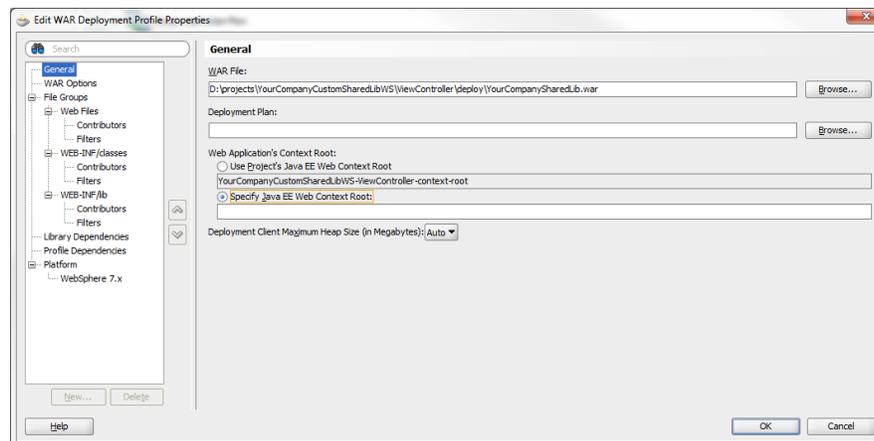
- c. Provide a unique and meaningful name for the Deployment Profile and click OK.



- d. The Edit WAR Deployment Profile Properties dialog is shown.



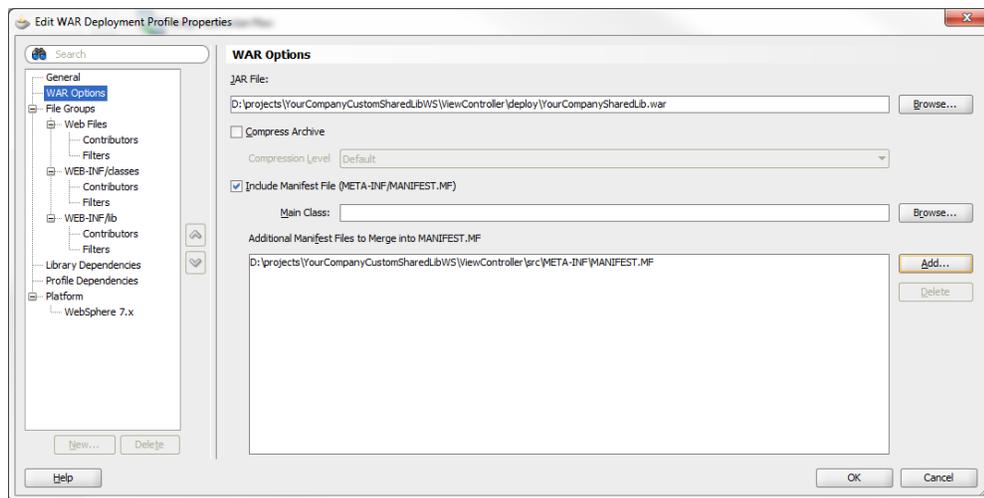
- e. Under the General section, make sure that the Specify Java EE Web Context Root is selected without any value.



When you navigate away from this section – you will be prompted to confirm that you really want a blank context root. Respond Yes to the confirmation dialog.



- f. Under the WAR Options section, enable the Include Manifest File option and add the MANIFEST.MF file you created under .../View-Controller/src/META-INF.



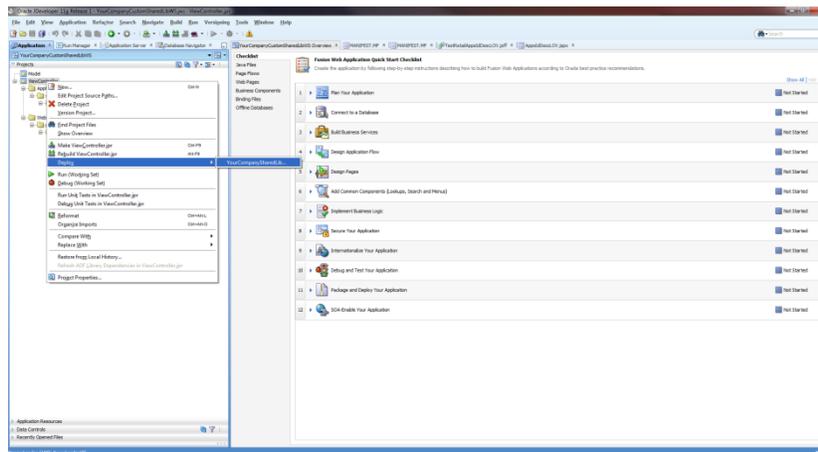
- g. Click OK.

Generate and Deploy the Custom Shared Library WAR

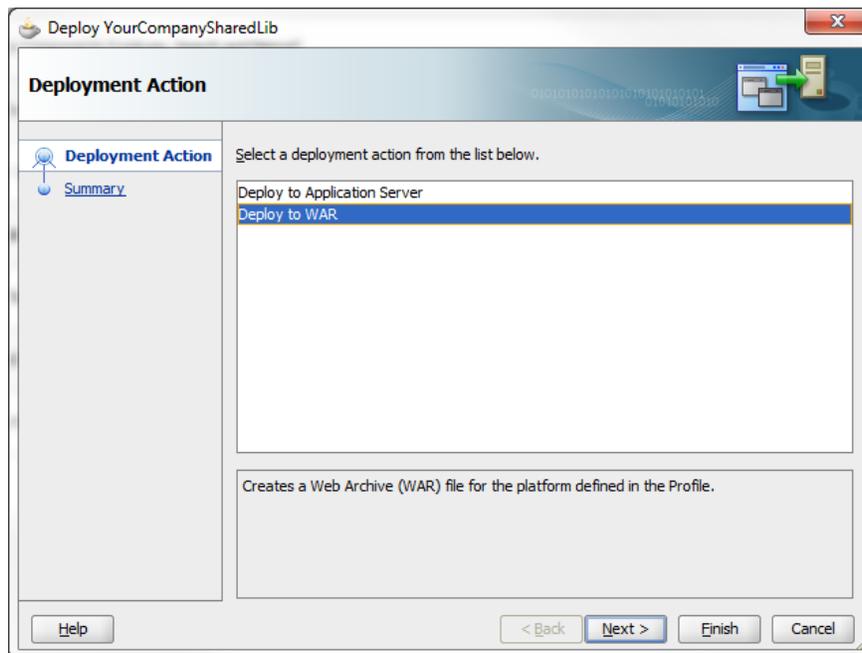
This section describes the steps to generate and deploy the Custom Shared Library WAR file.

1. Generate the share library WAR file through JDeveloper.
 - a. Open the Custom Shared Library workspace.

- b. Right-click on the *View-Controller* project and choose the shared library WAR deployment profile created previously under the *Deploy* option.



- c. The *Deploy* dialog opens. Select *Deploy to WAR* and click *Finish*.



- d. JDeveloper will generate the WAR file into the *View-Controller* project folder subdirectory, *deploy*.

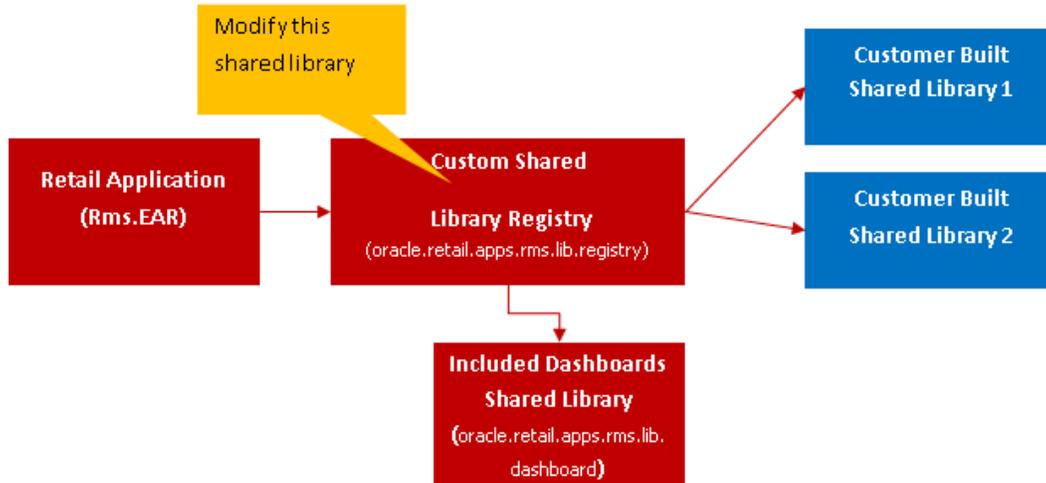


2. Deploy the generated WAR file to the same managed server as the Retail Application as a shared library. Refer to section 6 of the Deploying Applications to Oracle WebLogic Server documentation at this location:
<http://docs.oracle.com/middleware/1221/wls/DEPGD/deploy.htm#DEPGD212>

This task has to be performed using the WebLogic Administration Console or Enterprise Manager Fusion Middleware Control with a user having WebLogic administration permissions.

Reference the Custom Shared Library from the Retail Application

Once the retailer has created and deployed its own custom shared library, the retailer needs to modify the configuration of the Custom Shared Library Registry to add references to the retailer's shared library.



1. Log into the WebLogic Administration Console as a user with administrative permissions.
2. If the administration console was configured with domain configuration locking, go ahead and click on the Lock & Edit button to ensure that other administrators can be prevented from making changes during your edit session.



3. Navigate to the Deployments section.

The screenshot shows the Oracle WebLogic Server Administration Console. The main content area is titled "Summary of Deployments" and contains a table of deployed applications. The table has the following columns: Name, State, Health, Type, and Deployment Order. The table lists several applications, including libraries and enterprise applications, with their respective states and health indicators.

Name	State	Health	Type	Deployment Order
oracle.oracle.businesseditor(1.0,11.1.1.2.0)	Active		Library	100
oracle.oracle.domain(1.0,11.1.1.2.0)	Active		Library	100
oracle.oracle.domain.webapp(1.0,11.1.1.2.0)	Active		Library	100
alloc.customer.sample.extension	Active		Library	100
AllocAsyncMessageConsumer (V2.0)	Active	OK	Enterprise Application	100
content-app-ib(11.1.1.11.1.1)	Active		Library	300
content-web-ib(11.1.1.11.1.1)	Active		Library	300
OHS Application (11.1.1.1.0)	Active	OK	Web Application	5
EarAICCommonPortal (V2.0)	Active	OK	Enterprise Application	100
EarAICCommonPortal-help	Active	OK	Enterprise Application	100
EarAICCommonPortalmc (V2.0)	Failed		Enterprise Application	100
em	Active	OK	Enterprise Application	400
email	Active		Library	100
emas	Active		Library	100
emcore	Active		Library	100
FMW Welcome Page Application (11.1.0.0.0)	Active	OK	Enterprise Application	5
java-framework-web-ib(10.3.2,10.3.2)	Active		Library	300
jersey-web-ib(1.0,1.0.2)	Active		Library	300
lib(1.2,1.2.9.0)	Active		Library	100

4. Look for the Retail Application deployment and shut it down. Choose “Force Stop Now” when appropriate. Wait for the shutdown process to complete.

5. Get the deployment location of the Retail Application’s custom shared library registry. Under the deployments list, click on the link for the library named oracle.retail.apps.ac.portal.extensions(16.0,16.0). The Settings page for the library appears.

The screenshot shows the Oracle WebLogic Server Administration Console, specifically the Settings page for the library named "oracle.retail.apps.ac.portal.extensions". The page displays various configuration fields and their descriptions.

Settings for oracle.retail.apps.ac.portal.extensions

Overview | Targets | Notes

Click the **Lock & Edit** button in the Change Center to modify the settings on this page.

Use this page to view, and sometimes change, general configuration information about the Java EE library, such as its name, path to the source of the application, and staging mode. Use the **Deployment Order** field to change the order that the library is deployed at server startup, relative to other deployments.

Name: oracle.retail.apps.ac.portal.extensions. The name of this Java EE library. [More Info...](#)

Path: /scratch/u01/webadmin/product/fmw/wls_wlsc/user_projects/domains/APFDomain/servers/AdminServer/upload/oracle.retail.apps.ac.portal.extensions/app/WarAICPortalExtensionsSharedLibrary.war. The path to the source of the deployable unit on the Administration Server. [More Info...](#)

Staging Mode: stage. The mode that specifies whether an application's files are copied from a source on the Administration Server to the Managed Server's staging area during application preparation. [More Info...](#)

Deployment Order: 100. An integer value that indicates when this unit is deployed, relative to other deployable units on a server, during startup. [More Info...](#)

Principal Name: . A string value that indicates what principal should be used when deploying the file or archive during startup and shutdown. This principal will be used to set the current subject when calling out into application code for interfaces such as ApplicationLifecycleListener. If no principal name is specified, then the anonymous principal will be used. [More Info...](#)

Applications that reference this Library

Name	Type
EarAICCommonPortal (V2.0)	Enterprise Application

The Settings page illustrates the file location of the registry’s WAR file under the Path entry. Make note of this file location.

6. Using the operating system's file manager application, go to the location of the WAR file. You will need read and write permissions to the file system where the WAR file is located.

7. Make a copy of the WAR file as back-up.
8. Open the original WAR file using an archive file manager and update the /WEB-INF/weblogic.xml by adding a new <library-ref> entry pointing to your custom shared library.

```
<library-ref>  
<library-name>companyname.custom.shared.lib</library-name>  
</library-ref>
```

Note: The library-name has to match the Extension-Name you provided in your custom shared library's MANIFEST.MF file. Once this change is done, you have now linked your custom shared library to the Retail Application.

9. Return to the WebLogic Administration Console. Go to Environments' Servers section. Under the Control tab, select the managed server where Retail Application is deployed to. Shut it down and start it back up again.

Creating New ADF Contents

Custom ADF and Java based components can be coded into the Custom Shared Library workspace. Typically, retailers can include task flows that add new UI workflows into the Retail Application UI. These task flows may implement :

- A new workflow to maintain the Retail Application's data.
- A dashboard page to highlight key performance indicators and important metrics based on the Retail Application's data.

This section contains important points that developers must consider when building new custom ADF-based contents for Retail Applications.

Custom Shared Library must be Regenerated

Custom ADF content must be implemented within the custom shared library workspace. Retailers must create a new custom shared library workspace if none exists.

The WAR artifact for the custom shared library must be regenerated and redeployed.

Refer to the section, Adding a Custom Shared Library.

New Components should have Security Grants

Since Retail Applications are secured web applications, the addition of new ADF UI pages and task flows will need to be secured using the application roles and policies that are recognized by the Retail Application. The provisioning process is done using the Oracle Enterprise Manager Fusion Middleware Control console.

Refer to Section 10, Managing Policies of the Oracle® Fusion Middleware Securing Applications with Oracle Platform Security Services for details (<https://docs.oracle.com/middleware/1221/opss/JISEC.pdf>).

ADF Best Practices Must Be Applied

Developers will be building components in both Model and View-Controller layers using the tools and standards available for Oracle ADF. New ADF screens will have their own business components, task flows, pages, and bindings. These components must comply to published best practices and standards for that technology.

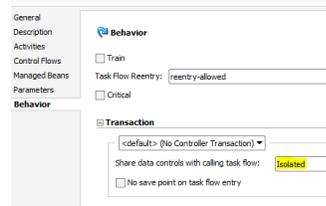
Refer to the Oracle JDeveloper and Oracle ADF Technical Resources web site (<http://www.oracle.com/technetwork/developer-tools/jdev/learnmore/index.html>) .

Of particular interests are coding guidelines published under the ADF Architecture Square section of the site (<http://www.oracle.com/technetwork/developer-tools/adf/learnmore/adfarchitect-1639592.html>).

Task Flow and Page Configuration Must be Supported

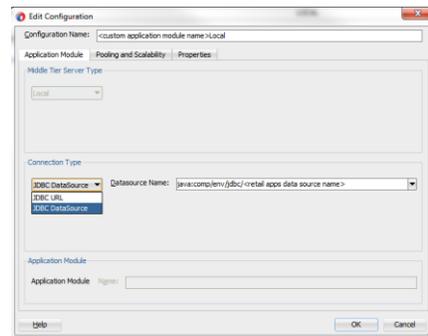
Task flows must be built as bounded task flows using JSF page fragments.

The task flow must have an isolated data control scope. This is configured in JDeveloper's task flow designer by going to the Behavior tab and setting data control share option as "Isolated".



The Same Data Source Must Be Used

New application modules must be configured to use Data Sources and not JDBC URLs when accessing the database layer. This can be configured in JDeveloper's application module designer.



The data source name must be the same as the Retail Application's primary data source name(jdbc/RmsDBDS).

Adding or Modifying an Item in the Reports Menu

To add new contents in the Retail Application's Reports Menu, the retailer must override the reports menu model.

All items seen by the user in the Reports Menu are registered in an XML file called the Reports Menu Model.

This file is kept in the application's EAR file but can be extracted, copied and modified so retailers can add references to their own dashboards or reports.

To add or modify an item in the Reports Menu, follow the steps below:

1. Add a custom shared library. See the section, "Adding a Custom Shared Library" for details.
2. Using JDeveloper, open the Custom Shared Library workspace in Developer Role.
3. Add new custom reports as required.

4. Regenerate the shared library WAR file from the workspace and redeploy the shared library. Shut down and restart of the Retail Application and its shared library registry is required.
5. Changes Required in Shared Library Registry for Custom Reports Menu Model.
 - a. Obtain the application's Reports menu model XML file.

Tasks Menu Model XML	Ear path is RmsViewController.war\WEB-INF\classes\oracle\retail\apps\rms\view\uishell\config\custom\HomeSidebarNavigationTreeModel.xml
----------------------	--

- b. Refer the Section "Reference the Custom Shared Library from the Retail Application," to register the custom reports menu model shared library within the Retail Application.
 - c. Open the original WAR file using an archive file manager and copy the above Reports menu model xml file in View-Controller project src directory, preferably under a subdirectory called custom.
 - d. Add a new or modify existing file called PageTemplateOverrideModel.properties under the View-Controller/src directory.
 - e. Modify this file and add the following entry:
`Home.taskMenuModelXml=<path to sidebar model xml within view-controller/src>`
 Example:
`Home.taskMenuModelXml=/custom/HomeTaskMenuModel.xml`
 - f. Modify the copy of the model. Add new items, which are created in the shared library project. Refer to the section, Reports Menu Model XML Items.
6. Test the Retail Application.

Dashboard Customization Scenarios

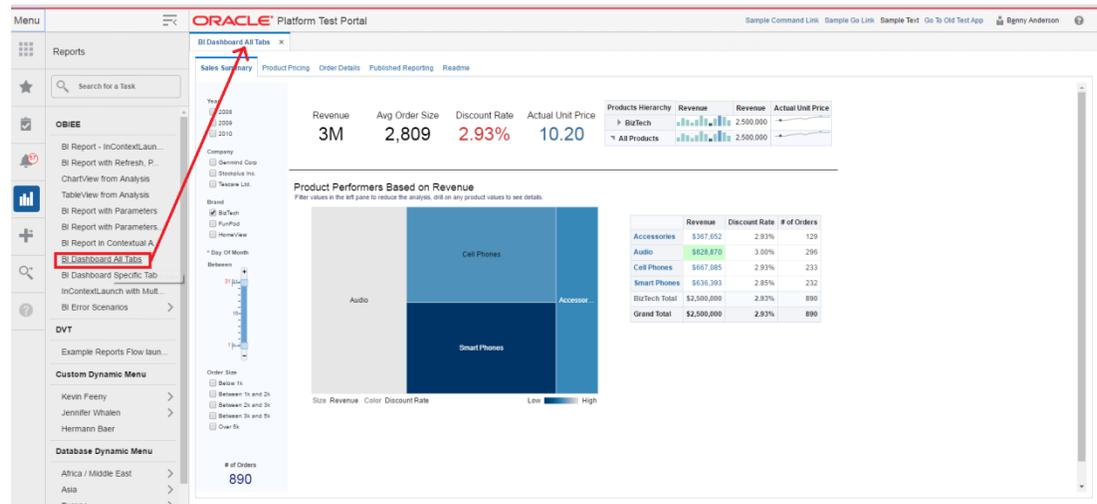
Understanding Dashboards in Retail Applications

A dashboard is a page within a Retail Application that displays the status of metrics and key performance indicators relevant to the Retail Application. They are typically tailored for specific roles and they allow users to easily monitor the status of the current data within the application.

Users can access dashboards from the application's Reports menu.

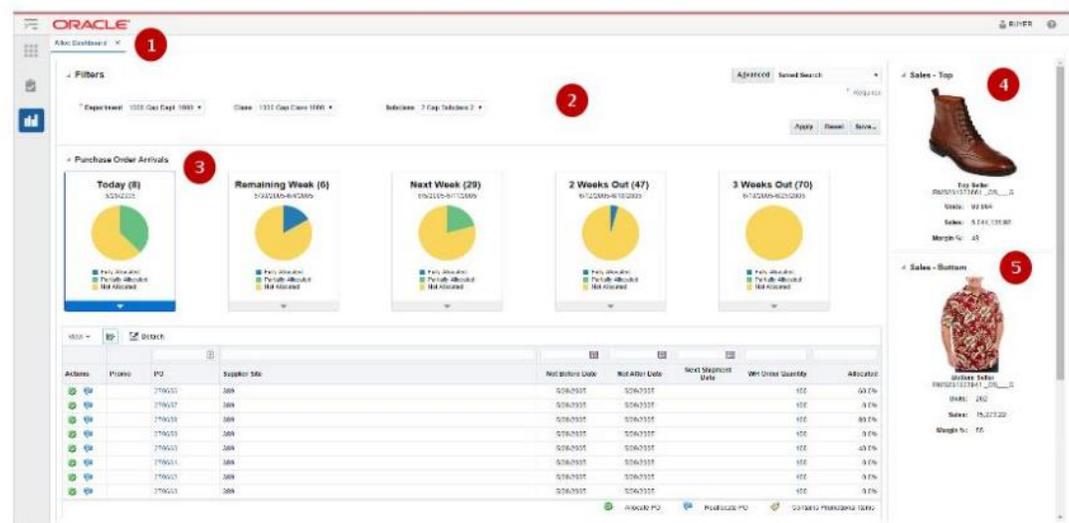


The example below shows the dashboard page as rendered in the local area of a Retail Application's UI.



Anatomy of a Dashboard

Depicted below is an example of a dashboard that might appear in a Retail Application



1	Dashboard Page	A page rendering a dashboard. A dashboard is composed of 1 or more reports. The dashboard page is launched as a tab within the Retail Application
2	Prompt Region	The prompt region contains a set of editable input components that allow users to modify the parameters of the various reports in the dashboard. This allows users to see different sets of data for the same set of reports. This is optional on a dashboard page. There can be multiple prompt regions in a dashboard affecting different sets of reports on a dashboard.
3,4,5	Report Regions	A report region shows a graphical view of the application's business data.

Supported Implementation of Dashboards

Retail Applications support the following implementation of dashboards:

- Dashboards built using Oracle ADF.
- Dashboards built in external BI reporting tools and accessible via a web browser URL.
- Dashboards built using Oracle ADF is the recommended implementation of dashboards as they have seamless integration with the rest of the Retail Application.
- Dashboards built in external BI reporting tools are supported as long as the dashboards can be accessed in a web browser via a URL. Oracle Retail recommends using Oracle Business Intelligence Enterprise Edition (OBIEE) for URL based dashboards.

Retail Application Included Dashboards

As mentioned in the section, Understanding the Deployment of Retail Applications, a Retail Application is installed with a shared library containing included dashboards that are rendered by default within the application.

These dashboards have the following characteristics:

- The dashboards are implemented in Oracle ADF for seamless integration with the rest of the application workflows.
- The dashboards are implemented using a common framework called ADF Dashboard Prompts to allow for a standard way of customizing them.

Retailers can customize included dashboards.

See the section, "Dashboard Customization Scenarios," for details on the types of customization retailers can perform on included dashboards.

Adding a New ADF-based Dashboard in the Reports Menu

This section describes the steps for adding ADF based dashboard reports into the Retail Application's UI Sidebar. ADF-based reports can be integrated into the Retail Application by adding the dashboard report's task flow URL in the application's reports menu model XML file:

1. Add a new custom shared library. See the section, "Adding a Custom Shared Library."
2. Implement the dashboard page components in the custom shared library workspace. Dashboard page components are built as Oracle ADF bounded task flows. See the section, "Creating New ADF Content," for important considerations when building custom ADF content.
3. Note the task flow URL and the parameters of the dashboard task flow.
4. Register the task flow URL and parameters into the application's Reports Menu Model. See the section, "Adding or Modifying an Item in the Reports Menu."
5. Re-generate the Custom Shared Library WAR file from the Custom Shared Library workspace. Shut down the Retail Application and its Custom Shared Library Registry, redeploy the Custom Shared Library WAR file, and restart the Retail Application components.
6. Grant security permissions to dashboard page components. See the section, "Creating New ADF Contents."
7. Test the Retail Application. Log in and verify the newly added content.

Adding a New External Dashboard into the Reports Menu

This section describes the steps for adding external dashboard reports into the Retail Application's UI Sidebar. The reports such as OBIEE, BI Publisher or Microstrategy can be integrated into the Retail Application by adding the web URL of the dashboard report in the application's reports menu model XML file.

1. Create the dashboard report in the reporting server instance.
2. Obtain the URL to the dashboard report along with any parameters needed to render it.
3. Add a New Custom Shared Library. See the section, "Adding a Custom Shared Library."
4. Open the Custom Shared Library workspace in JDeveloper.
5. Add a new item in the reports menu for the URL and parameters of the dashboard report. See the section, "Adding or Modifying an Item in the Reports Menu."
6. Re-generate the Custom Shared Library WAR file from the Custom Shared Library workspace. Shut down the Retail Application and its Custom Shared Library Registry, redeploy the Custom Shared Library WAR file, and restart the Retail Application components.
7. Test the Retail Application. Log in and verify that a link to the BI dashboard appears in the UI's sidebar task tree under a folder My Custom Dashboards.

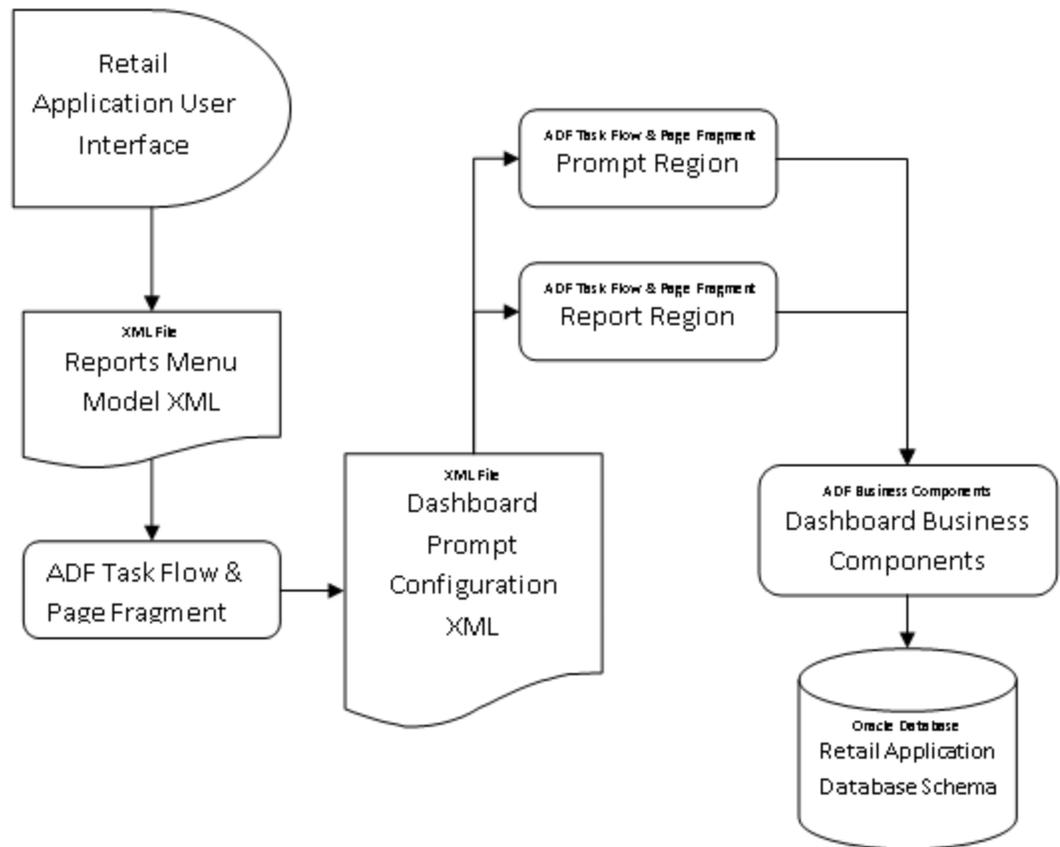
Retail Application Included Dashboard Customization Scenarios

This section contains supported customization use cases for out-of-the-box dashboards included in the Retail Application installation.

Understanding Design Patterns of Included Dashboards

As noted in the section, "Anatomy of a Dashboard," dashboards are composed of prompt regions and report regions.

The representation of those parts at runtime are depicted in the diagram below:

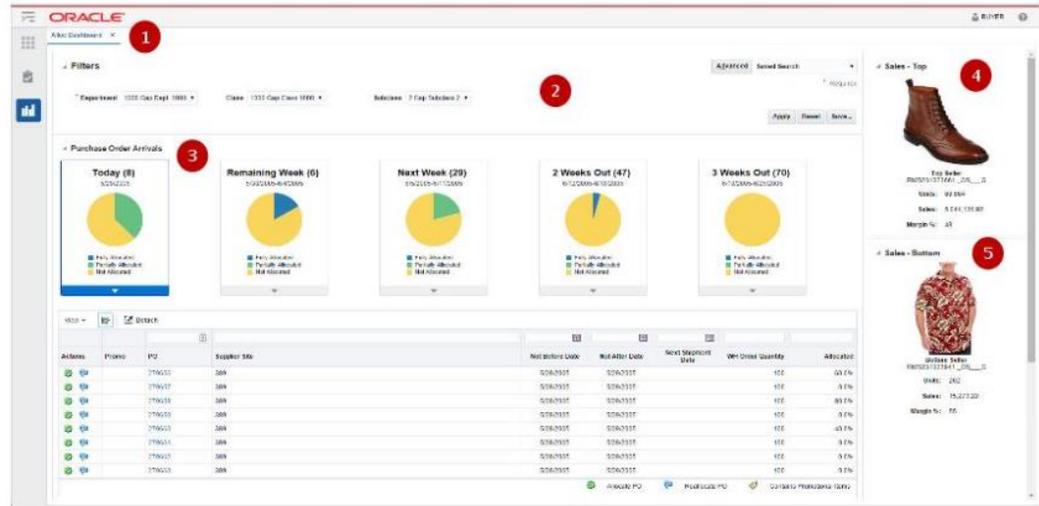


The dashboard page, prompts, and reports are implemented as ADF bounded task Flows and page fragments. The prompts and regions are configured in an XML file called a Dashboard Prompt Configuration XML. The dashboard prompt XML arranges visually the prompts and regions on the dashboard page.

The dashboard task flow is launchable from the Retail Application User Interface's Reports Menu. In order to make the report launchable, the dashboard task flow is registered in the application's reports menu model XML.

A set of ADF business components enable data from the Retail Application schema to be rendered in the report regions.

The figure below shows a rendering of a dashboard (called Alloc Dashboard) in a Retail Application:



The various parts of this example dashboard are described below:

1	Dashboard Page	For this example – this called the Allocations Dashboard or Alloc Dashboard. It contains 1 prompt and 3 reports. It is implemented as an ADF bounded task flow called AllocDashboardFlow.
2	Prompt Region	This is the prompt region that users modify to change the view of the various reports in this dashboard. It is implemented as an ADF bounded task flow called DashboardFilterFlow.
3	Order Status Report	A report region showing a row of information tiles and a detail table. It is implemented as an ADF bounded task flow called OrderStatusFlow.
4	Top Seller Report	A report region showing the top selling item. It is implemented as an ADF bounded task flow called TopSellerFlow
5	Bottom Seller Report	A report region showing the bottom selling item. It is implemented as an ADF bounded task flow called BottomSellerFlow

The dashboard is launched from the application's reports menu. The reports menu model XML file (as shown below) contains an item called allocDashboard registering the task flow, AllocDashboardFlow, into the reports menu.

Note: The dashboard task flow includes a parameter called dashboardConfigXML which references the location of the Dashboard Prompt Configuration XML file

```
<?xml version="1.0" encoding="UTF-8" standalone="yes"?>
<NavigationDefinition id="Folder_1"
  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
  xmlns="urn:www.oracle.com:oracle.retail.apps.framework.navigation"
  xsi:schemaLocation="urn:www.oracle.com:oracle.retail.apps.framework.navigation:
classpath:oracle/retail/apps/framework/uishell/navigation/model/schema/NavigationM
odel.xsd">
  <Items>
    <Item id="allocDashboard" title="Alloc Dashboard"
      shortDesc="Alloc Dashboard"
      type="taskflow">
      <url>/WEB-INF/AllocDashboardFlow.xml#AllocDashboardFlow</url>
      <Parameters>
        <Parameter
id="dashboardConfigXML">/oracle/AllocDashboardPrompt.xml</Parameter>
      <Parameters>
      </Item>
    </Items>
  </NavigationDefinition>
```

The Dashboard Prompt Configuration XML file for this example is called the AllocDashboardPrompt.xml and it contains the following entries:

```
<?xml version="1.0" encoding="UTF-8" ?>
<Dashboard layout="column"
  xmlns="urn:www.oracle.com:oracle.retail.apps.framework.dashboard"
  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
  xsi:schemaLocation="urn:www.oracle.com:oracle.retail.apps.framework.dashboard:
classpath:oracle/retail/apps/framework/dashboard/model/schema/dashboardSchema.xsd"
  >
  <Vectors>
    <Vector>
      <Items>
        <Item id="filter" type="prompt">
          <url>/WEB-
INF/DashboardFilterFlow.xml#DashboardFilterFlow</url>
        </Item>
        <Item id="orderStatus" type="report">
          <url>/WEB-
INF/oracle/retail/apps/framework/dvtcontextawarereport/publicui/flow/DVTContextAwa
reReportFlow.xml#DVTContextAwareReportFlow</url>
          <Parameters>
            <Parameter id="taskflowURL">/WEB-
INF/OrderStatusFlow.xml#OrderStatusFlow</Parameter>
            <Parameter id="parameterName1">departmentIds</Parameter>
            <Parameter id="payloadKeyName1">DepartmentId</Parameter>
            <Parameter id="parameterName2">classIds</Parameter>
            <Parameter id="payloadKeyName2">ClassId</Parameter>
            <Parameter id="parameterName3">subclassIds</Parameter>
            <Parameter id="payloadKeyName3">SubclassId</Parameter>
          </Parameters>
        </Item>
      </Items>
    </Vector>
    <Vector width="300px">
      <Items>
        <Item id="topSeller" type="report">
          <url>/WEB-
INF/oracle/retail/apps/framework/dvtcontextawarereport/publicui/flow/DVTContextAwa
reReportFlow.xml#DVTContextAwareReportFlow</url>
```

```

        <Parameters>
            <Parameter id="taskflowURL"/>/WEB-
INF/oracle/retail/apps/flow/TopSellerFlow.xml#TopSellerFlow</Parameter>
            <Parameter id="parameterName1">departmentIds</Parameter>
            <Parameter id="payloadKeyName1">DepartmentId</Parameter>
            <Parameter id="parameterName2">classIds</Parameter>
            <Parameter id="payloadKeyName2">ClassId</Parameter>
            <Parameter id="parameterName3">subclassIds</Parameter>
            <Parameter id="payloadKeyName3">SubclassId</Parameter>
        </Parameters>
    </Item>
    <Item id="bottomSeller" type="report">
        <url>/WEB-
INF/oracle/retail/apps/framework/dvtcontextawarereport/publicui/flow/DVTContextAwa
reReportFlow.xml#DVTContextAwareReportFlow</url>
        <Parameters>
            <Parameter id="taskflowURL"/>/WEB-
INF/oracle/retail/apps/flow/BottomSellerFlow.xml#BottomSellerFlow</Parameter>
            <Parameter id="parameterName1">departmentIds</Parameter>
            <Parameter id="payloadKeyName1">DepartmentId</Parameter>
            <Parameter id="parameterName2">classIds</Parameter>
            <Parameter id="payloadKeyName2">ClassId</Parameter>
            <Parameter id="parameterName3">subclassIds</Parameter>
            <Parameter id="payloadKeyName3">SubclassId</Parameter>
        </Parameters>
    </Item>
</Items>
</Vector>
</Vectors>
</Dashboard>

```

The file registers the prompts and regions into a hierarchy of vectors and items. The structure of the dashboard prompt configuration XML file is explained in the following section.

The Dashboard Prompt Configuration XML File

The root element of the model is the “Dashboard” element. Other than schema and namespace information, the “Dashboard” element has a “layout” attribute that must be provided, and must contain a value of either “row” or “column”. This attribute arranges the regions of the dashboard in rows or columns.

Inside the “Dashboard” element is a “Vectors” element, which contains a list of “Vector” elements. If the dashboard layout=“row”, then each “Vector” corresponds to a single row in the dashboard. Conversely, if layout=“column”, then each “Vector” corresponds to a column. To add a new row or column, simply add another “Vector” element inside the “Vectors” element.

Up to 12 rows and 6 columns can be displayed in a dashboard, not including separate header and footer regions. Any items beyond 12 rows or 6 columns will not appear on the dashboard.

Rows in the dashboard display from top to bottom. If layout=“row”, the first “Vector” will appear toward the top of the page. Columns appear from left to right, so if layout=“column”, the first “Vector” will appear toward the left of the page.

Each “Vector” element contains an “Items” element, which contains a list of “Item” elements. The xml structure for “Item” elements is very similar to the tasks or reports sidebar menu models.

The following attributes are supported on dashboard items:

Name	Description	Required
id	A unique id for this region	Yes
type	The type of content contained in this region. Valid values: prompt, report	Yes
width	The width of this region. See below for more details.	No
height	The height of a region. See below for more details.	No
url	The task flow url to open in this region	Yes
Parameters	Parameters to be sent to the URL	No

In addition to the items arranged in rows or columns, you can add a header and footer region to the dashboard. Both horizontally span the full width of the dashboard, with the header appearing at the top and the footer appearing at the bottom. These can be added in the model by specifying a “HeaderItem” or “FooterItem” child of the root “Dashboard” element. These regions support all the same properties as other items in the dashboard.

Vectors and items expose width and height attributes to enable applications to resize elements to best fit on the dashboard. Certain rules and best practices apply when using these attributes.

First, if the dashboard is in row layout, then all items in a row must have the same height. The height attribute from the vector is used to specify the height of every item in that row. Because rows are assumed to span the full width, the vector width is ignored in this layout.

In the same way, when the dashboard is arranged in column layout, all items in a column must have the same width. The vector width is used to specify the width of every item in that column. Columns should expand vertically to accommodate all content, allowing a scrollbar if necessary. Therefore, vector height is ignored in this layout.

In addition to the width and height attributes, a vector also provides a dimensionsFrom attribute. The valid values of the dimensionsFrom attribute can be auto, child or parent. The dimensionsFrom attribute is used to specify whether the Items in the vector inherit their dimensions from the parent decorative box or not. When the dimensionsFrom attribute is parent, the Items in the vector will be stretched to fill any available space in the vector. When the dimensionsFrom attribute is child, the Items in the vector won't stretch but will display scroll bars if they cannot be accommodated in the available space in the decorative box. The default value of the dimensionsFrom attribute is 'auto' which gives preference to the child dimensions.

The following table indicates which attributes are used, based on which dashboard layout has been selected.

	Row layout	Column layout
Vector width	ignored	valid
Vector height	valid	ignored
Item height	ignored	valid
Item width	valid	ignored

In column layout, either every vector width should be provided, or every width should be omitted. If all widths are omitted, columns default to divide the total width evenly. When the widths are omitted, the `dimensionsFrom` attribute of the vector can be set to parent if stretching of the Items is desired. If widths are provided, they should be specified as a percentage, and the sum of all widths should equal 100%.

In row layout, heights may be specified or omitted as desired. If a row's height is omitted, it will default to get the dimension from the content inside the regions. When the row's height is omitted, the `dimensionsFrom` attribute of the vector can be set to parent if stretching of the Items is desired. Heights should be specified using an exact unit such as px or em. Percentages should not be used.

If these rules and best practices are ignored, results may be inconsistent or undesirable. Items may not size as intended, and horizontal scrollbars or nested vertical scrollbars may appear.

In addition to resizing entire rows or columns, individual items inside the row or column can also be resized. As mentioned above, items in a row should all have the same height, so in row layout, item height is ignored and only item width is valid. And items in a column should have the same width, so item width is ignored and only item height is valid. See the table above for details.

The same rules and best practices that apply to vector height and width also apply to setting item height and width.

Refreshing Reports on Prompt Changes

When the user changes a value on the dashboard prompt, that value translates as a parameter change on the different reports on the dashboard.

This is accomplished through a feature in ADF called Contextual Events. In simple terms, contextual events are signals that a page in a task flow can generate for which pages in other task flows can listen and react to. For the included dashboards, prompt regions generate a contextual event whenever the user changes a prompt field value. When a report "hears" an event, it will process the contextual event information (called a payload) to extract the changed prompt value, use that value as a parameter to the report query and refresh the report region.

In the example depicted in the previous section, the dashboard prompt configuration XML file shows all three reports registered under a wrapper task flow called the `DVTContextAwareReportFlow`.

```
<Item id="bottomSeller" type="report">
  <url>/WEB-INF/oracle/retail/apps/framework/dvtcontextawarereport/publicui/flow/DVTContextAwareReportFlow.xml#DVTContextAwareReportFlow</url>
  <Parameters>
    <Parameter id="taskflowURL">/WEB-INF/oracle/retail/apps/flow/BottomSellerFlow.xml#BottomSellerFlow</Parameter>
    <Parameter id="parameterName1">departmentIds</Parameter>
```

```

        <Parameter id="payloadKeyName1">DepartmentId</Parameter>
        <Parameter id="parameterName2">classIds</Parameter>
        <Parameter id="payloadKeyName2">ClassId</Parameter>
        <Parameter id="parameterName3">subclassIds</Parameter>
        <Parameter id="payloadKeyName3">SubclassId</Parameter>
    </Parameters>
</Item>

```

The DVContextAwareReportFlow wrapper task flow enables a report to listen to contextual events generated by prompt regions.

Note the pairs of parameterNameN and payloadKeyNameN registered in the <Parameters> section for the report task flow BottomSellerFlow. A pair of parameter name and payload key name tells the framework what prompt value is mapped to a specific report parameter. From the example above, parameterName1 is specified as “departmentIds” and payloadKeyName1 as “departmentIds”.

This means the following:

- There is a task flow parameter for the BottomSellerFlow called “departmentIds”.
- When the user changes the value of the department ID in the prompt region, the system will generate a contextual event with the new department value stored as reference “DepartmentId”.
- The report is context aware (as indicated by the wrapper task flow – DVContextAwareReportFlow). When the report detects the contextual event, the framework will get the new department ID value from the payload by using the key “DepartmentId”, use it to update the BottomSellerFlow task flow’s “departmentId” parameter, and refresh the report region.

For more details on the DVContextAwareReportFlow, see the section, “Adding a DVT Taskflow-based Contextual Report.”

List of Retail Applications Included in Dashboards

This section contains a listing of all dashboard reports included in the Retail Application.

The screenshot displays the 'Inventory Control Dashboard' with several key components:

- Filters:** Includes dropdowns for Chain, Area (33), Location, Supplier Site, Department, Class, and Subclass. A note states: "** At least one is required".
- Summary Cards:**
 - Transfers Pending Approval: 0
 - Overdue Transfers: 389
 - Overdue Allocations: 64
 - Overdue RTVs: 3
 - Stock Orders Pending Close: 124
 - Stock Counts Missing: 44
 - Stock Count Unit Variance: 8
 - Unexpected Inventory: 416
- Stock Count Unit Variance Report:**
 - Report Refreshed: 10/25/10 1:54 AM
 - Table columns: Location Name, Count Type, Count Date, Total Variance %, Variance % Graph.
 - Data rows include locations like KK_VWH, P_VWH-10, VWH1-20, MRT_MAT_VWH, SS_PWH 17 - VW..., VWH-10, and VWH2-20.
 - The Variance % Graph shows horizontal bars for each location, with red bars indicating negative variance and blue bars indicating positive variance.

The salient features of Inventory Control Dashboard are-

- Resolves inventory discrepancies and manages stock counts at the corporate level
- Key responsibilities
 - Ensuring transfers and RTVs closed
 - Ensuring stock counts are reconciled and processed
- Dashboard focused on
 - Timely closing of transfers, allocations, and RTVs
 - Timely processing of stock counts

Benefits

- Reduces time searching for stock count exceptions from 5 clicks to zero
- Eliminates the need to build custom reporting to monitor for negative inventory
- Allows for approval of multiple transfers in one action
- Provides suggested actions for resolving
 - Inventory in inactive/discontinued locations
 - Items unintentionally ranged to a location

Job roles with access to the Inventory control dashboard are as follows

- CORPORATE_INVENTORY_CONTROL_ANALYST_JOB
- INVENTORY_CONTROL_MANAGER_JOB
- RMS_APPLICATION_ADMINISTRATOR_JOB

Different reports under the Inventory control dashboard are as below:-

- Transfer Pending Approval
- Overdue Transfers
- Overdue Allocations
- Overdue RTVs
- Stock Orders Pending Close
- Stock Counts Missing
- Stock Count Unit Variance
- Unexpected Inventory

The Task flow url of the Inventory control dashboard is below

/WEB-

INF/oracle/retail/apps/rms/dashboards/view/inventorycontrol/flow/InventoryControlDashboardFlow.xml#InventoryControlDashboardFlow

The only required parameter for the Inventory control task flow is “dashboardXml” which is a metadata or model file for this dashboard reports and prompt/filter

The path for the “dashboardXml” parameter for Inventory control dashboard is as below
/oracle/retail/apps/rms/dashboards/model/inventorycontrol/InventoryControlDashboardModel.xml

The Inventory control prompt and description are as follows

Prompt Name	Description
Chain/chainIds	Chain
Area/areaIds	Area
Location/locationIds	Location
Supplier Site/supplierSiteIds	SupplierSite
Department/departmentIds	Department
Class/classIds	ClassId
Subclass/subClassIds	SubClass

The payload map between each prompt of IC is as follows

Prompt Name	Pay Load Key Name
Chain/chainIds	Chain
Area/areaIds	Area
Location/locationIds	Location
Supplier Site/supplierSiteIds	SupplierSite
Department/departmentIds	Department
Class/classIds	ClassId
Subclass/subClassIds	SubClass

The screenshot displays the Oracle Merchandising Data Steward Dashboard. At the top, it shows the Oracle logo and 'Merchandising' text. The user is logged in as 'RMS_ADMIN'. The dashboard includes a 'Data Steward Dashboard' tab and a 'Filters' section with several dropdown menus: Department (set to 11), Class, Subclass, Supplier Site, Country of Sourcing, and Created Date. There are also 'Search', 'Reset', and 'Save...' buttons. Below the filters is a table titled 'Incomplete Items' with a 'Report Refreshed' timestamp of 10/25/16 2:34 AM. The table has columns for 'Required Completion', 'Item', 'Description', 'Reference Items', 'VAT', 'Simple Packs', 'UDAs', and 'Locations'. The data rows are as follows:

Required Completion	Item	Description	Reference Items	VAT	Simple Packs	UDAs	Locations
<input type="radio"/>	106000025	sample	<input type="radio"/>	<input checked="" type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
<input type="radio"/>	107100010	Test	<input type="radio"/>	<input checked="" type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
<input type="radio"/>	127000008	Test Item	<input type="radio"/>	<input checked="" type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
<input type="radio"/>	127000067	Test Tin Bug	<input type="radio"/>	<input checked="" type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
<input type="radio"/>	127400044	Test Tin FP Bug 1274000...	<input type="radio"/>	<input checked="" type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>

Data Steward Dashboard

The salient features of Data Steward Dashboard are:-

- Maintains standards and accuracy of foundation and item data
- Key responsibilities
 - Maintaining cross-enterprise attribution
 - Ensuring completeness of item data
- Dashboard focused on
 - Timeliness and completeness of item setup

Benefits

- Monitors for specific attributes required by department
 - Highlights overall completeness of the item
 - Tracks required attributes even after item approval
 - Single click to take action from the report

Job roles with access to the Data steward dashboard are as follows

- RMS_DATA_STEWARD_JOB
- RMS_APPLICATION_ADMINISTRATOR_JOB

Different reports under the Data steward dashboard are as follows

- Incomplete Items

Task flow URL of Inventory control dashboard is as below

/WEB-

INF/oracle/retail/apps/rms/dashboards/view/datasteward/flow/DataStewardDashboardFlow.xml#DataStewardDashboardFlow

The only required parameter for the Inventory control task flow is “dashboardXml” which is a metadata or model file for this dashboard reports and prompt/filter

The path for the “dashboardXml” parameter for Inventory Data steward control dashboard is as below

/oracle/retail/apps/rms/dashboards/model/dataSteward/DataStewardDashboardModel.xml

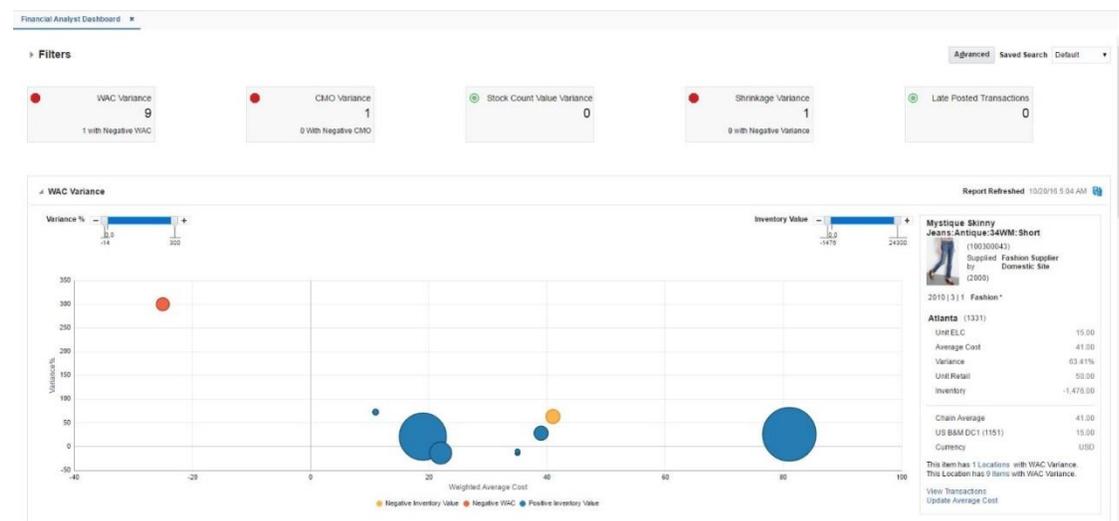
The Data steward prompt and description are as follows:

Prompt Name	Description
Department/departmentIds	Department
Class/classIds	ClassId
Subclass/subclassIds	Subclass
Store/storeIds	Store
Supplier Site/supplierSiteIds	SupplierSite
Brands/brands	BrandName
Country of Origin/originCountries	CountryOfOrigin
UDA Values/udaValues	UdaValue
Location/locationIds	Location
Create Date/createDates	CreateDate
Item Type/itemTypeId	ItemType

Prompt Name	Description
Item Level/itemLevelId	ItemLevel
Transaction Level/tranLevelId	TransactionLevel
UDA/udaId	UDA

The payload map between each prompt of DS is as follows

Prompt Name	Pay Load Key Name
Department/departmentIds	Department
Class/classIds	ClassId
Subclass/subclassIds	Subclass
Store/storeIds	Store
Supplier Site/supplierSiteIds	SupplierSite
Brands/brands	BrandName
Country of Origin/originCountries	CountryOfOrigin
UDA Values/udaValues	UdaValue
Location/locationIds	Location
Create Date/createDates	CreateDate
Item Type/itemTypeId	ItemType
Item Level/itemLevelId	ItemLevel
Transaction Level/tranLevelId	TransactionLevel
UDA/udaId	UDA



Financial Analyst Dashboard

The salient features of Finance Analyst Dashboard are:-

- Focuses on inventory valuation and stock ledger accuracy

Key responsibilities

- Reviewing and approving the value of a physical inventory
- Resolving inventory valuation discrepancies prior to month end

Dashboard focused on

- WAC and/or CMO variances outside of an expected tolerance
- Monitoring late posted transactions from previous month
- Addressing variances in inventory valuation based on counts

Benefits

- Highlights financial anomalies in inventory valuation prior to month close
- Proactively manage WAC variances to ensure inventory valuation accuracy
- Allows for stock count value variance to be accepted in a single action
- Proactively monitor and update budgeted shrink and CMO based on business trends

Benefits

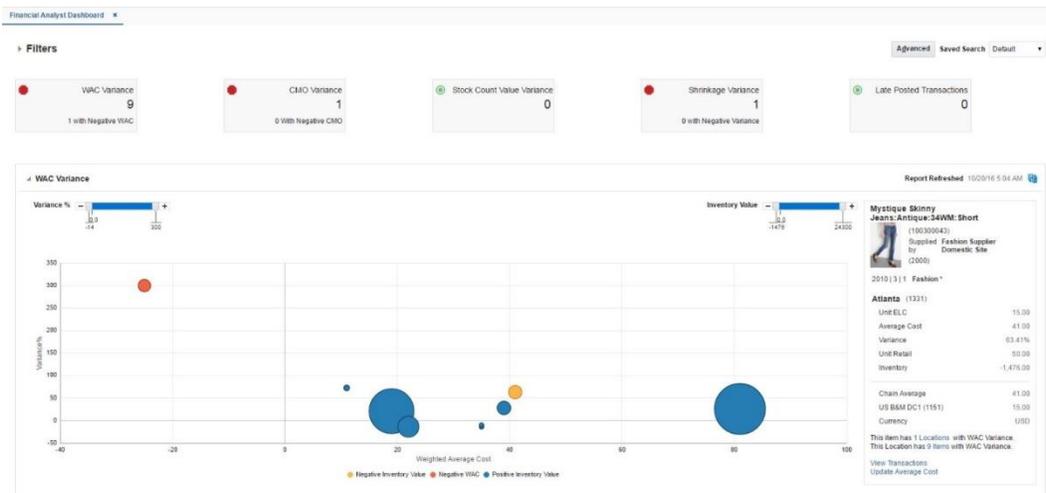
- Highlights financial anomalies in inventory valuation prior to month close
- Proactively manage WAC variances to ensure inventory valuation accuracy
- Allows for stock count value variance to be accepted in a single action
- Proactively monitor and update budgeted shrink and CMO based on business trends

Job roles with access to the Financial analyst dashboard are as follows

- FINANCE_ANALYST_JOB
- FINANCE_MANAGER_JOB
- RMS_APPLICATION_ADMINISTRATOR_JOB

Different reports available in this dashboard are:

1. WAC Variance Report



/WEB-

INF/oracle/retail/apps/rms/dashboards/view/financialanalyst/flow/FinancialAnalystDashboardPromptFlow.xml#FinancialAnalystDashboardPromptFlow

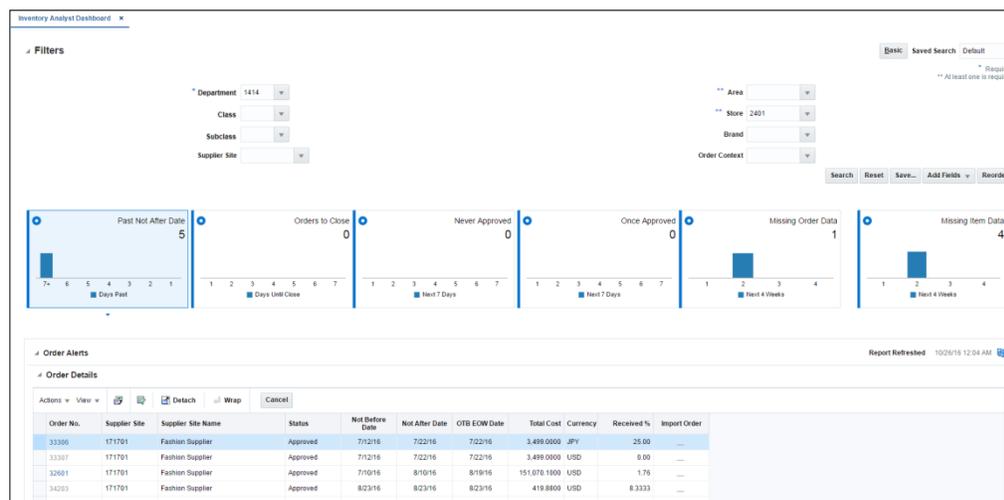
The Finance Analyst prompt and description are as follows

Prompt Name	Description
setOfBooksIds	SetOfBooksId
orgUnitIds	OrgUnitId
locationIds	Location
departmentIds	Department
classIds	ClassId
subclassIds	Subclass

The below table describes the prompt names along with the Payload Key names

Prompt Name	Payload Key Name
setOfBooksIds	SetOfBooksId
orgUnitIds	OrgUnitId
locationIds	Location
departmentIds	Department
classIds	ClassId
subclassIds	Subclass
Area	Area
Brand	BrandName
Order Context	OrderType

Below is the screenshot of the dashboard.



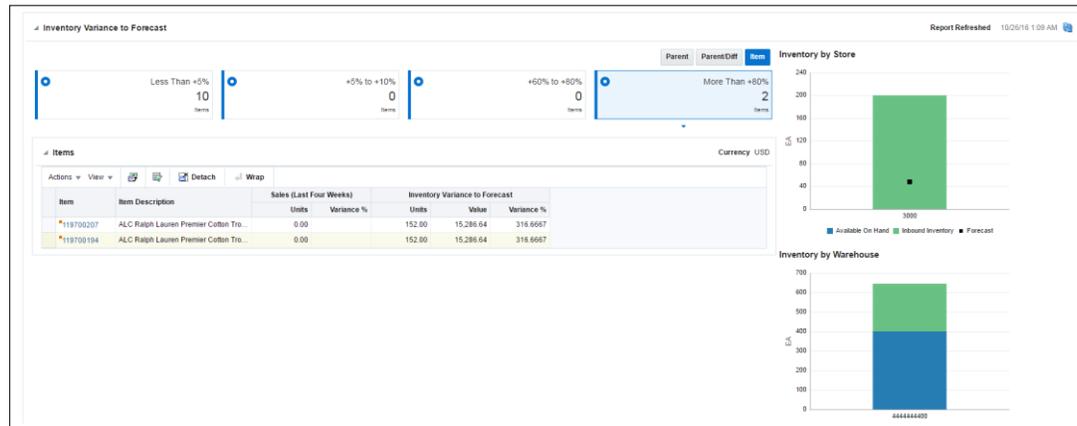
Dashboard Filter and Order Alert Report

Negative Inventory Report Refreshed 10/26/16 12:04 AM

Actions View Detach Wrap

Item	Description	Standard UOM	Location	Location Name	Stock On Hand	In Progress Shipments		
						Orders	Transfers	Allocations
*03350165	SS - V-Neck Tee-WHITE Petite Large	EA	2401	SS Central Franchise Store222222...	0.00	0.00	0.00	0.00
*03350281	SS - V-Neck Tee-Red Petite	EA	2401	SS Central Franchise Store222222...	0.00	0.00	0.00	0.00
*03350425	SS - V-Neck Tee Yellow Petite Large	EA	2401	SS Central Franchise Store222222...	0.00	0.00	0.00	0.00
*03350100	SS - V-Neck Tee-BLACK Petite	EA	2401	SS Central Franchise Store222222...	0.00	0.00	0.00	0.00
*03350222	SS - V-Neck Tee-BROWN Petite	EA	2401	SS Central Franchise Store222222...	0.00	0.00	0.00	0.00
*03350370	SS - V-Neck Tee Yellow Regular	EA	2401	SS Central Franchise Store222222...	0.00	0.00	0.00	0.00
*03350134	SS - V-Neck Tee-WHITE Regular	EA	2401	SS Central Franchise Store222222...	0.00	0.00	0.00	0.00
*03350290	SS - V-Neck Tee-Red Petite Medium	EA	2401	SS Central Franchise Store222222...	0.00	0.00	0.00	0.00
*03350337	SS - V-Neck Tee Blue Regular Large	EA	2401	SS Central Franchise Store222222...	0.00	0.00	0.00	0.00
*03350265	SS - V-Neck Tee Red Regular Me...	EA	2401	SS Central Franchise Store222222...	0.00	0.00	0.00	0.00
*03350302	SS - V-Neck Tee Red Petite Large	EA	2401	SS Central Franchise Store222222...	0.00	0.00	0.00	0.00
*03350311	SS - V-Neck Tee Blue Regular	EA	2401	SS Central Franchise Store222222...	0.00	0.00	0.00	0.00
*03350361	SS - V-Neck Tee Blue Petite Large	EA	2401	SS Central Franchise Store222222...	0.00	0.00	0.00	0.00
*03350417	SS - V-Neck Tee Yellow Petite Med.	EA	2401	SS Central Franchise Store222222...	0.00	0.00	0.00	0.00
*03350071	SS - V-Neck Tee-BLACK Regular	EA	2401	SS Central Franchise Store222222...	0.00	0.00	0.00	0.00
*03350118	SS - V-Neck Tee-BLACK Petite Me...	EA	2401	SS Central Franchise Store222222...	0.00	0.00	0.00	0.00
*03350126	SS - V-Neck Tee-BLACK Petite Large	EA	2401	SS Central Franchise Store222222...	0.00	0.00	0.00	0.00
*03350164	SS - V-Neck Tee-WHITE Regular L...	EA	2401	SS Central Franchise Store222222...	0.00	0.00	0.00	0.00

Negative Inventory Report



Inventory Variance to Forecast Report



The salient features of Inventory Analyst Dashboard are:-

- Manages inventory levels across locations
- Key responsibilities:
 - Monitoring stock variance to plan
 - Creating adjustments to open orders or replenishment settings
- Dashboard highlights
 - Monitoring forecasted sales to forecasted inventory
 - Resolving purchase order issues which may prevent timely shipping or receiving

Benefits

- Shortens order approval process by 60% or more
- Eliminates need to build custom reports to monitor ordering exceptions
- Shortens steps for closing overdue orders from 10 to 1
- Proactively looks for forecasted inventory shortfall
- Provides single-step resolutions
 - Canceling orders
 - Creating orders or transfers
 - Re-allocation

Job roles with access to the Inventory analyst dashboard are as follows

- INVENTORY_ANALYST_JOB
- INVENTORY_MANAGER_JOB
- RMS_APPLICATION_ADMINISTRATOR_JOB

The following are the reports available in the dashboard:

- Order Alerts
- Negative Inventory
- Inventory Variance to Forecast

Dashboard Task flow URL:

/WEB-

INF/oracle/retail/apps/rms/dashboards/view/inventoryanalyst/flow/InventoryAnalystDashboardFlow.xml#InventoryAnalystDashboardFlow

Dashboard Parameters:

- dashboardXml - Location of metadata/model file that contains dashboard prompt and reports in the dashboard. Value -
 /oracle/retail/apps/rms/dashboards/model/inventoryanalyst/InventoryAnalystDashboardModel.xml

Dashboard Prompt task flow URL:

/WEB-

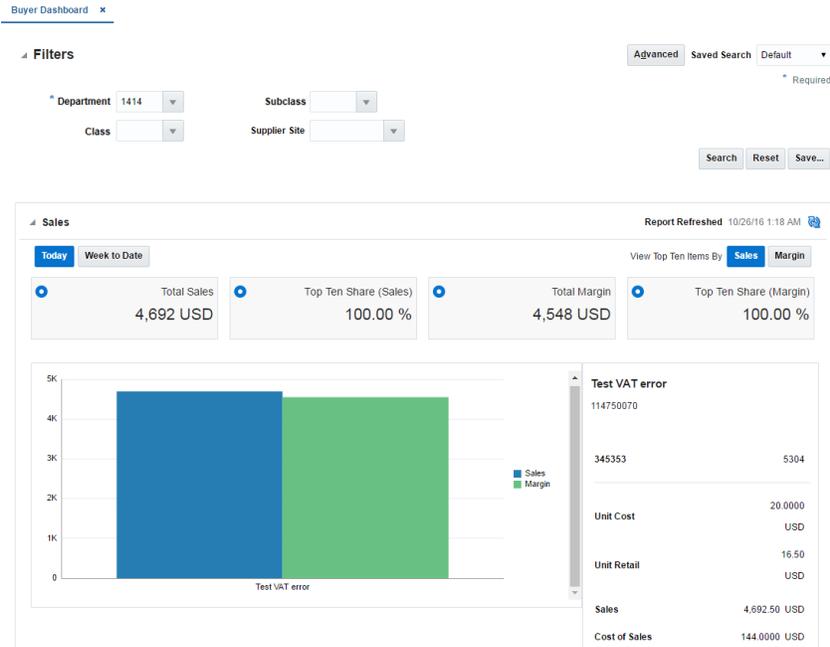
INF/oracle/retail/apps/rms/dashboards/view/inventoryanalyst/flow/InventoryAnalystDashboardPromptFlow.xml#InventoryAnalystDashboardPromptFlow

The table below describes the prompt names along with the Payload Key names:

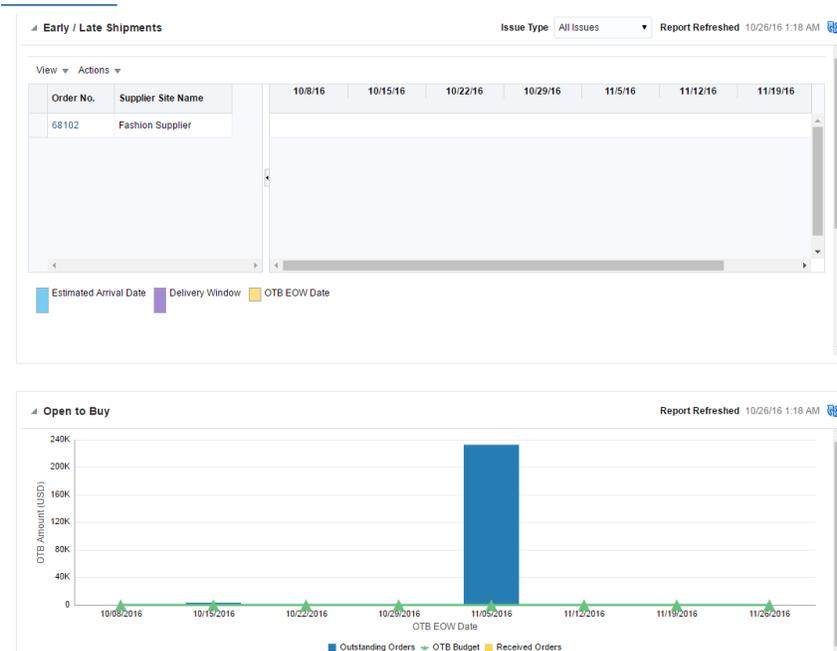
SI. No	Prompt Name	Payload Key Name	Description
1.	Department	Department	Department Id
2.	Class	ClassId	Class Id
3.	Subclass	Subclass	Subclass Id
4.	Supplier Site	SupplierSite	Supplier Site Id
5.	Store	Store	Store Id
6.	Store Grade Group	StoreGradeGroup	Store Grade Group Id
7.	Area	Area	Area Id
8.	Brand	BrandName	Item Brand Name

Sl. No	Prompt Name	Payload Key Name	Description
9.	Order Context	OrderType	Order Context Id

Buyer Dashboard



Dashboard Filter and Daily Sales Report:



Early/Late Shipment and Open to Buy Reports:

Order No.	Supplier Site Name	Not Before Date	Not After Date	Total Cost	Total Retail	Currency	Created By
No data to display.							

Order Pending Approval Report

The salient features of the Buyer Dashboard are:-

- Focuses on managing sales and margin performance to plan
- Key responsibilities:
 - Monitoring open to buy
 - Approving purchase orders approval
- Dashboard highlights
 - Monitoring shipments for potential impacts on OTB
 - Highlighting orders awaiting their approval

Benefits

- Provides snapshot view of today's top 10 selling items
- Proactively monitors order shipment accuracy
- Provides single step resolutions for
 - Updating PO Dates
 - Canceling Orders
- Allows for approval of multiple orders in one action
- Provides graphical view of buying budget and spend by period

Job roles with access to the Buyer dashboard are as follows

- BUYER_JOB
- RMS_APPLICATION_ADMINISTRATOR_JOB

The following reports are available in the dashboard

- Daily Sales
- Early / Late Shipments
- Open to Buy
- Orders Pending Approval

Dashboard Task flow URL:

/WEB-INF/oracle/retail/apps/rms/dashboards/view/buyer/flow/BuyerDashboardFlow.xml#BuyerDashboardFlow

Dashboard Parameters:

- dashboardXml – Location of metadata/model file that contains dashboard prompt and reports in the dashboard. Value -
/oracle/retail/apps/rms/dashboards/model/buyer/BuyerDashboardModel.xml

Dashboard Prompt task flow URL:

/WEB-INF/oracle/retail/apps/rms/dashboards/view/buyer/flow/BuyerDashboardFilterFlow.xml
#BuyerDashboardFilterFlow

SI. No	Prompt Name	Payload Key Name	Description
1.	Department	Department	Department Id
2.	Class	ClassId	Class Id
3.	Subclass	Subclass	Subclass Id
4.	Store	Store	Store Id
5.	Supplier Site	SupplierSite	Supplier Site Id
6.	Brand	BrandName	Item Brand Name
7.	Order Context	OrderType	Order Context Id
8.	Country of Sourcing	CountryOfOrigin	Country of Sourcing Location

Adding or Replacing a Report in an Included Dashboard

This section discusses the steps needed to add or replace a report on an included dashboard.

1. Add a custom shared library. Refer to the section, Add a Custom Shared Library. An existing custom shared library can be reused.
2. Identify the dashboard to be modified. Refer to the section, List of Retail Application Included Dashboards.
3. If needed, open the custom shared library workspace in JDeveloper and build your own custom report as an ADF bounded task flow. Refer to the section, Creating New ADF Content.

If the report has to refresh when a prompt on the dashboard is changed by the user, make sure that task flow parameters are present for the prompt values your report will need to react to. Refer to the section, List of Retail Application Included Dashboards, for a list of prompt payload values that can be generated for the dashboard.

4. Note the new report's task flow URL and parameters.
5. Obtain a copy of the dashboard prompt configuration XML file for the dashboard to be modified. Refer to the section, List of Retail Application Included Dashboards.
6. Modify the copy of the dashboard prompt configuration XML file to add an item or replace an existing item for the new report. Refer to the section, The Dashboard Prompt Configuration XML File for details on properly creating a new item entry in the configuration file.

If the report has to refresh when a prompt on the dashboard is changed by the user, make sure that report task flow is wrapped in a DVContextAwareTaskFlow for its item entry in the dashboard prompt configuration XML file. Use the example in the section, Understanding Design Pattern for Included Dashboards, as reference. Refer to the section, Adding a DVT Taskflow-based Contextual Report for usage of the DVContextAwareTaskFlow framework.

7. Modify the included dashboard's entry in the application's Reports Menu to reference the location of your copy of the dashboard prompt configuration XML file. Refer to the section, Adding or Modifying an Item in the Reports Menu.
8. Re-generate the Custom Shared Library WAR file from the Custom Shared Library workspace. Shutdown the Retail Application and its Custom Shared Library Registry, redeploy the Custom Shared Library WAR file, and restart the Retail Application components.
9. Test the Retail Application.

Removing a Report from an Included Dashboard

This section discusses the steps needed to remove a report from an included dashboard.

1. Add a custom shared library. Refer to the section, Add a Custom Shared Library. An existing custom shared library can be reused.
2. Identify the dashboard to be modified. Refer to the section, List of Retail Application Included Dashboards.
3. Obtain a copy of the dashboard prompt configuration XML file for the dashboard to be modified. Refer to the section, List of Retail Application Included Dashboards.
4. Modify the copy of the dashboard prompt configuration XML file and remove the report. You may need to readjust the position of the other reports. Refer to the section, The Dashboard Prompt Configuration XML File for details on the configuration XML attributes that control the rendering of the dashboard.
5. Modify the included dashboard's entry in the application's Reports Menu to reference the location of your copy of the dashboard prompt configuration XML file. Refer to the section, Adding or Modifying an Item in the Reports Menu.
6. Re-generate the Custom Shared Library WAR file from the Custom Shared Library workspace. Shutdown the Retail Application and its Custom Shared Library Registry, redeploy the Custom Shared Library WAR file, and restart the Retail Application components.
7. Test the Retail Application.

Change the Layout of an Included Dashboard

This section discusses the steps needed to change the layout of the reports in included dashboards.

This section discusses the steps needed to change the layout of the reports in included dashboards.

1. Add a custom shared library. Refer to the section, Add a Custom Shared Library. An existing custom shared library can be reused.
2. Identify the dashboard to be modified. Refer to the section, List of Retail Application Included Dashboards.
3. Obtain a copy of the dashboard prompt configuration XML file for the dashboard to be modified. Refer to the section, List of Retail Application Included Dashboards.
4. Modify the copy of the dashboard prompt configuration XML file and rearrange the position of the item entries. Refer to the section, The Dashboard Prompt Configuration XML File for details on the attributes to control the layout of dashboard.
5. Modify the included dashboard's entry in the application's Reports Menu to reference the location of your copy of the dashboard prompt configuration XML file. Refer to the section, Adding or Modifying an Item in the Reports Menu.

6. Re-generate the Custom Shared Library WAR file from the Custom Shared Library workspace. Shutdown the Retail Application and its Custom Shared Library Registry, redeploy the Custom Shared Library WAR file, and restart the Retail Application components.
7. Test the Retail Application.

Adding Contextual Reports

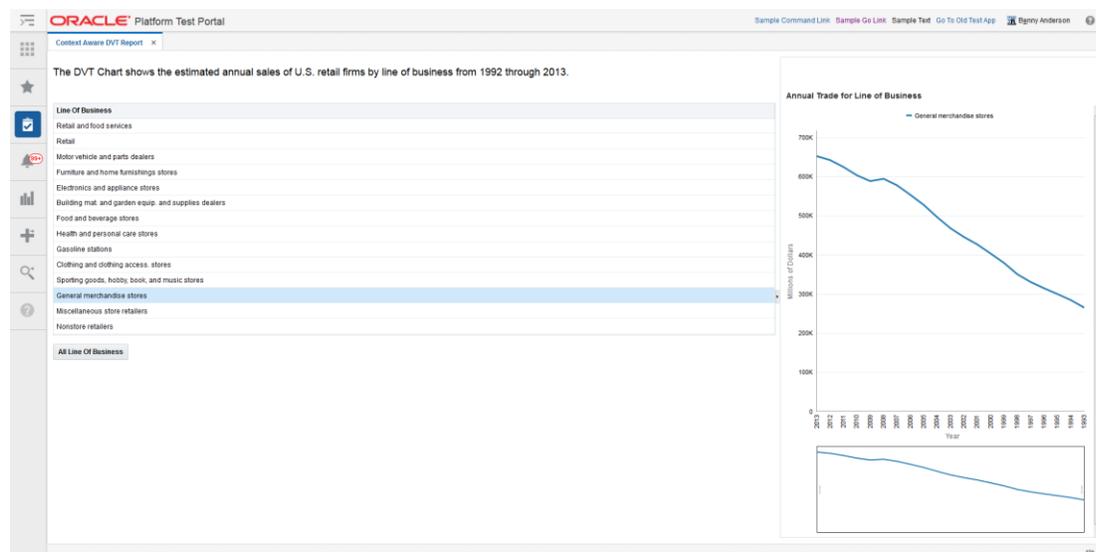
Contextual Reports are reports that appear in a task flow's contextual area section. As discussed in the section, Understanding the Retail Application User Interface, the Contextual Area is a collapsible section to the right of the local area that provides a space to present information that can assist users in completing their tasks.

Since information presented in a contextual area is presented depending on the task or workflow the user is on, contextual areas are associated to task flows and there can be at most one contextual area per task flow.

Within a contextual area – multiple contextual reports can be configured.

Each contextual report can change its contents depending on the action being performed in the user's current workflow.

For example, in the screenshot below, as the user selects the line of business from the table, the line chart will show the annual trade for the selected line of business.



Each task flow publishes contextual business events on key activities happening in the screen. Contextual reports can listen to those events and change its content depending on the payload information associated with the event.

Contextual reports can either be ADF based DVT reports or can be maintained in a separate BI reporting tool. The ADF based DVT reports are ADF taskflows, which are built using the ADF Data Visualization components. The reports maintained in an external reporting tool are URLs accessible in a web browser via a URL. Oracle Retail recommends using Oracle Business Intelligence Enterprise Edition (Oracle BI EE) for an external reporting tool.

The contextual reports are added into a Retail Application by adding and configuring the task flow's contextual area model XML file into the Custom Shared Library.

Retail Applications may provide retailers with a list of possible contextual business events each flow generates. Retailers can configure their contextual reports to react to these events.

Each event will also include information about the event's payload information (example: the item ID of the item being selected in an Allocation Maintenance screen).

List of Contextual Business Events and Payloads

Event Name	Task Flow	Page	ContextualArea Name	Contextual Area Model XML Location	Generated when...	Payload Values
ItemSearchItemSelectedEvent	SearchItemFlow	SearchItems	SearchItem	Rms.ear → RmsViewController.war → AdfRmsFoundationItem.jar → /oracle/retail/apps/framework/uishell/config/contextualarea/ItemSearch_ContextualAreaModel.xml	The user selects an record on the 'Item Results' table search Items screen.	Item – the item selected by the user.
ItemSelectedEvent	MaintainItemFlow	MaintainItem	MaintainItem	Rms.ear → RmsViewController.war → AdfRmsFoundationItem.jar → /oracle/retail/apps/framework/uishell/config/contextualarea/Item_ContextualAreaModel.xml	While opening 'Item Master' screen from 'Item Search' screen.	Item – the item selected by the user.

Event Name	Task Flow	Page	ContextualArea Name	Contextual Area Model XML Location	Generated when...	Payload Values
PackDetailsSelectionEvent	MaintainPacksByCompFlow	MaintainPacksByComp	PackDetails	Rms.ear → RmsViewController.war → AdfRmsFoundationItem.jar → /oracle/retail/apps/framework/uishell/config/contextualarea/PackDetails_ContextualAreaModel.xml	When a row in the 'Packs by component' table is selected by user.	<ul style="list-style-type: none"> Item – the item selected by the user.
ItemSupplierCountrySelectedEvent	MaintainItemSupplierCountryFlow	MaintainItemSupplierCountry	MaintainItemSupplierCountry	Rms.ear → RmsViewController.war → AdfRmsFoundationItem.jar → /oracle/retail/apps/framework/uishell/config/contextualarea/ItemSupplierCountry_ContextualAreaModel.xml	When a row in the Item Supplier country table is selected by user.	<ul style="list-style-type: none"> Item – the item selected by the user. Supplier – the Supplier id selected by the user. Country – the country id selected by the user.
ItemLocationSelectedEvent	MaintainItemLocationsFlow	MaintainItemLocations	ItemLoc	Rms.ear → RmsViewController.war → AdfRmsFoundationItem.jar → /oracle/retail/apps/framework/uishell/config/contextualarea/ItemLoc_ContextualAreaModel.xml	When a row in the Locations table is selected.	<ul style="list-style-type: none"> Item – the item selected by the user. Location – the location id selected by the user.

Event Name	Task Flow	Page	ContextualArea Name	Contextual Area Model XML Location	Generated when...	Payload Values
ItemChildSelectedEvent	MaintainItemChildrenFlow	MaintainItemChildren	MaintainItemChildren	Rms.ear → RmsViewController.war → AdfRmsFoundationItem.jar → /oracle/retail/apps/framework/uishell/config/contextualarea/ItemChildren_ContextualAreaModel.xml	When a row in the Item children table is selected.	<ul style="list-style-type: none"> Item – the item selected by the user.
ItemChildDiffSelectedEvent	MaintainItemChildrenDiffFlow	MaintainItemChildrenDiff	MaintainItemChildrenDiff	Rms.ear → RmsViewController.war → AdfRmsFoundationItem.jar → /oracle/retail/apps/framework/uishell/config/contextualarea/ItemChildrenByDiff_ContextualAreaModel.xml	When a row in the Item children Diff table is selected.	<ul style="list-style-type: none"> Item –the item selected by the user.

Event Name	Task Flow	Page	ContextualArea Name	Contextual Area Model XML Location	Generated when...	Payload Values
ItemSupplierSelectedEvent	MaintainItemSuppsFlow	MaintainItemSupps	MaintainItemSupplier	Rms.ear → RmsViewController.war → AdfRmsFoundationItem.jar → /oracle/retail/apps/framework/uishell/config/contextualarea/ItemSupplier_ContextualAreaModel.xml	When a row in the Item Supplier table is selected.	<ul style="list-style-type: none"> Item – the item selected by the user. Supplier – the supplier id selected by the user.
SubstituteItemSelectedEvent	MaintainSubstituteItemsFlow	MaintainSubstituteItems	SubstituteItems	Rms.ear → RmsViewController.war → AdfRmsFoundationItem.jar → /oracle/retail/apps/framework/uishell/config/contextualarea/SubstituteItems_ContextualAreaModel.xml	When a row in the Items table is selected.	<ul style="list-style-type: none"> Item – the item selected by the user. SubstituteItem -- the substitute item selected by the user.
AlcOrderFinalizeSelectionEvent	SearchAlcFinalFlow	SearchAlcFinal	AlcOrderFinalize	Rms.ear → RmsViewController.war → AdfRmsImport.jar → /oracle/retail/apps/framework/uishell/config/contextualarea/AlcOrderFinalize_ContextualAreaModel.xml	When a row in the 'Alc Order Finalization' table is selected.	<ul style="list-style-type: none"> Order – the 'order no' selected by the user. Shipment – the 'shipment id' selected by the user.

Event Name	Task Flow	Page	ContextualArea Name	Contextual Area Model XML Location	Generated when...	Payload Values
Customs EntryLoadedEvent	MaintainCeFlow	MaintainCe	CustomsEntry	Rms.ear → RmsViewController.war → AdfRmsImport.jar → /oracle/retail/apps/framework/uishell/config/contextualarea/CustomsEntry_ContextualAreaModel.xml	When the screen is opening from 'Custom Entry search' screen.	<ul style="list-style-type: none"> ▪ CustEntryID -- the 'Custom Entry Id' selected by the user.
LetterOfCreditHeadLoadEvent	MaintainLcAppHeadFlow	MaintainLcAppHead	LetterOfCreditHead	Rms.ear → RmsViewController.war → AdfRmsImport.jar → /oracle/retail/apps/framework/uishell/config/contextualarea/LetterOfCreditHead_ContextualAreaModel.xml	When the screen is opening from tasklist i.e. while creating of 'Letter of credit'.	<ul style="list-style-type: none"> ▪ LCReferenceNum – Letter of Credit Reference number
LetterOfCreditOrderSelectionEvent	MaintainLcOrderApplyFlow	MaintainLcOrderApply	LetterOfCreditOrder	Rms.ear → RmsViewController.war → AdfRmsImport.jar → /oracle/retail/apps/framework/uishell/config/contextualarea/LetterOfCreditOrder_ContextualAreaModel.xml	When a row in the 'Associate LCs with Orders' table is selected.	<ul style="list-style-type: none"> ▪ Applicant – the 'Applicant id' selected by the user. ▪ Bank –the 'Bank id' selected by the user. ▪ Order – the 'Order no' selected by the user.

Event Name	Task Flow	Page	ContextualArea Name	Contextual Area Model XML Location	Generated when...	Payload Values
ObligationLoadEvent	MaintainObligationFlow	MaintainObligation	Obligation	Rms.ear → RmsViewController.war → AdfRmsImport.jar → /oracle/retail/apps/framework/uishell/config/contextualarea/Obligation_ContextualAreaModel.xml	When the screen is opening from tasklist i.e. while creating of 'Create Obligation' and opening from 'Obligation Search' screen.	<ul style="list-style-type: none"> ObligationID - the 'obligation id' selected by the user.
TransportationLoadEvent	MaintainTransportFlow	MaintainTransport	Transportation	Rms.ear → RmsViewController.war → AdfRmsImport.jar → /oracle/retail/apps/framework/uishell/config/contextualarea/Transportation_ContextualAreaModel.xml	When the screen is opening from tasklist i.e. while creating of 'Create Transportation by Order' and opening from 'Transportation Search' screen.	<ul style="list-style-type: none"> TransportationID - the 'transportation id' selected by the user.

Event Name	Task Flow	Page	ContextualArea Name	Contextual Area Model XML Location	Generated when...	Payload Values
FinalizeTransportationValueChangeEvent	MaintainTransportFinalFlow	MaintainTransportFinal	FinalizeTransportation	Rms.ear → RmsViewController.war → AdfRmsImport.jar → /oracle/retail/apps/framework/uishell/config/contextualarea/FinalizeTransportation_ContextualAreaModel.xml	Upon entry of a valid Vessel, Voyage, Estimated Departure Date combination.	<ul style="list-style-type: none"> ▪ VesselID - the 'Vessel id' selected by the user. ▪ VoyageFlightID - the 'voyage flight id' selected by the user. ▪ EstDepartureDate - the 'Estimate Departure Date' selected by the user.
BuyerWorksheetSelectionEvent	MaintainBuyerWorksheetFlow	MaintainBuyerWorksheet	ContractCost	Rms.ear → RmsViewController.war → AdfRmsProcurement.jar → /oracle/retail/apps/framework/uishell/config/contextualarea/BuyerWorksheet_ContextualAreaModel.xml	When a row in the 'Worksheet Items' table is selected by the user.	<ul style="list-style-type: none"> ▪ Item - the item selected by the user. ▪ Supplier - the 'supplier id' selected by the user. ▪ SourceType - the 'source type' selected by the user. ▪ Location - the 'location id' selected by the user. ▪ DisplayCurrency - the 'display currency' selected by the user in which currency mode is viewing.

Event Name	Task Flow	Page	ContextualArea Name	Contextual Area Model XML Location	Generated when...	Payload Values
ContractItemEvent	MaintainContractFlow	MaintainContract	ContractCost	Rms.ear → RmsViewController.war → AdfRmsProcurement.jar → /oracle/retail/apps/framework/uishell/config/contextualarea/ContractItem_ContextualAreaModel.xml	The user selects an record on the 'Contract Results' table 'Contract Search' screen by the user.	<ul style="list-style-type: none"> ▪ ContractNum – the 'Contract no' selected by the user. ▪ Supplier – the 'Supplier id' selected by the user. ▪ StartDate – the 'Startdate' selected by the user. ▪ EndDate – the 'Endate' selected by the user. ▪ Status – the 'Status' selected by the user. ▪ ContractCurrency – the 'contract currency' selected by the user. i.e. the currency of the contract created. ▪ DisplayCurrency -- the 'display currency' selected by the user in which currency mode is viewing.
ContractItemEvent	MaintainContractCostFlow	MaintainContractCost		Rms.ear → RmsViewController.war → AdfRmsProcurement.jar → /oracle/retail/apps/framework/uishell/config/contextualarea/ContractItem_ContextualAreaModel.xml	When a row in the 'Contract Item' table is selected by the user.	<ul style="list-style-type: none"> ▪ ContractNum -- the 'Contract no' selected by the user. ▪ Item -- the 'item' selected by the user. ▪ ItemParent – the 'item parent' selected by the user. ▪ Diff1 – the 'diff1' selected by the user. ▪ Diff2 – the 'diff2' selected by the user. ▪ Diff3 – the 'diff3' selected by the user. ▪ Diff4 – the 'diff4' selected by the user.

Event Name	Task Flow	Page	ContextualArea Name	Contextual Area Model XML Location	Generated when...	Payload Values
ContractItemEvent	MaintainContractCost Flow	MaintainContractCost		Rms.ear → RmsViewController.war → AdfRmsProcurement.jar → /oracle/retail/apps/framework/uishell/config/contextualarea/ContractItem_ContextualAreaModel.xml	When a row in the 'Contract Item Location' table is selected by the user.	<ul style="list-style-type: none"> ▪ ContractNum -- the 'Contract no' selected by the user. ▪ Item -- the item selected by the user. ▪ Location -- the 'location id' selected by the user. ▪ ReadyDate -- the 'ready date' selected by the user.
ContractOrderItemEvent	MaintainContractOrders Flow	MaintainContractOrders	ContractItemOrderItem	Rms.ear → RmsViewController.war → AdfRmsProcurement.jar → /oracle/retail/apps/framework/uishell/config/contextualarea/ContractItemOrderItem_ContextualAreaModel.xml	When a row in the Items table is selected	<ul style="list-style-type: none"> ▪ ContractNum -- the 'Contract no' selected by the user. ▪ Item -- the item selected by the user. ▪

Event Name	Task Flow	Page	ContextualArea Name	Contextual Area Model XML Location	Generated when...	Payload Values
ContractedOrderItemEvent	MaintainCntrctOrders Flow	MaintainCntrctOrders	ContractItemOrderItem	Rms.ear → RmsViewController.war → AdfRmsProcurement.jar → /oracle/retail/apps/framework/uishell/config/contextualarea/ContractItemOrderItem_ContextualAreaModel.xml	When a row in the Locations table is selected.	<ul style="list-style-type: none"> ▪ ContractNum - the 'Contract no' selected by the user. ▪ Item - the 'item' selected by the user. ▪ Location - the 'location id' selected by the user. ▪
ContractedOrderItemEvent	MaintainCntrctOrders Flow	MaintainCntrctOrders	ContractItemOrderItem	Rms.ear → RmsViewController.war → AdfRmsProcurement.jar → /oracle/retail/apps/framework/uishell/config/contextualarea/ContractItemOrderItem_ContextualAreaModel.xml	When a row in the Orders table is selected.	<ul style="list-style-type: none"> ▪ ContractNum - the 'Contract no' selected by the user. ▪ Item - the 'item' selected by the user. ▪ Location - the 'location id' selected by the user. ▪ Supplier - the 'supplier id' selected by the user. ▪ OrderType - the 'order type' selected by the user.

Event Name	Task Flow	Page	ContextualArea Name	Contextual Area Model XML Location	Generated when...	Payload Values
CostChangeItemSelectedEvent	MaintainCostChangeByItemFlow	MaintainCostChangeByItem	CostChangeByItem	Rms.ear → RmsViewController.war → AdfRmsProcurement.jar → /oracle/retail/apps/framework/uishell/config/contextualarea/CostChgByItem_ContextualAreaModel.xml	When a row in the Item table is selected.	<ul style="list-style-type: none"> ▪ CostChangeID – the ‘Cost change id’ selected by the user. ▪ Item – the ‘item’ selected by the user. ▪ Supplier – the ‘supplier id’ selected by the user. ▪ OriginCountry – the ‘origin country id’ selected by the user. ▪ EffectiveDate – the ‘effective date’ selected by the user. ▪ DisplayCurrency – the ‘display currency’ selected by the user in which currency mode is viewing.
CostChangeItemLocationSelectedEvent	MaintainCostChangeByLocationFlow	MaintainCostChangeByLocation	CostChgByItemLoc	Rms.ear → RmsViewController.war → AdfRmsProcurement.jar → /oracle/retail/apps/framework/uishell/config/contextualarea/CostChgByItemLoc_ContextualAreaModel.xml	When a row in the Item/Location table is selected by user.	<ul style="list-style-type: none"> ▪ CostChangeID – the ‘Cost change id’ selected by the user. ▪ Item – the ‘item’ selected by the user. ▪ Location – the ‘Location id’ selected by the user. ▪ Supplier – the ‘supplier id’ selected by the user. ▪ OriginCountry – the ‘origin country id’ selected by the user. ▪ EffectiveDate – the ‘effective date’ selected by the user. ▪ DisplayCurrency – the ‘display currency’ selected by the user in which currency mode is viewing.

Event Name	Task Flow	Page	ContextualArea Name	Contextual Area Model XML Location	Generated when...	Payload Values
DealHeadEvent	MaintainDealFlow	MaintainDeal	DealHeadReportArea	Rms.ear → RmsViewController.war → AdfRmsProcurement.jar → /oracle/retail/apps/framework/uishell/config/contextualarea/DealHead_ContextualAreaModel.xml	When the screen is opened for an existing deal from 'Search Deals' screen by user.	<ul style="list-style-type: none"> ▪ DealID - the 'deal id' selected by the user. ▪ Supplier - the 'supplier id' selected by the user. ▪ DealCurrency - the 'deal currency' selected by the user. ▪ DisplayCurrency - the 'display currency' selected by the user in which currency mode is viewing.
DealHeadEvent	MaintainDealFlow	MaintainDeal	DealHeadReportArea	Rms.ear → RmsViewController.war → AdfRmsProcurement.jar → /oracle/retail/apps/framework/uishell/config/contextualarea/DealHead_ContextualAreaModel.xml	When a row in the component table is selected by user.	<ul style="list-style-type: none"> ▪ DealID - the 'deal id' selected by the user. ▪ Supplier - the 'supplier id' selected by the user. ▪ DealCompID - the 'deal comp id' selected by the user. ▪ DealCurrency - the 'deal currency' selected by the user. ▪ DisplayCurrency - the 'display currency' selected by the user in which currency mode is viewing.

Event Name	Task Flow	Page	ContextualArea Name	Contextual Area Model XML Location	Generated when...	Payload Values
FixedDealHeadEvent	MaintainFixedDealFlow	MaintainFixedDeal	DealHeadReportArea	Rms.ear → RmsViewController.war → AdfRmsProcurement.jar → /oracle/retail/apps/framework/uishell/config/contextualarea/FixedDealHead_ContextualAreaModel.xml	When the screen is opened for an existing fixed deal from 'Fixed deal search' screen by user.	<ul style="list-style-type: none"> ▪ DealID - the 'deal id' selected by the user. ▪ Supplier - the 'supplier id' selected by the user. ▪ DisplayCurrency - the 'display currency' selected by the user in which currency mode is viewing.
FixedDealHeadEvent	MaintainFixedDealFlow	MaintainFixedDeal	DealHeadReportArea	Rms.ear → RmsViewController.war → AdfRmsProcurement.jar → /oracle/retail/apps/framework/uishell/config/contextualarea/FixedDealHead_ContextualAreaModel.xml	When a supplier or partner is specified during the creation of a new fixed deal by user.	<ul style="list-style-type: none"> ▪ Supplier - the 'supplier id' selected by the user. ▪ DisplayCurrency - the 'display currency' selected by the user in which currency mode is viewing.

Event Name	Task Flow	Page	ContextualArea Name	Contextual Area Model XML Location	Generated when...	Payload Values
OrderDetailEvent	MaintainOrderDetailFlow	MaintainOrderDetail	OrderDetailItem	Rms.ear → RmsViewController.war → AdfRmsProcurement.jar → /oracle/retail/apps/framework/uishell/config/contextualarea/OrderDetail_ContextualAreaModel.xml	When a row in the item table is selected by user.	<ul style="list-style-type: none"> ▪ Order – the 'order no' selected by the user. ▪ Item – the 'item' selected by the user. ▪ ItemParent – the 'item parent' selected by the user. ▪ Diff1 – the first differentiator 'diff1' selected by the user. ▪ Diff2 – the second differentiator 'diff2' selected by the user. ▪ Diff3 – the third differentiator 'diff3' selected by the user. ▪ Diff4 – the fourth differentiator 'diff4' selected by the user. ▪ OrderCurrency – the 'order currency' selected by the user. ▪ DisplayCurrency – the 'display currency' selected by the user in which currency mode is viewing. ▪ DisplayUOM – the 'display uom' selected by the user in which UOM mode is viewing.

Event Name	Task Flow	Page	ContextualArea Name	Contextual Area Model XML Location	Generated when...	Payload Values
OrderDetailEvent	MaintainOrderDetailFlow	MaintainOrderDetail	OrderDetailItem	Rms.ear → RmsViewController.war → AdfRmsProcurement.jar → /oracle/retail/apps/framework/uishell/config/contextualarea/OrderDetail_ContextualAreaModel.xml	When a row in the item/location table is selected by user.	<ul style="list-style-type: none"> ▪ Order - the 'order no' selected by the user. ▪ Item - the 'item' selected by the user. ▪ Location- the 'Location id' selected by the user. ▪ DisplayCurrency - the 'display currency' selected by the user in which currency mode is viewing. ▪ DisplayUOM - the 'display uom' selected by the user in which UOM mode is viewing.
OrderDetailEvent	MaintainOrderDetailFlow	MaintainOrderDetail	OrderDetailItem	Rms.ear → RmsViewController.war → AdfRmsProcurement.jar → /oracle/retail/apps/framework/uishell/config/contextualarea/OrderDetail_ContextualAreaModel.xml	When a row in the allocation table is selected by user.	<ul style="list-style-type: none"> ▪ Order - the 'order no' selected by the user. ▪ Item - the 'item' selected by the user. ▪ Location- the 'Location id' selected by the user. ▪ PhysicalWh - the 'physical warehouse id' selected by the user. ▪ Allocation - the 'allocation id' selected by the user. ▪ DisplayCurrency - the 'display currency' selected by the user in which currency mode is viewing. ▪ DisplayUOM - the 'display uom' selected by the user in which UOM mode is viewing.

Event Name	Task Flow	Page	ContextualArea Name	Contextual Area Model XML Location	Generated when...	Payload Values
OrderDistItemSelectionEvent	MaintainOrderDistributionFlow	MaintainOrderDistribution	OrderDistItem	Rms.ear → RmsViewController.war → AdfRmsProcurement.jar → /oracle/retail/apps/framework/uishell/config/contextualarea/OrderDistItem_ContextualAreaModel.xml	When a row in the distribution table is selected by user.	<ul style="list-style-type: none"> ▪ Order – the 'order no' selected by the user. ▪ Supplier – the 'supplier id' selected by the user. ▪ Item – the 'item' selected by the user. ▪ Location – the 'Location id' selected by the user.
OrderHeaderLoadEvent	MaintainPurchaseOrderFlow	MaintainPurchaseOrder	DealHeaderReportArea	Rms.ear → RmsViewController.war → AdfRmsProcurement.jar → /oracle/retail/apps/framework/uishell/config/contextualarea/OrderHeader_ContextualAreaModel.xml	When an existing order is opening from 'Order Search' screen by user.	<ul style="list-style-type: none"> ▪ Order – the 'order no' selected by the user. ▪ Supplier – the 'supplier id' selected by the user. ▪ ImportInd – the 'import indicator' selected by the user. ▪ Department – the 'department id' selected by the user. ▪ Location – the 'location id' selected by the user. ▪ Status – the 'status' of order selected by the user. ▪ OrderCurrency – the 'order currency' selected by the user. ▪ OrderType – the 'order type' selected by the user.

Event Name	Task Flow	Page	ContextualArea Name	Contextual Area Model XML Location	Generated when...	Payload Values
OrderSearchOrderSelectedEvent	SearchOrderFlow	SearchOrder	DealHeadReportArea	Rms.ear → RmsViewController.war → AdfRmsProcurement.jar → /oracle/retail/apps/framework/uishell/config/contextualarea/OrderSearch_ContextualAreaModel.xml	When a row in the table 'Order details' selected by user.	<ul style="list-style-type: none"> ▪ Order - the 'order no' selected by the user. ▪ Supplier - the 'supplier id' selected by the user. ▪ Status - the 'status' of order selected by the user. ▪ Currency - the 'currency' selected by the user.
ReceivingCostAdjItemSelectionEvent	MaintainReceivingCostAdjFlow	MaintainReceivingCostAdj	ContractCost	Rms.ear → RmsViewController.war → AdfRmsProcurement.jar → /oracle/retail/apps/framework/uishell/config/contextualarea/ReceiverCostAdjItem_ContextuaAreaModel.xml	When a row is selected by user	<ul style="list-style-type: none"> ▪ Order - the 'order no' selected by the user. ▪ Supplier - the 'supplier id' selected by the user. ▪ Department Department - the 'department id' selected by the user. ▪ Item - the 'item' selected by the user. ▪ Location - the 'location id' selected by the user. ▪ DisplayCurrency - the 'display currency' selected by the user in which currency mode is viewing.

Event Name	Task Flow	Page	ContextualArea Name	Contextual Area Model XML Location	Generated when...	Payload Values
Franchise OrderEvent	MaintainFranchiseOrderFlow	MaintainFranchiseOrder	FranchiseOrderHead	Rms.ear → RmsViewController.war → AdfRmsInventory.jar → /oracle/retail/apps/framework/uishell/config/contextualarea/FranchiseOrderHead_ContextualAreaModel.xml	When 'Franchise Order' page is opening from 'Franchise Order Search' screen by user.	<ul style="list-style-type: none"> ▪ WfOrderNo - the 'franchise order no' selected by the user. ▪ Customer - the 'customer id' selected by the user.
Franchise OrderEvent	MaintainFranchiseOrderFlow	MaintainFranchiseOrder	FranchiseOrderHead	Rms.ear → RmsViewController.war → AdfRmsInventory.jar → /oracle/retail/apps/framework/uishell/config/contextualarea/FranchiseOrderHead_ContextualAreaModel.xml	When row in the 'Franchise Orders' table is selected by user.	<ul style="list-style-type: none"> ▪ WfOrderNo - the 'franchise order no' selected by the user. ▪ Item - the 'item' selected by the user. ▪ SourceLocation - the 'source location' selected by the user. ▪ CustomerLoc - the 'customer location' selected by the user.

Event Name	Task Flow	Page	ContextualArea Name	Contextual Area Model XML Location	Generated when...	Payload Values
Franchise ReturnHeadLoadEvent	MaintainFranchiseReturnsFlow	MaintainFranchiseReturns	FranchiseReturnHead	Rms.ear → RmsViewController.war → AdfRmsInventory.jar → /oracle/retail/apps/framework/uishell/config/contextualarea/FranchiseReturnHead_ContextualAreaModel.xml	When 'Franchise Return' page is opening from 'Franchise Search' screen.	<ul style="list-style-type: none"> ▪ RMANum - the 'RMANum' selected by the user.
Franchise ReturnItemSelectionEvent	MaintainFranchiseReturnsFlow	MaintainFranchiseReturns	FranchiseReturnHead	Rms.ear → RmsViewController.war → AdfRmsInventory.jar → /oracle/retail/apps/framework/uishell/config/contextualarea/FranchiseReturnHead_ContextualAreaModel.xml	When row in 'Franchise Returns' table is selected by user.	<ul style="list-style-type: none"> ▪ WfOrderNo - the 'franchise order no' selected by the user. ▪ Item - the 'item' selected by the user.

Event Name	Task Flow	Page	ContextualArea Name	Contextual Area Model XML Location	Generated when...	Payload Values
ItemLocInventory Selection Event	SearchItemLocInvFlow	SearchItemLocInv	ItemLocInventory	Rms.ear → RmsViewController.war → AdfRmsInventory.jar → /oracle/retail/apps/framework/uishell/config/contextualarea/ItemLocInventory_ContextualAreaModel.xml	When row in table is selected by the user.	<ul style="list-style-type: none"> ▪ Item – the 'Item' selected by the user. ▪ Location – the 'location' selected by the user.
MassReturnTransferEvent	MaintainMassReturnTransferFlow	MaintainMassReturnTransfer	MassReturnTransfer	Rms.ear → RmsViewController.war → AdfRmsInventory.jar → /oracle/retail/apps/framework/uishell/config/contextualarea/MassReturnTransferHead_ContextualAreaModel.xml	When screen opens for an existing MRT	<ul style="list-style-type: none"> ▪ MRT – the 'mass return transfer no' selected by the user. ▪ Warehouse – the 'warehouse id' selected by the user. ▪ Status – the 'status' selected by the user. ▪ Supplier – the 'supplier id' selected by the user. ▪ DisplayCurrency – the 'display currency' selected by the user in which currency mode is viewing.

Event Name	Task Flow	Page	ContextualArea Name	Contextual Area Model XML Location	Generated when...	Payload Values
MassReturnTransferEvent	MaintainMassReturnTransferFlow	MaintainMassReturnTransfer	MassReturnTransfer	Rms.ear → RmsViewController.war → AdfRmsInventory.jar → /oracle/retail/apps/framework/uishell/config/contextualarea/MassReturnTransferHead_ContextualAreaModel.xml	When a row in the Items table is selected	<ul style="list-style-type: none"> ▪ MRT - the 'mass return transfer no' selected by the user. ▪ Item - the 'item' selected by the user. ▪ DisplayCurrency - the 'display currency' selected by the user in which currency mode is viewing.
MassReturnTransferEvent	MaintainMassReturnTransferFlow	MaintainMassReturnTransfer	MassReturnTransfer	Rms.ear → RmsViewController.war → AdfRmsInventory.jar → /oracle/retail/apps/framework/uishell/config/contextualarea/MassReturnTransferHead_ContextualAreaModel.xml	When a row in the Locations table is selected	<ul style="list-style-type: none"> ▪ MRT - the 'mass return transfer no' selected by the user. ▪ Item - the 'item' selected by the user. ▪ Location - the 'location id' selected by the user. ▪ DisplayCurrency - the 'display currency' selected by the user in which currency mode is viewing.

Event Name	Task Flow	Page	ContextualArea Name	Contextual Area Model XML Location	Generated when...	Payload Values
ReplAttributeLoadEvent	MaintainReplAttrFlow	MaintainReplAttr	ReplAttribute	Rms.ear → RmsViewController.war → AdfRmsInventory.jar → /oracle/retail/apps/framework/uishell/config/contextualarea/ReplAttribute_ContextualAreaModel.xml	When 'Replenishment Attributes' page is opening from 'Replenishment Search' screen selected by the user.	<ul style="list-style-type: none"> Item - the 'item' selected by the user. Location - the 'location id' selected by the user.
ReplAttributeSearchSelectionEvent	SearchReplAttrFlow	SearchReplAttr	ReplAttributeSearch	Rms.ear → RmsViewController.war → AdfRmsInventory.jar → /oracle/retail/apps/framework/uishell/config/contextualarea/ReplAttributeSearch_ContextualAreaModel.xml	When row in results 'Replenishment Attributes' table is selected by the user.	<ul style="list-style-type: none"> Item - the 'item' selected by the user. Location - the 'location id' selected by the user.

Event Name	Task Flow	Page	ContextualArea Name	Contextual Area Model XML Location	Generated when...	Payload Values
ReturnToVendorEvent	MaintainRtvDetailFlow	MaintainRtvDetail	ReturnToVendor	Rms.ear → RmsViewController.war → AdfRmsInventory.jar → /oracle/retail/apps/framework/uishell/config/contextualarea/ReturnToVendorHead_ContextualAreaModel.xml	When screen opens for an existing RTV from 'Return to Vendor Search' screen.	<ul style="list-style-type: none"> ▪ RTV - the 'return to vendor id' selected by the user. ▪ Location - the 'location id' selected by the user. ▪ Supplier - the 'supplier id' selected by the user. ▪ Status - the 'status' of return to vendor selected by the user. ▪ DisplayCurrency - the 'display currency' selected by the user in which currency mode is viewing. ▪ DisplayUOM - the 'display Unit Of measure' selected by the user in which UOM is viewing.

Event Name	Task Flow	Page	ContextualArea Name	Contextual Area Model XML Location	Generated when...	Payload Values
ReturnToVendorEvent	MaintainRtvDetailFlow	MaintainRtvDetail	ReturnToVendor	Rms.ear → RmsViewController.war → AdfRmsInventory.jar → /oracle/retail/apps/framework/uishell/config/contextualarea/ReturnToVendorHead_ContextualAreaModel.xml	When a row in the Return Details table is selected by user.	<ul style="list-style-type: none"> ▪ RTV – the ‘return to vendor id’ selected by the user. ▪ Item – the ‘item’ selected by the user. ▪ Location – the ‘location id’ selected by the user. ▪ Supplier – the ‘supplier id’ selected by the user. ▪ DisplayCurrency – the ‘display currency’ selected by the user in which currency mode is viewing. ▪ InventoryStatus– the ‘status’ of return to vendor selected by the user. ▪ DisplayUOM – the ‘display Unit Of measure’ selected by the user in which UOM is viewing.
SalesIssuesByLocationEvent	SearchSalesIssueFlow	SearchSalesIssue	SalesIssueByLoc	Rms.ear → RmsViewController.war → AdfRmsInventory.jar → /oracle/retail/apps/framework/uishell/config/contextualarea/SalesIssuesByLoc_ContextualAreaModel.xml	When row in ‘sales issues by location’ table is selected by user.	<ul style="list-style-type: none"> ▪ Item – the ‘item’ selected by the user. ▪ Location – the ‘location id’ selected by the user.

Event Name	Task Flow	Page	ContextualArea Name	Contextual Area Model XML Location	Generated when...	Payload Values
ShipmentPageEvent	MaintainShipmentDetailFlow	MaintainShipmentDetail	ShipmentHead	Rms.ear → RmsViewController.war → AdfRmsInventory.jar → /oracle/retail/apps/framework/uishell/config/contextualarea/ShipmentHead_ContextualAreaModel.xml	When the screen is opening of 'Shipment Search' screen	<ul style="list-style-type: none"> ▪ Shipment - the 'shipment id' selected by the user. ▪ ASN - the 'asn no' selected by the user. ▪ BOL - the 'bol no' selected by the user. ▪ EstArrivalDate - the 'estimated arrival date' selected by the user. ▪ InvcMatchDate - the 'invoice match date' selected by the user. ▪ ReceiveDate - the 'receive date' selected by the user. ▪ ShipDate - the 'ship date' selected by the user. ▪ NotBeforeDate - the 'not before date' selected by the user. ▪ NotAfterDate - the 'not after date' selected by the user. ▪ ToLoc - the 'to location' selected by the user. ▪ Status - the 'status' of the shipment selected by the user.

Event Name	Task Flow	Page	ContextualArea Name	Contextual Area Model XML Location	Generated when...	Payload Values
ShipmentPageEvent	MaintainShipmentDetailFlow	MaintainShipmentDetail	ShipmentHead	Rms.ear → RmsViewController.war → AdfRmsInventory.jar → /oracle/retail/apps/framework/uishell/config/contextualarea/ShipmentHead_ContextualAreaModel.xml	When a row in the Details table is selected by user.	<ul style="list-style-type: none"> ▪ Shipment - the 'shipment id' selected by the user. ▪ Item - the 'item' selected by the user. ▪ Carton - the 'carton id' selected by the user.
ShipmentSearchSelectionEvent	SearchShipmentFlow	SearchShipment	ShipmentSearch	Rms.ear → RmsViewController.war → AdfRmsInventory.jar → /oracle/retail/apps/framework/uishell/config/contextualarea/ShipmentSearch_ContextualAreaModel.xml	When a row is selected by the user in 'Shipment Search Results' table.	<ul style="list-style-type: none"> ▪ Shipment - the 'shipment id' selected by the user. ▪ Order - the 'order no' selected by the user. ▪ BOL - the 'bol no' selected by the user. ▪ ToLoc - the 'to location' selected by the user.

Event Name	Task Flow	Page	ContextualArea Name	Contextual Area Model XML Location	Generated when...	Payload Values
StockCountResults Selection Event	MaintainStockCountResultStkvwedtkwFlow	MaintainStockCountResultStkvwedt	StockCountResults	Rms.ear → RmsViewController.war → AdfRmsInventory.jar → /oracle/retail/apps/framework/uishell/config/contextualarea/StockCountResults_ContextualAreaModel.xml	When a row in the 'item and location' table selected by user.	<ul style="list-style-type: none"> ▪ StockCountID - the 'stock count id' selected by the user. ▪ Item - the 'item' selected by the user. ▪ Location - the 'location id' selected by the user.
StockOrderExceptionEvent	MaintainStockOrderReconciliationFlow	MaintainStockOrderReconciliation	StockOrderExceptionDetail	Rms.ear → RmsViewController.war → AdfRmsInventory.jar → /oracle/retail/apps/framework/uishell/config/contextualarea/StockOrderExceptionDetail_ContextualAreaModel.xml	When a row in the Exception table is selected by user.	<ul style="list-style-type: none"> ▪ BOL - the 'bol no' selected by the user. ▪ FromLoc - the 'from location id' selected by the user. ▪ ToLoc - the 'to location id' selected by the user. ▪ Shipment - the 'shipment id' selected by the user.

Event Name	Task Flow	Page	ContextualArea Name	Contextual Area Model XML Location	Generated when...	Payload Values
StockOrderExceptionEvent	MaintainStockOrderReconciliationFlow	MaintainStockOrderReconciliation	StockOrderExceptionDetail	Rms.ear → RmsViewController.war → AdfRmsInventory.jar → /oracle/retail/apps/framework/uishell/config/contextualarea/StockOrderExceptionDetail_ContextualAreaModel.xml	When a row in the Carton table is selected by user.	<ul style="list-style-type: none"> ▪ Carton - the 'carton id' selected by the user. ▪ Shipment - the 'shipment id' selected by the user. ▪ BOL - the 'bol id' selected by the user. ▪ FromLoc - the 'from location id' selected by the user. ▪ ToLoc - the 'to location id' selected by the user.

Event Name	Task Flow	Page	ContextualArea Name	Contextual Area Model XML Location	Generated when...	Payload Values
Transfer DetailSelectionEvent	MaintainTransferDetailFlow	MaintainTransferDetail	MaintainTransferContextAwareReport	Rms.ear → RmsViewController.war → AdfRmsInventory.jar → /oracle/retail/apps/framework/uishell/config/contextualarea/TransferDetail_ContextualAreaModel.xml	When a row in the Items table is selected by user.	<ul style="list-style-type: none"> ▪ Transfer – the 'transfer id' selected by the user. ▪ Item – the 'item' selected by the user. ▪ ItemParent – the 'item parent' selected by the user. ▪ Diff1 – the 'diff1' selected by the user. ▪ Diff2 – the 'diff2' selected by the user. ▪ Diff3 – the 'diff3' selected by the user. ▪ Diff4 – the 'diff4' selected by the user. ▪ InventoryStatus – the 'inventory status' selected by the user. ▪ FromLoc – the 'from location id' selected by the user. ▪ ToLoc – the 'to location id' selected by the user. ▪ Finisher – the 'finisher id' selected by the user. ▪ DisplayCurrency – the 'display currency' selected by the user in which currency mode is viewing. ▪ DisplayUOM – the 'display unit of measure' selected by the user in which unit of measuring is viewing. ▪ DisplayLeg – the 'display leg' selected by the user in which leg is viewing.

Event Name	Task Flow	Page	ContextualArea Name	Contextual Area Model XML Location	Generated when...	Payload Values
TransferHeadLoadedEvent	MaintainTransferHeaderFlow	MaintainTransferHeader	TransferHead	Rms.ear → RmsViewController.war → AdfRmsInventory.jar → /oracle/retail/apps/framework/uishell/config/contextualarea/TransferHead_ContextualAreaModel.xml	When screen opens for an existing transfer from 'transfer search' screen.	<ul style="list-style-type: none"> ▪ Transfer - the 'transfer id' selected by the user. ▪ FromLoc - the 'from location id' selected by the user. ▪ ToLoc - the 'to location id' selected by the user. ▪ Finisher - the 'finisher id' selected by the user. ▪ Status - the 'status' of the transfer selected by the user. ▪ DeliveryDate - the 'delivery date' selected by the user.
TransferSearchSelectionEvent	SearchTransferFlow	SearchTransfer	TransferSearch	Rms.ear → RmsViewController.war → AdfRmsInventory.jar → /oracle/retail/apps/framework/uishell/config/contextualarea/TransferSearch_ContextualAreaModel.xml	When a row is selected in 'transfer search' result table.	<ul style="list-style-type: none"> ▪ Transfer - the 'transfer id' selected by the user. ▪ FromLoc - the 'from location id' selected by the user. ▪ ToLoc - the 'to location id' selected by the user. ▪ Finisher - the 'finisher id' selected by the user. ▪ Status - the 'status' of the transfer selected by the user.

Event Name	Task Flow	Page	ContextualArea Name	Contextual Area Model XML Location	Generated when...	Payload Values
UnitVarianceAndAdjustmentEvent	MaintainStockCountVarianceFlow	MaintainStockCountVariance	ValueVarAdjLoc	Rms.ear → RmsViewController.war → AdfRmsInventory.jar → /oracle/retail/apps/framework/uishell/config/contextualarea/UnitVarianceAndAdjustment_ContextualAreaModel.xml	When row in hierarchy table is selected by user.	<ul style="list-style-type: none"> ▪ StockCountID - the 'stock count id' selected by the user. ▪ Location - the 'location id' selected by the user. ▪ Item -- the 'item' selected by the user.
UnitVarianceAndAdjustmentEvent	MaintainStockCountVarianceFlow	MaintainStockCountVariance	ValueVarAdjLoc	Rms.ear → RmsViewController.war → AdfRmsInventory.jar → /oracle/retail/apps/framework/uishell/config/contextualarea/UnitVarianceAndAdjustment_ContextualAreaModel.xml	When row in table is selected by user.	<ul style="list-style-type: none"> ▪ StockCountID - the 'stock count id' selected by the user. ▪ Location - the 'location id' selected by the user. ▪ Department-- the 'Department' selected by the user. ▪ Class - the 'class id' selected by the user. ▪ SubClass - the 'SubClass id' selected by the user.

Event Name	Task Flow	Page	ContextualArea Name	Contextual Area Model XML Location	Generated when...	Payload Values
ValueVarAdjLocSelectionEvent	MaintainStockCountDollarChangeFlow	MaintainStockCountDollarChange	ValueVarAdjLoc	Rms.ear → RmsViewController.war → AdfRmsInventory.jar → /oracle/retail/apps/framework/uishell/config/contextualarea/ValueVarianceAndAdjustment_ContextualAreaModel.xml	When row in location table is selected by user.	<ul style="list-style-type: none"> ▪ StockCountID - the 'stock count id' selected by the user. ▪ Location - the 'location id' selected by the user.
VirtualWhDistSelectionEvent	MaintainStockCountVWHDistributionFlow	MaintainStockCountVWHDistribution	VirtualWhDist	Rms.ear → RmsViewController.war → AdfRmsInventory.jar → /oracle/retail/apps/framework/uishell/config/contextualarea/VirtualWhDist_ContextualAreaModel.xml	When item and physical warehouse selected by user.	<ul style="list-style-type: none"> ▪ StockCountID - the 'stock count id' selected by the user. ▪ Item -- the 'item' selected by the user. ▪ PhysicalWh- the 'physical warehouse id' selected by the user.

Preparing the Custom Shared Library for Adding Contextual Reports

This section describes how to prepare the Custom Shared Library so retailers will be able to add contextual reports in Retail Application task flows.

1. Perform the steps described in the section Add a Custom Shared Library to create a Custom Shared Library workspace, generate a shared library WAR, deploy the WAR, and associate the library to the Retail Application.
2. Using JDeveloper, open the Custom Shared Library workspace in Developer Role.
3. Add new contents as required.
4. Regenerate the shared library WAR file from the workspace and redeploy the shared library. A shutdown and restart of the Retail Application and its shared library registry is required.
5. Changes required in Shared Library Registry for Custom Contextual Reports
 - a. Obtain a copy of the task flow contextual area model XML files where contextual reports will be added. Refer to the section, List of Contextual Business Events and Payloads.

Example: If the contextual area model XML for the task flow AllocMaintFlow is called AllocMaintFlowContextualAreaModel.xml – then that file’s path must be View -Controller/src/custom/AllocMaintFlowContextualAreaModel.xml.
 - b. See the section Reference the Custom Shared Library from the Retail Application to register custom contextual reports shared library with in the Retail Applications.
 - c. Open the original WAR file using an archive file manager and copy the contextual area model XML file in the View-Controller project src directory, preferably under a subdirectory called custom.
 - d. Add a new or open the existing file called PageTemplateOverrideModel.properties under the View-Controller/src directory. Modify this file and add the following entry:

```
<ContextualAreaName>.contextualAreaModel=<path to the contextual area model for the flow>
```

Example:
AllocMaintFlow.sidebarModel=/custom/AllocMaintFlowContextualAreaModel.xml
6. Test the Retail Application. Navigate to the flow and make sure the flow is functional.

Adding a URL Based Contextual Report

Adding a contextual report to a task flow entails the modification of the task flow’s Contextual Area Model XML file. Multiple reports can be added to the model. Each report is rendered in a collapsible panel box.

Before adding a URL based contextual report, the retailer must have:

- Built, prepared, deployed, and tested the Custom Shared Library as described in the section, Preparing the Custom Shared Library for Adding Contextual Reports.
- Obtained information about the Retail Application’s list of contextual business events that can be broadcast from various workflows.
- Created one or more contextual BI reports in the BI reporting tool (for example, Oracle BI EE).
 - The web URL for each report must be available in order to proceed with the steps in this section.

- Any parameters to configure the content of the report must be known and should be accessible as parameters to the dashboard’s URL.

Once the above pre-requisites have been satisfied, proceed with the following steps:

1. Assume the following example scenario when following the steps:
 - A contextual report called “Item Metrics” showing information about an item is to be added to the Allocation Maintenance Flow’s main page. When the user selects an item on the page, the report will show information for the selected item.
 - The retail Application has provided the following information about the contextual business event:

Event Name	Task Flow	Page	Contextual Area Model XML Location	Generated when...	Payload Values
AllocMaintItemSelectedEvent	AllocMaintFlow	AllocMaintPage	EAR name → war name → jar name → /oracle/retail/apps/xyz/AllocMaintFlowContextualAreaModel.xml	The user selects an item on the Allocation Maintenance Flow’s main screen.	<ul style="list-style-type: none"> ▪ selectedItem – the item selected by the user. <i>Always generated.</i> ▪ selectedItemType – the type of item. Can be “regular” or “pack”. <i>Optional.</i> If not supplied, assume item is a regular type.

- The Item Metrics report was built in the retailer’s BI reporting tool (for example, Oracle BI EE). It was built considering the payload values the contextual business event will generate.
2. Open the Custom Shared Library workspace in JDeveloper.
 3. Open the task flow contextual area model XML file (ex. ViewController/src/custom/AllocMaintFlowContextualAreaModel.xml).
 4. Add an <Item> element within the topmost <Items> element that references the task flow called ViewContextAwareReportFlow. The ViewContextAwareReportFlow is a framework for rendering URL based reports that will be aware of contextual business events emanating from the Retail Application task flows.

Example:

```
<?xml version="1.0" encoding="UTF-8" standalone="yes"?>
<NavigationDefinition ... >
  <Items>
    ...
    ...
    <Item id="showItemMetric"
          type="taskflow"
          title="Item Metric">
      <url>
/WEB-
INF/oracle/retail/apps/framework/contextawarereport/publicui/flow/ViewContextAwareReportFlow.xml#ViewContextAwareReportFlow
      </url>
    </Item>

  </Items>
</NavigationDefinition>
```

Note the following:

- Make sure that the <Item> id is unique.
- Make sure the <Item> type is “taskflow”
- Provide a meaningful title.

5. Populate the parameters to the ViewContextAwareReportFlow by adding the following <Parameter>/<Parameters> elements.

```
<Parameter>/<Parameters> elements.
<?xml version="1.0" encoding="UTF-8" standalone="yes"?>
<NavigationDefinition ... >
  <Items>

    <Item id="showCustomerMetric"
          type="taskflow"
          title="Customer Metric">
      <url>
/WEB-
INF/oracle/retail/apps/framework/contextawarereport/publicui/flow/ViewContextAwareReportFlow.xml#ViewContextAwareReportFlow
      </url>

      <Parameters>
        <Parameter id="reportDescription">Item Metric</Parameter>
        <Parameter id="actionType">AllocMaintItemSelectedEvent</Parameter>
        <Parameter id="primaryUrl">
<![CDATA[/faces/oracle/retail/apps/framework/contextawarereport/publicui/page/ViewContextAwareReportTestPage.jspx?paramItemId=<selectedItemId>&paramItemType=<selectedItemType:token01>&paramLanguage=<language>]]>
        </Parameter>
        <Parameter id="token01">regular</Parameter>
      </Parameters>

    </Item>

  </Items>
</NavigationDefinition>
```

Note the following:

- The `<Parameter id="reportDescription">` element is the title of the contextual area report. Set this to a meaningful value.
 - The `<Parameter id="actionType">` element indicates the contextual business event name the report will listen to.
 - The `<Parameter id="primaryUrl">` element indicates the URL to the contextual area report in the BI server. The entire URL must be marked as character data (for example: enclosed in CDATA). Note that the parameters to the URL are *tokenized*. The example above uses a test page called `ViewContextAwareReportTestPage.jspx` which can be replaced with the actual report URL.
 - The `"?paramItemId=<selectedItemId>"` portion of the URL instructs the system to pass the contextual business event payload value called `selectedItemId` into the URL parameter `paramItemId` when rendering the contextual report.
 - The `"?paramItemType=<selectedItemType:token01>"` portion of the URL instructs the system to pass the contextual business event payload value called `selectedItemType` into the URL parameter `paramItemType` when rendering the contextual report. If that payload value is empty or null at runtime, then a default value of `regular` is used as referenced in a `<Parameter id="token01">` entry. *The colon symbol separates the payload value from the default value if the payload value is null.*
 - The `"?paramLanguage=<language>"` portion of the URL instructs the system about the current locale of the user. The `"language"` identifier is a reference to a value in the contextual event payload. This is a built-in value that all Retail Application contextual business event payloads will have.
 - The `<Parameter id="token01">` element holds the default value for the URL parameter `selectedItemType`. Token parameters hold default values and you can define up to 20 default value tokens
6. Re-generate the Custom Shared Library WAR file from the Custom Shared Library workspace. Shut down the Retail Application and its Custom Shared Library Registry, redeploy the Custom Shared Library WAR file, and restart the Retail Application components.
 7. Test the Retail Application. Go to the flow where the report was added and verify that the report is rendered correctly

Adding a DVT Taskflow-based Contextual Report

Before adding a DVT taskflow based contextual report, the retailer must have:

- Built, prepared, deployed, and tested the Custom Shared Library as described in the section *Preparing the Custom Shared Library for Adding Contextual Reports*.
- Obtained information about the Retail Application's business events that can be broadcast from various workflows.
- Created one or more taskflows using the ADF DVT components.
 - Any parameters to configure the content of the report must be known and should be accessible as taskflow parameters.

Once the above pre-requisites have been satisfied, proceed with the following steps:

1. Assume the following example scenario when following the steps:
 - A contextual report called "Item Metrics" showing information about an item is to be added to the Allocation Maintenance Flow's main page.

- When the user selects an item on the page, the report will show information for the selected item.
- Retail Application has provided the following information about the contextual business event:

Event Name	Task Flow	Page	Contextual Area Model XML Location	Generated when...	Payload Values
AllocMaintItemSelectedEvent	AllocMaintFlow	AllocMaintPage	EAR name → war name → jar name → /oracle/retail/apps/xyz/AllocMaintFlowContextualAreaModel.xml	The user selects an item on the Allocation Maintenance Flow's main screen.	<ul style="list-style-type: none"> ▪ selectedItem - the item selected by the user. <i>Always generated.</i> ▪ selectedItemType - the type of item. Can be "regular" or "pack". <i>Optional.</i> If not supplied, assume item is a regular type.

- The Item Metrics report was built in ADF. It was built considering the payload values the contextual business event will generate.
2. Re-generate the Custom Shared Library WAR file from the Custom Shared Library workspace. Shutdown the Retail Application and its Custom Shared Library Registry, redeploy the Custom Shared Library WAR file, and restart the Retail Application components.
 3. Changes Required in Shared Library Registry for Custom DVT Taskflow based Contextual Report
 - a. Obtain the task flow contextual area model XML file (For example: ViewController/src/custom/AllocMaintFlowContextualAreaModel.xml).
 - b. Add an <Item> element within the topmost <Items> element that references the task flow called DVTContextAwareReportFlow. The DVTContextAwareReportFlow is a framework for rendering ADF DVT based reports that will be aware of contextual business events emanating from the Retail Application task flows.

Example:

```
<?xml version="1.0" encoding="UTF-8" standalone="yes"?>
<NavigationDefinition ... >
  <Items>
    ...
    ...
    <Item id="showItemMetric"
      type="taskflow"
      title="Item Metric">
      <url>
/WEB-INF/oracle/retail/apps/framework/dvtcontextawarereport/publicui/flow/DVTContextAwareReportFlow.xml#DVTContextAwareReportFlow
      </url>
    </Item>
  </Items>
</NavigationDefinition>
```

Note

Make sure that the <Item> id is unique.

Make sure the <Item> type is “taskflow”

Provide a meaningful title.

c. Populate the parameters to the DVTContextAwareReportFlow by adding the following <Parameter>/<Parameters> elements

```
<NavigationDefinition ... >
  <Items>

    <Item id="showCustomerMetric"
      type="taskflow"
      title="Customer Metric">
      <url>
/WEB-INF/oracle/retail/apps/framework/dvtcontextawarereport/publicui/flow/DVTContextAwareReportFlow.xml#DVTContextAwareReportFlow
      </url>
      <Parameters>
        <Parameter id="taskflowURL"/>/WEB-
INF/oracle/retail/apps/framework/uishell/navigation/contextualarea/flow/ItemMetric
Flow.xml#ItemMetricFlow</Parameter>
        <Parameter id="parameterName1">itemId</Parameter>
        <Parameter id="payloadKeyName1">ItemId</Parameter>
        <Parameter
id="parameterValue1">#{payload.valueMap['ItemId']}</Parameter>
        <Parameter id="refreshOnDisclosure">#{true}</Parameter>
      </Parameters>
    </Item>

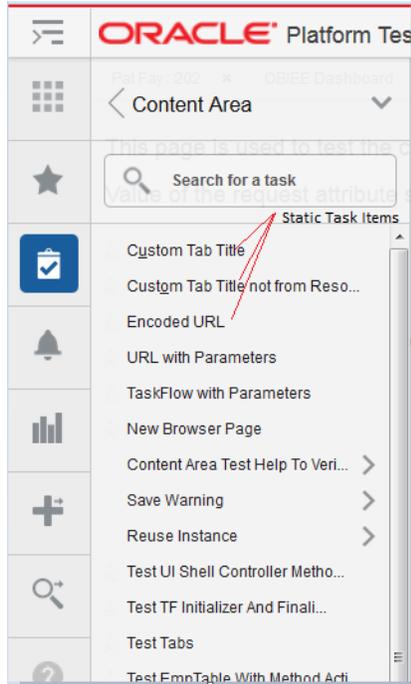
  </Items>
</NavigationDefinition>
```

Note the following:

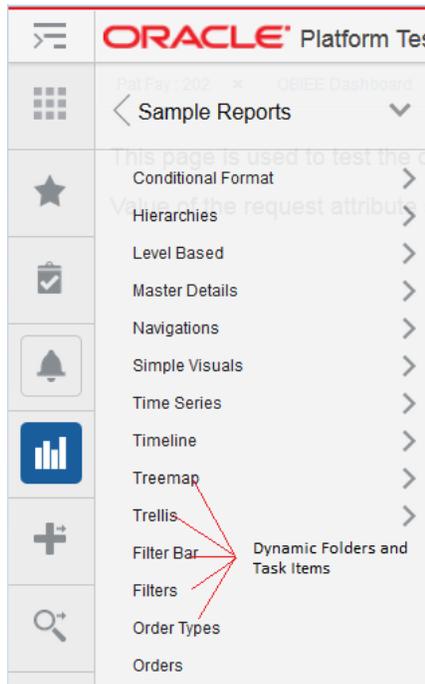
- The <Parameter id="taskflowURL"> element is the URL of the DVT taskflow that will be displayed in the contextual area.
 - The <Parameter id="parameterName1"> element indicates a parameter for the DVT taskflow. A maximum of 15 parameters can be specified.
 - The <Parameter id="payloadKeyName1"> element indicates the name of the payload key that maps to the taskflow parameter parameterName1. A maximum of 15 payload keys can be specified. The value of the payload key in the event payload will be used for the parameter in the DVT taskflow.
 - The <Parameter id="parameterValue1"> element holds the default value of the DVT taskflow. A maximum of 15 parameter values can be specified.
 - The <Parameter id="refreshOnDisclosure"> element is used to refresh the DVT taskflow on the disclosure of the tab
4. Test the Retail Application. Go to the flow where the report was added and verify that the report is rendered correctly.

Enabling Dynamic Task Items in the Retail Application

The Retail Application provides the capability to display both static and dynamic tasks in the task list. The static tasks are based off an xml model file, for example, HomeTaskMenuModel.xml, HomeReportsMenuModel.xml. These tasks should be predefined in the tasks or reports menu model file.



The dynamic tasks can be sourced from any data source such as OBIEE, BIPublisher, database tables, Web Services, and so on. The complete hierarchy of task items including the folders and actions can be generated dynamically at runtime.



From the user's perspective, there is no difference between the static versus the dynamic task items. They look and behave the same.

The DynamicContentHandler Interface

The Retail Application provides an interface called `DynamicContentHandler` that can be implemented to add dynamic tasks in the task menu. The interface is as follows:

```

package oracle.retail.apps.framework.uishell.taskmenu.dynamiccontent.handler;
import java.util.List;
import
oracle.retail.apps.framework.uishell.taskmenu.dynamiccontent.dataobject.TaskMenuIt
em;
import
oracle.retail.apps.framework.uishell.util.RetailUiShellHttpSessionListeners;

public interface DynamicContentHandler extends
RetailUiShellHttpSessionListeners.RemovalListener {
    public List<TaskMenuItem> getChildren(TaskMenuItem taskMenuItem) throws
Exception;

    public List<TaskMenuItem> getDefaultTaskMenuItems() throws Exception;

    public TaskMenuItem getContextTaskMenuItem(String taskMenuItemId) throws
Exception;

```

The interface extends another interface called

`RetailUiShellHttpSessionListeners.RemovalListener`

The `RemovalListener` interface provides the following method.

```

public static interface RemovalListener extends Serializable {
    public void onSessionRemove();
}

```

The below table explains the use of these methods.

<code>getChildren(TaskMenuItem taskMenuItem)</code>	This method provides the child task items for a given task menu item.
<code>getDefaultTaskMenuItems()</code>	This method provides the task items that will be loaded as the default content in the Retail Application.
<code>getContextTaskMenuItem(String taskMenuItemId)</code>	This method provides the task item based on a given task menu id for In-Context launching.
<code>onSessionRemove()</code>	This method will be called by the Retail Application when the user logs out of the application. The method can be used to do any resource cleanup or can be used to logout from the external data source.

DynamicContent Type

The Item element of the menu model file supports a type called 'dynamicContent' along with the type's taskflow and link. When the Retail Application encounters the type as `dynamicContent`, it looks for the attribute called `dynamicContentHandler`. The `dynamicContentHandler` attribute will tell the Retail Application the name of the implementation class which will be used to provide the dynamic task items.

The class should implement the `DynamicContentHandler` interface.

```
<Item id="customDynamicMenuFolder" visible="{true}" title="Custom Dynamic Menu"
      type="folder">
  <Items>
    <Item id="customTreeModelDynamicContent" title="Custom Dynamic
Menu"
          type="dynamicContent"

dynamicContentHandler="oracle.retail.apps.framework.uishell.taskmenu.handler.custo
m.CustomTreeModelDynamicContentHandler"
          visible="{securityContext.authenticated}"/>
  </Items>
</Item>
```

The `Item` element with the `type` attribute as `dynamicContent` can be placed anywhere in the hierarchy of the static task items in the tasks or reports menu model file. The dynamic task items will begin exactly from the location where the `dynamicContent` `Item` is defined.

The menu model file can be customized to add an entry for the dynamic content as shown above. Refer to the section [Adding or Modifying an Item in the Tasks Menu](#) for more details.

Example Implementation of the `DynamicContentHandler` Interface

The following simple example shows how to display the dynamic task items in the task menu. The example uses the `org.apache.myfaces.trinidad.model.TreeModel` implementation called `org.apache.myfaces.trinidad.model.RowKeyPropertyTreeModel` to pull and display data in the task menu hierarchy. The class `org.apache.myfaces.trinidad.model.RowKeyPropertyTreeModel` is a subclass of `org.apache.myfaces.trinidad.model.ChildPropertyTreeModel`. The class `ChildPropertyTreeModel` creates a `TreeModel` from a list of beans and the class `RowKeyPropertyTreeModel` adds the support of row keys to the `TreeModel`. Let's assume that we have a `TreeModel` object created for a hierarchy of `Employee` objects. We want to display the names of the employees as tasks in the task list. The managers will be displayed as folders.

The action on the manager's name will open the list of directs for the manager. The action on the leaf employee names will open the employee details screen for the selected employee.

The implementation class for the `DynamicContentHandler` interface is named as `CustomTreeModelDynamicContentHandler`. It has a method that builds the `TreeModel` from the list of `Employee` objects.

```
/**
 * Build a custom tree model.
 * @return
 */
private TreeModel buildTreeModel() {
    List<Employee> allEmployees = new ArrayList<Employee>();

    Employee manager1 = new Employee(197, "Kevin Feeny");
    Employee emp = new Employee(198, "Donald OConnel");
    manager1.addDirect(emp);
    emp = new Employee(199, "Douglas Grant");
    manager1.addDirect(emp);

    Employee manager2 = new Employee(200, "Jennifer Whalen");
    emp = new Employee(201, "Michael Harstein");
    manager2.addDirect(emp);
}
```

```

Employee manager3 = new Employee(203, "Susan Marvis");
emp = new Employee(202, "Pat Fay");
manager3.addDirect(emp);
manager2.addDirect(manager3);

emp = new Employee(204, "Hermann Baer");

allEmployees.add(manager1);
allEmployees.add(manager2);
allEmployees.add(emp);

//create a RowKeyPropertyTreeModel with employeeId as a row key and
directs as child property
return new RowKeyPropertyTreeModel(allEmployees, "directs", "employeeId");
}

```

The TreeModel is initialized on the initialization of the class CustomTreeModelDynamicContentHandler.

```

public class CustomTreeModelDynamicContentHandler implements
DynamicContentHandler {

private static final ADFLogger LOG =
ADFLogger.createADFLogger(CustomTreeModelDynamicContentHandler.class);
private TreeModel treeModel;
private static final String TASKMENU_ITEM_ID_SEPARATOR = "_";

public CustomTreeModelDynamicContentHandler() {
super();
this.treeModel = buildTreeModel();
}

Implement the getChildren method of the DynamicContentHandler interface.
/**
 * This method is called by the Retail Application to get the dynamic task
items.
 * @param taskMenuItem
 * @return
 */
public List<TaskMenuItem> getChildren(TaskMenuItem taskMenuItem) {
//if taskMenuItem is null then framework is trying to load the root
objects,
//otherwise the framework has already navigated the tree
if (taskMenuItem != null) {
this.treeModel.setRowKey(taskMenuItem.getPath());
this.treeModel.enterContainer();
this.treeModel.setRowIndex(0);
} else {
this.treeModel.setRowKey(null); //move to root
}
return getChildrenForCurrentRowKey();
}

/**
 * This method returns the children for the current row key in the tree model.
 * @param model
 * @return
 */
private List<TaskMenuItem> getChildrenForCurrentRowKey() {
List<TaskMenuItem> childTaskMenuItemList = new ArrayList<TaskMenuItem>();
int rowCount = this.treeModel.getRowCount();
for (int i = 0; i < rowCount; i++) {
this.treeModel.setRowIndex(i);
}
}
}

```

```

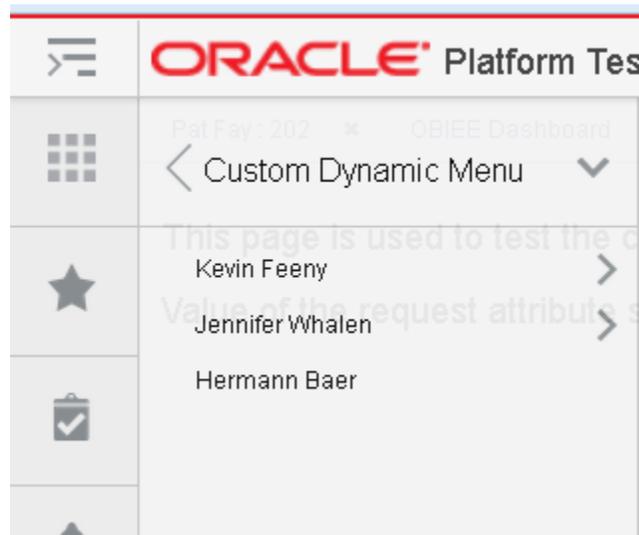
        TaskMenuItem childTaskMenuItem = buildTaskMenuItem();
        if (childTaskMenuItem != null)
            childTaskMenuItemList.add(childTaskMenuItem);
    }
    return childTaskMenuItemList;
}

```

The Retail Application will instantiate the class `oracle.retail.apps.framework.uishell.taskmenu.handler.custom.CustomTreeModelDynamicContentHandler`. It will cache the class for subsequent requests after instantiating it once.

On the initial load of the page, the Retail Application will call `getChildren` on the `dynamicContentHandler` implementation class by passing it a null value of the `TaskMenuItem` object.

When the argument is null, the method `getChildren` should return the root dynamic task items. The Retail Application will display the root items on the initial load of the page. In the example above, the employees Kevin, Jennifer and Hermann are the root items of the `TreeModel` so they will be loaded by the Retail Application and displayed as the initial dynamic task items for the `CustomTreeModelDynamicContentHandler`.



When the user clicks on a Folder, the Retail Application will send the current selected `TaskMenuItem` object to the `getChildren` method to load the children of the current selected task item. When the argument of the `getChildren` method is a valid `TaskMenuItem` object, the `getChildren` method should return the children of the `TaskMenuItem` object.

The Retail Application always lazy loads the dynamic items. This means that the children of a dynamic Folder will only be loaded when the user clicks on the Folder. Once the children are loaded they are cached by the Retail Application. In the above example, even though the Retail Application lazy loads the dynamic items, the `TreeModel` pre-loads all the employees.

TaskMenuItem Class

```

private Object path;
private String id;
private String title;
private String url;
private String shortDesc;

```

```
private boolean visible = true;
private TaskMenuItemType type = TaskMenuItemType.LINK;
private TaskMenuTarget target;
private List<Parameter> parameters;
private List<Attribute> attributes;
private String accessKey;
private boolean disabled;
```

The following is a java doc screenshot of the TaskMenuItem class

oracle.retail.apps.framework.uishell.taskmenu.dynamiccontent.dataobject

Class TaskMenuItem

java.lang.Object

oracle.retail.apps.framework.uishell.taskmenu.dynamiccontent.dataobject.TaskMenuItem

```
public class TaskMenuItem
extends java.lang.Object
```

Nested Class Summary

Nested Classes

Modifier and Type	Class and Description
static class	TaskMenuItem.TaskMenuItemType
static class	TaskMenuItem.TaskMenuTarget

Constructor Summary

Constructors

Constructor and Description
TaskMenuItem()

All Methods	Instance Methods	Concrete Methods
Modifier and Type		Method and Description
java.lang.String		getAccessKey()
java.util.List<Attribute>		getAttributes()
java.lang.String		getId()
java.util.List<Parameter>		getParameters()
java.lang.Object		getPath()
java.lang.String		getShortDesc()
TaskMenuItem.TaskMenuTarget		getTarget()
java.lang.String		getTitle()
TaskMenuItem.TaskMenuItemType		getType()
java.lang.String		getUri()
boolean		isDisabled()
boolean		isVisible()
java.lang.String		retrieveAttribute(java.lang.String attributeName) return attribute value for a given attribute.
java.lang.String		retrieveParameter(java.lang.String parameterName) return parameter value for a given parameter.
void		setAccessKey(java.lang.String accessKey)
void		setAttribute(java.lang.String attributeName, java.lang.String attributeValue) This method sets a single attribute for task menu item.
void		setAttributes(java.util.List<Attribute> attributes)
void		setDisabled(boolean disabled)
void		setId(java.lang.String id)
void		setParameter(java.lang.String parameterName, java.lang.String parameterValue) This method sets a single parameter for task menu item.
void		setParameters(java.util.List<Parameter> parameters)
void		setPath(java.lang.Object path)
void		setShortDesc(java.lang.String shortDesc)
void		setTarget(TaskMenuItem.TaskMenuTarget target)
void		setTitle(java.lang.String title)
void		setType(TaskMenuItem.TaskMenuItemType type)

The attribute 'path' can be used to store the information of the current node. The 'path' attribute is of type 'Object' so it can store any object which can be used to identify the current node in the external hierarchy. In the example above, the path stores the row key of the TreeModel which is an ArrayList of employeeId's starting from the root to the current selected node. The path can vary based on the implementation of the DynamicContentHandler interface. For example, if the DynamicContentHandler implementation reads the database tables to display a dynamic menu then the path can be an ArrayList of the primary keys of the database tables, if the DynamicContentHandler implementation calls an external web service to generate a dynamic menu then the path can be a String object which identifies a node in the external system.

The following code snippet shows how a TaskMenuItem can be built in the DynamicContentHandler implementation class.

```
/**
 * build a task menu item from the custom object.
 * @param rowKey
 * @param employee
 * @return
 */
private TaskMenuItem buildTaskMenuItem() {

    List<Object> rowKey = (List<Object>)this.treeModel.getRowKey();
    Employee employee = (Employee)this.treeModel.getRowData();

    if (this.treeModel.isContainer()) {
        return buildContainerTaskMenuItem(rowKey, employee);
    } else {
```

```

        return buildLeafTaskMenuItem(rowKey, employee);
    }
}

/**build container taskmenu item.
 * @param rowKey
 * @param employee
 * @return
 */
private TaskMenuItem buildContainerTaskMenuItem(List<Object> rowKey, Employee
employee) {
    TaskMenuItem taskMenuItem = new TaskMenuItem();
    taskMenuItem.setPath(rowKey);
    taskMenuItem.setId(generateUniqueId(rowKey));
    taskMenuItem.setTitle(employee.getName());
    taskMenuItem.setShortDesc(employee.getName() + " : " +
employee.getEmployeeId());
    taskMenuItem.setType(TaskMenuItem.TaskMenuItemType.FOLDER);
    return taskMenuItem;
}

/**build leaf task menu item.
 * @param rowKey
 * @param employee
 * @return
 */
private TaskMenuItem buildLeafTaskMenuItem(List<Object> rowKey, Employee
employee) {
    TaskMenuItem taskMenuItem = new TaskMenuItem();
    taskMenuItem.setPath(rowKey);
    taskMenuItem.setId(generateUniqueId(rowKey));
    taskMenuItem.setTitle(employee.getName());
    taskMenuItem.setShortDesc(employee.getName() + " : " +
employee.getEmployeeId());
    taskMenuItem.setType(TaskMenuItem.TaskMenuItemType.TASKFLOW);
    taskMenuItem.setUrl("/WEB-INF/oracle/retail/apps/framework/uishell/navigation/contentarea/flow/TestOverrideC
loseTabFlow.xml#TestOverrideCloseTabFlow");

    //set attributes on taskMenuItem
    taskMenuItem.setAttribute("TabTitle",
employee.getName() + " : " +
employee.getEmployeeId());
    taskMenuItem.setAttribute("tabShortDesc",
"Employee " + employee.getName() + " has employee
Id " +
employee.getEmployeeId() + ".");

    //set parameters on taskMenuItem
    taskMenuItem.setParameter("EmployeeId",
String.valueOf(employee.getEmployeeId()));
    return taskMenuItem;
}

/**This method generages a unique id for the taskmenu item.
 * @param rowKey
 * @return
 */
private String generateUniqueId(List<Object> rowKey) {
    return rowKey.stream().map(c ->
c.toString()).collect(Collectors.joining(TASKMENU_ITEM_ID_SEPARATOR));
}

```

Default Dynamic Task Items

When a user logs in the Retail Application, the user sees the default content that has been configured in the Retail Application. The default content opens up directly in a Retail UIShell tab without the user having to launch it from the task menu. The dynamic task items can also be configured as the default content in the application. The DynamicContentHandler interface provides a method called getDefaultTaskMenuItems which returns a List of TaskMenuItem objects. The items returned from this method will be launched automatically after the user logs in.

The following is a sample implementation of the getDefaultTaskMenuItems method:

```

    /**Get the default task menu items. These will be automatically loaded on
    initial load of the home page.
     * @return
     */
    @Override
    public List<TaskMenuItem> getDefaultTaskMenuItems() {
        List<Object> rowKey = new ArrayList<Object>();
        rowKey.add(200); //"Jennifer Whalen"
        rowKey.add(203); //"Susan Marvis"
        rowKey.add(202); //"Pat Fay"
        Employee employee = (Employee)this.treeModel.getRowData(rowKey); //get Pat
        Fay

        TaskMenuItem taskMenuItem = buildLeafTaskMenuItem(rowKey, employee);
        List<TaskMenuItem> defaultTaskMenuItems = new ArrayList<TaskMenuItem>();
        defaultTaskMenuItems.add(taskMenuItem);
        return defaultTaskMenuItems;
    }

```

In-Context Launch of Dynamic Task Items

The Retail Application provides a way to launch links and taskflows directly using a browser URL. The content is launched in the context of the external application by passing parameters in the URL.

The Retail Application requires a navModelItemId parameter in the request to identify the content that needs to be launched. The Retail Application checks the static task menu model files to see if an Item exists with that id in the model file. If the Retail Application does not find a static item, it consults the list of dynamic content handlers configured with the application to look for the item. If the dynamic content handler returns a TaskMenuItem for the given id, it will be launched in the Retail Application. The dynamic content handler should implement a method called getInContextTaskMenuItem(String taskMenuItemId) which will return the TaskMenuItem object for the taskMenuItemId received. The getInContextTaskMenuItem should be able to find the Item using the taskMenuItemId.

The following is a sample implementation of the getInContextTaskMenuItem(String taskMenuItemId) method.

```

    /**This method returns an in-context item.
     * @param taskMenuItemId
     * @return
     */
    @Override
    public TaskMenuItem getInContextTaskMenuItem(String taskMenuItemId) {
        TaskMenuItem inContextTaskMenuItem = null;
        Employee employee = null;
        List<Object> rowKey = null;
        if (taskMenuItemId != null) {

```

```

        try {
            rowKey =

Arrays.stream(taskMenuItemId.split(TASKMENU_ITEM_ID_SEPARATOR)).map(c ->
Integer.valueOf(c)).collect(Collectors.toList());
            employee = (Employee) this.treeModel.getRowData(rowKey);

            if ((employee != null) && (employee.getDirects().isEmpty())) {
                inContextTaskMenuItem = buildLeafTaskMenuItem(rowKey,
employee);
            }
        } catch (Exception e) {
            if (LOG.isWarning()) {
                LOG.warning("Error occurred looking for dynamic In-Context
Item with id " +
                                taskMenuItemId, e);
            }
        }
    }
    return inContextTaskMenuItem;
}

```

The custom dynamic content handler can be added to the classpath of the Retail Application by deploying it as a shared library. Refer to the section Adding a Custom Shared Library for details on how to deploy a shared library.

Report Adapters

The DynamicContentHandler interface can be used to add dynamic task items in the task menu from a variety of data sources including OBIEE, BIPublisher, database tables, and custom Web Services and so on. The Retail Application provides the default implementation of the DynamicContentHandler interface for OBIEE and BIPublisher. These implementations are called Report Adapters. The Report Adapters can display the user's reports from OBIEE and BIPublisher as tasks in the task list. The user can click on the tasks to display individual reports and dashboards.

Note: The connection to both the OBIEE and BIPublisher should be secured using TLS protocol. Ensure that the Report Adapter integration is done using secured protocol in the production environments.

OBIEE Report Adapter

The RetailAppsObieeAdapter relies on a properties file called RetailAppsObieeAdapter.properties for the OBIEE configuration. Create RetailAppsObieeAdapter.properties in any location on the file system. Modify the setDomainEnv.cmd for the Windows environment or setDomainEnv.sh file for the Unix environment to add the RetailAppsObieeAdapter.properties as a Java system property specifying the location in the file system where the RetailAppsObieeAdapter.properties file resides.

For example for windows environment

```

set EXTRA_JAVA_PROPERTIES=-
DRetailAppsObieeAdapter.properties=D:\Work\RetailAppsFramework\RetailAppsObieeAdap
ter.properties %EXTRA_JAVA_PROPERTIES%

```

For Unix environment

```

EXTRA_JAVA_PROPERTIES="-DRetailAppsObieeAdapter.properties
=/scratch/u00/product/Oracle/Middleware/user_projects/domains/RAFDomain/custom_pro
perties/RetailAppsObieeAdapter.properties ${EXTRA_JAVA_PROPERTIES}"
export EXTRA_JAVA_PROPERTIES

```

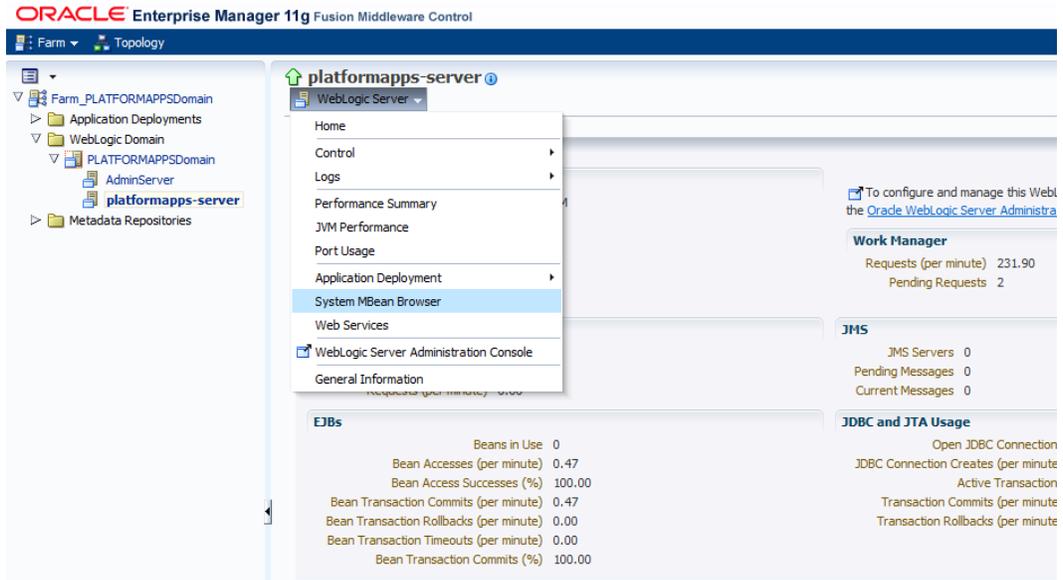
The RetailAppsObieeAdapter.properties can be used to specify the following properties for the OBIEE Report Adapter configuration.

Property	Required	Description
bi.connection.name	Yes	This property is used to specify the name of the BISOapConnection used to display the OBIEE reports. The connection should be created using the Enterprise Manager. Refer to the section Creating the BIconnection for more details.
bi.bipublisher.context.path	No	If the OBIEE installation includes both the OBIEE analytics and BIPublisher then the BIPublisher reports can also be displayed using the RetailAppsObieeAdapter. The bi.bipublisher.context.path is used to specify the context root of the BIPublisher installation. The default value is xmlpserver.
bi.external.integration	No	This property can be used to disable the RetailAppsObieeAdapter integration. Setting this to false will disable the RetailAppsObieeAdapter integration. The default value is true.
bi.report.new.browser.tab	No	This property can be used to display the OBIEE reports in a new browser tab on in a new Retail UI Shell tab. The default behavior of the RetailAppsObieeAdapter is to open the OBIEE reports in a new browser tab. Setting this property to false opens the reports in a new UI Shell tab. The default value is true.

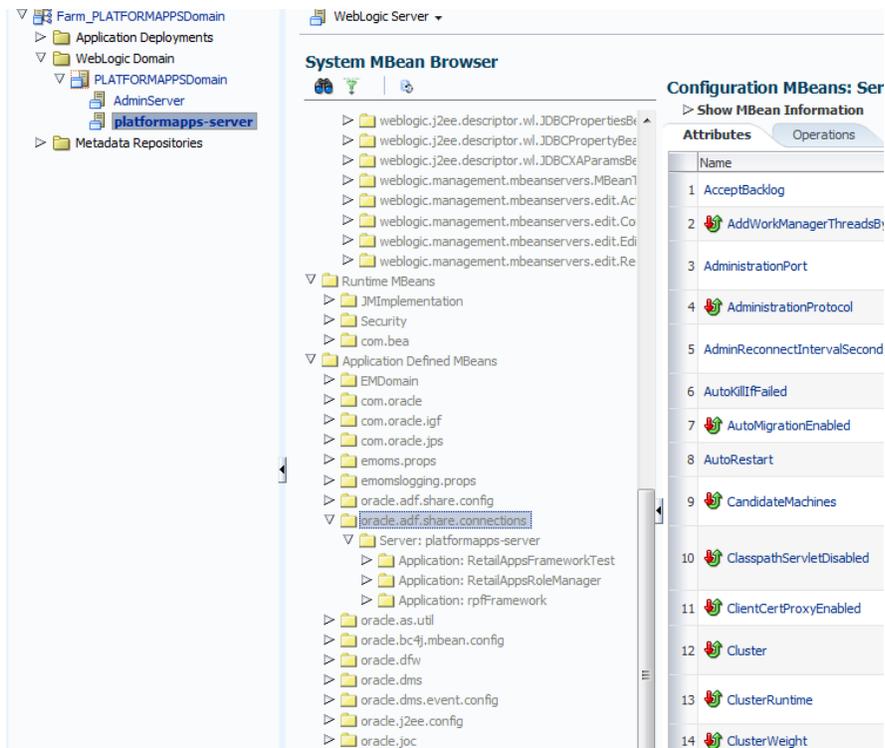
Creating the BICconnection

Follow these steps to create the BICconnection for the RetailAppsObieeAdapter configuration.

1. Log in to the Enterprise Manager of the environment where the Retail Application is deployed and go to 'System MBean Browser' as shown below.



2. Go to 'Application Defined MBeans' and its sub folder 'oracle.adf.share.connections'



3. Expand 'Application: <AppName>' and select 'ADFConnections'. The following example uses RetailAppsFrameworkTest as the AppName.

ORACLE Enterprise Manager 11g Fusion Middleware Control

Farm Topology

Farm_PLATFORMAPPSDomain

- Application Deployments
- WebLogic Domain
 - PLATFORMAPPSDomain
 - AdminServer
 - platformapps-server**
 - Metadata Repositories

platformapps-server WebLogic Server

System MBean Browser

- Configuration MBeans
 - JMImplementation
 - Security
 - com.bea
- Runtime MBeans
 - JMImplementation
 - Security
 - com.bea
- Application Defined MBeans
 - EMDomain
 - com.orade
 - com.orade.igf
 - com.orade.jps
 - emoms.props
 - emomslogging.props
 - orade.adf.share.config
 - orade.adf.share.connections
 - Server: platformapps-server
 - Application: RetailAppsFrameworkTest
 - ADFConnections
 - Application: RetailAppsRoleManager
 - Application: rpfframework
 - orade.as.util
 - orade.bc4j.mbean.config
 - orade.dfw
 - orade.dms
 - orade.dms.event.config
 - orade.dms.config

Applicati

Show

Attribut	
	Name
1	Confi
2	Conn
3	Conn
4	even
5	even
6	objec
7	Read
8	Reste
9	state
10	statis
11	Syste

- Go to the 'Operations' tab and click 'createConnection'.

The screenshot shows the 'System MBean Browser' on the left and the 'Application Defined MBeans: ADFConnections:ADFCConnections' on the right. The 'Operations' tab is active, displaying a table of operations. The 'createConnection' operation is highlighted with a red box.

Name	Description	Parameters	Return Type
1 createConnection	Create a connection MBean for the connection name of the connection type	2 javax.management.ObjectName	
2 export	Export the last saved document as XML.	0 java.lang.String	
3 exportToDisk	Export the last saved document as an XML file to disk on the server.	1 void	
4 exportToDiskWithResolvedExpressions	Export the last saved document with expressions resolved as an XML file to disk on the server.	1 void	
5 exportWithResolvedExpressions	Export the last saved document with expressions resolved as XML.	0 java.lang.String	
6 findOrCreateConnection	Create a connection MBean for the connection name of the connection type	2 javax.management.ObjectName	
7 getADFConnectionGenericAttribute	Get value of Connection Reference element attribute.	2 java.lang.String	
8 getConnectionNames	List all connections (names) for a given connectionType.	1 Array of java.lang.String	
9 isEditable	True if the connections document can be edited using mbeans.	0 java.lang.Boolean	
10 listConnectionMBeans	Lookup the connection by connection type.	0 Array of javax.management.ObjectName	
11 listConnectionMBeans	Lookup the connection by connection type.	1 Array of javax.management.ObjectName	
12 listManageInOracleEnterpriseManagerConnections	List the connections(objectnames) that are configurable in EM UI.	1 Array of javax.management.ObjectName	
13 listManageInOracleEnterpriseManagerConnections	List the connections(objectnames) that are configurable in EM UI.	0 Array of javax.management.ObjectName	
14 refresh	Refresh connection mbeans.	0 void	
15 removeConnection	Remove the connection object identified by this name.	1 void	

- Enter Connection Type as BISOapConnection and Connection Name as BICConnection. Click Invoke on top right.

The screenshot shows the 'System MBean Browser' on the left and the 'Operation: createConnection' details on the right. The 'Parameters' section is expanded, showing the 'Connection Type' and 'Connection Name' fields.

MBean Name: oracle.adf.share.connections:ApplicationName=RetailAppFrameworkTest,Location=platformapps-server,name=ADFCConnections,beanType=Runtime,type=ADFCConnections,Application=RetailAppFrameworkTest

Operation Name: createConnection

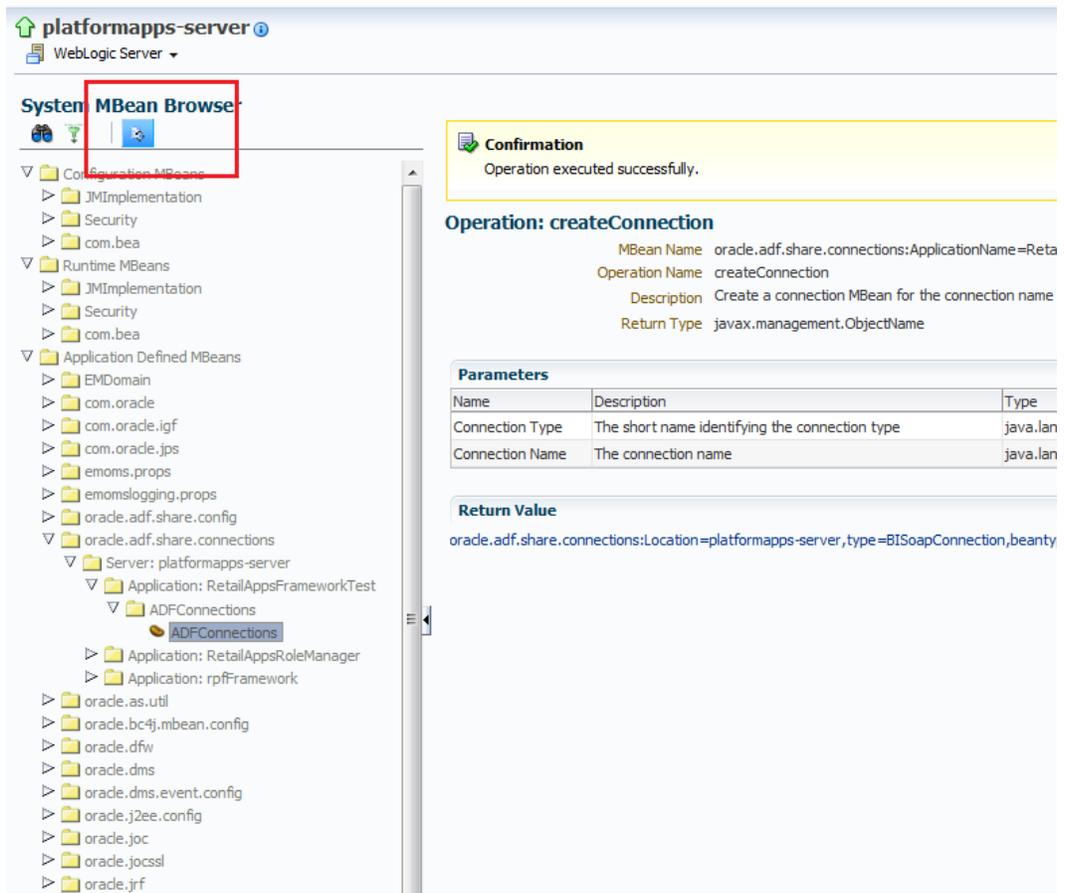
Description: Create a connection MBean for the connection name of the connection type

Return Type: javax.management.ObjectName

Name	Description	Type	Value
Connection Type	The short name identifying the connection type	java.lang.String	BISOapConnection
Connection Name	The connection name	java.lang.String	BICConnection

Return Value:

6. Refresh the tree using the button 'Refresh cached tree data'.



Confirmation
Operation executed successfully.

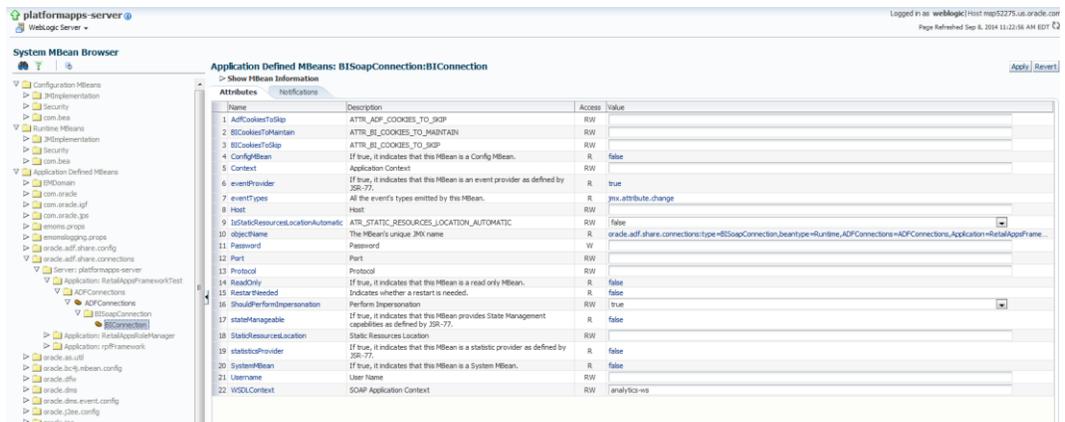
Operation: createConnection

MBean Name oracle.adf.share.connections:ApplicationName=RetailAppsFrameworkTest
 Operation Name createConnection
 Description Create a connection MBean for the connection name
 Return Type javax.management.ObjectName

Parameters		
Name	Description	Type
Connection Type	The short name identifying the connection type	java.lang
Connection Name	The connection name	java.lang

Return Value
oracle.adf.share.connections:Location=platformapps-server,type=BISoapConnection,beauty

7. Once the tree is refreshed, you will notice that 'ADFConnections' becomes a folder and it has a new connection 'BIConnection'. Click on the BIConnection to open its properties in the right pane.



Application Defined MBeans: BISoapConnection:BIConnection

Show MBean Information

Name	Description	Access	Value
1 ADFCookiesToSkip	ATTR_ADF_COOKIES_TO_SKIP	RW	
2 BICookiesToMaintain	ATTR_BI_COOKIES_TO_MAINTAIN	RW	
3 BICookiesToSkip	ATTR_BI_COOKIES_TO_SKIP	RW	
4 ConfigMBean	If true, it indicates that this MBean is a Config MBean.	R	false
5 Context	Application Context	RW	
6 eventProvider	If true, it indicates that this MBean is an event provider as defined by JSR-77.	R	true
7 eventTypes	All the event's types emitted by this MBean.	R	javax.attribute.change
8 Host	Host	RW	
9 J2SEStaticResourceLocationAutomatic	ATTR_STATIC_RESOURCES_LOCATION_AUTOMATIC	RW	false
10 objName	The MBean's unique JMX name	R	oracle.adf.share.connections:type=BISoapConnection,beauty=Runtime_ADFConnections=ADFConnections,Application=RetailAppsFrameworkTest
11 Password	Password	W	
12 Port	Port	RW	
13 Protocol	Protocol	RW	
14 ReadOnly	If true, it indicates that this MBean is a read-only MBean.	R	false
15 RestartNeeded	Indicates whether a restart is needed.	R	false
16 ShouldPerformImpersonation	Perform Impersonation	RW	true
17 stateManageable	If true, it indicates that the MBean provides State Management capabilities as defined by JSR-77.	R	false
18 StaticResourceLocation	Static Resources Location	RW	
19 staticProvider	If true, it indicates that this MBean is a static provider as defined by JSR-77.	R	false
20 SystemMBean	If true, it indicates that this MBean is a System MBean.	R	false
21 Username	User Name	RW	
22 WSDLContext	SOAP Application Context	RW	analytics-ns

8. Enter the values shown in the table below. Once done click Apply on the top right.

Context	The context root of the Oracle BI EE installation. Example: analytics
Host	The host of the Oracle BI EE installation.
IsStaticResourcesLocationAutomatic	The IsStaticResourcesLocationAutomatic attribute can be either true or false. When this attribute is true, the BI Presentation Server will be used to specify the location of the static resources. When this attribute is false, the attribute 'StaticResourcesLocation' should be provided. The attribute StaticResourcesLocation will be used to find the static resources when the attribute IsStaticResourcesLocationAutomatic is false.
Password	The password for the BIImpersonateUser.
Port	The port of the Oracle BI managed server. Example 9704
Protocol	http or https For production environments, use https.
ShouldPerformImpersonation	true (keep it true, default is true)
StaticResourcesLocation	The path of the location where the static resources are available. If the IsStaticResourcesLocationAutomatic attribute is true then use the following URL for the StaticResourcesLocation attribute. Error! Hyperlink reference not valid. If the IsStaticResourcesLocationAutomatic attribute is false, then use the following URL for the StaticResourcesLocation attribute. Error! Hyperlink reference not valid.> The Oracle BI EE static resources, namely the 'res' and 'analyticsRes' directories should be available from the Oracle Web Tier's htdocs directory when the IsStaticResourcesLocationAutomatic attribute is false.
Username	The username that has been configured on the Oracle BI EE with the impersonate permissions. Example: BIImpersonateUser
WSDLContext	analytics-ws

9. Go to 'ADFConections' to save the changes. Click on 'save' operation in the Operations tab.

Name	Description	Parameters	Return Type
1 createConnection	Create a connection MBean for the connection name of the connection type	2 javax.management.ObjectName	
2 export	Export the last saved document as XML.	0 java.lang.String	
3 exportToDisk	Export the last saved document as an XML file to disk on the server.	1 void	
4 exportToDiskWithResolvedExpressions	Export the last saved document with expressions resolved as an XML file to disk on the server.	1 void	
5 exportWithResolvedExpressions	Export the last saved document with expressions resolved as XML.	0 java.lang.String	
6 findOrCreateConnection	Create a connection MBean for the connection name of the connection type	2 javax.management.ObjectName	
7 getADFConnectionGenericAttribute	Get value of Connection Reference element attribute.	2 java.lang.String	
8 getConnectionNames	List all connections (names) for a given connection type.	1 Array of java.lang.String	
9 isEditable	True if the connections document can be edited using mbeans.	0 java.lang.Boolean	
10 listConnectionMBeans	Lookup the connection by connection type.	0 Array of javax.management.ObjectName	
11 listConnectionNames	Lookup the connection by connection type.	1 Array of javax.management.ObjectName	
12 listManageInOracleEnterpriseManagerConnections	List the connections(objectnames) that are configurable in EM UI.	1 Array of javax.management.ObjectName	
13 listManageInOracleEnterpriseManagerConnections	List the connections(objectnames) that are configurable in EM UI.	0 Array of javax.management.ObjectName	
14 refresh	Refresh connection mbeans.	0 void	
15 removeConnection	Remove the connection object identified by this name.	1 void	
16 restoreDomainConfigFromBase	For any ADF Connection property that support to ADFDomainConfig, writes any customizations of the property with the value in the base connections.xml document.	0 void	
17 restoreToOriginalConfiguration	Save connections.	0 void	
18 save	Save connections.	0 void	
19 updateADFConnectionGenericAttribute	Update generic Connection Reference element attributes.	3 void	
20 validate	Validate Connections.	0 void	

10. Click on 'Invoke' on the top right to save the changes.

Once the connection is created successfully, modify the RetailAppsObieeAdapter.properties to add the bi.connection.name as BICConnection.

Configuring the Reports Menu

Once we have configured the RetailAppsObieeAdapter.properties and created a BICConnection, we can add an Item of type="dynamicContent" in the HomeReportsMenuModel.xml file to display the OBIEE Reports in the Reports menu. The implementation of the DynamicContentHandler interface for OBIEE is called oracle.retail.apps.framework.report.obiee.handler.ObieeDynamicContentHandler.

```
<Item id="obieeFolder" visible="{true}" title="OBIEE" type="folder">
  <Items>
    <Item id="obieeDynamicContent" title="OBIEE Dynamic Menu"
type="dynamicContent"
dynamicContentHandler="oracle.retail.apps.framework.report.obiee.handler.ObieeDyna
micContentHandler"
visible="{securityContext.authenticated}"/>
  </Items>
</Item>
```

The reports menu model can be customized to add an entry for the dynamic content as shown above. Refer to the section Adding or Modifying an Item in the Reports Menu for more details.

BIPublisher Report Adapter

The RetailAppsBiPublisherAdapter relies on a properties file called RetailAppsBiPublisherAdapter.properties for the BIPublisher configuration. Create RetailAppsBiPublisherAdapter.properties in any location on the file system. Modify the setDomainEnv.cmd for the Windows environment or setDomainEnv.sh file for the Unix environment to add the RetailAppsBiPublisherAdapter.properties as a Java system property specifying the location in the file system where the RetailAppsBiPublisherAdapter.properties file resides.

For example for windows environment

```
set EXTRA_JAVA_PROPERTIES=-
DRetailAppsBiPublisherAdapter.properties=D:\Work\RetailAppsFramework\RetailAppsBiPublisherAdapter.properties %EXTRA_JAVA_PROPERTIES%
```

For Unix environment

```
EXTRA_JAVA_PROPERTIES="-
DRetailAppsBiPublisherAdapter.properties=/scratch/u00/product/Oracle/Middleware/user_projects/domains/RAFDomain/custom_properties/RetailAppsBiPublisherAdapter.properties ${EXTRA_JAVA_PROPERTIES}"
export EXTRA_JAVA_PROPERTIES
```

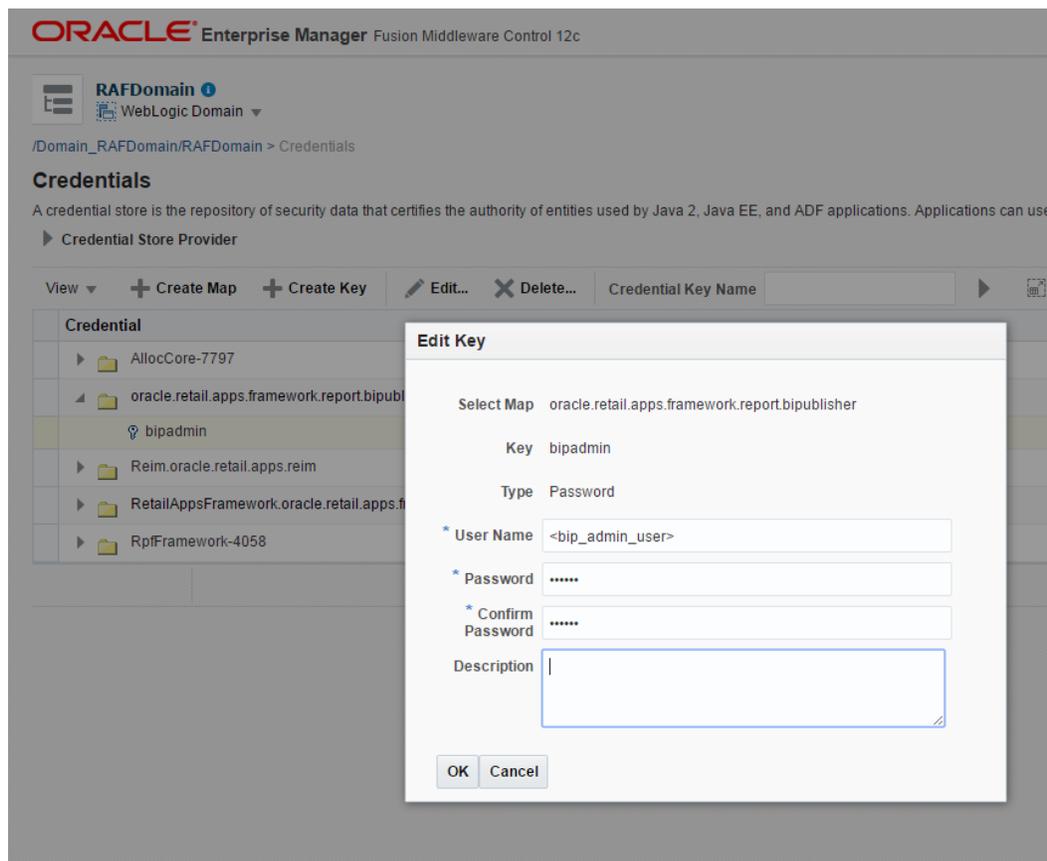
The RetailAppsBiPublisherAdapter.properties can be used to specify the following properties for the BIPublisher Report Adapter configuration.

Property	Required	Description
bipublisher.wsdl.location	Yes	This property is used to specify the WSDL URL of the BIPublisher PublicReportService. For example: https://bipublisher_host:bipublisher_port/xmlpserver/services/PublicReportService?wsdl
bipublisher.external.integration	No	This property can be used to disable the RetailAppsBiPublisherAdapter integration. Setting this to false will disable the RetailAppsBiPublisherAdapter integration. The default value is true.
bipublisher.report.new.browser.tab	No	This property can be used to display the BIPublisher reports in a new browser tab on in a new Retail UI Shell tab. The default behavior of the RetailAppsBiPublisherAdapter is to open the BIPublisher reports in a new browser tab. Setting this property to false opens the reports in a new UI Shell tab. The default value is true.

Store the BIPublisher Admin Credentials

The RetailAppsBiPublisherAdapter requires the admin credentials of the BIPublisher to be stored in the WebLogic domain credential store where the Retail Application is deployed. The admin credentials are used by the RetailAppsBiPublisherAdapter to connect to the BIPublisher.

Inside this map create a new key called 'bipadmin' and enter the BIPublisher admin credentials.



Configuring the Reports Menu

Once we have configured the RetailAppsBiPublisherAdapter.properties and have added the BIPublisher WSDL URL property in the properties file, we can add an Item of type="dynamicContent" in the HomeReportsMenuModel.xml file to display the BIPublisher Reports in the Reports menu. The implementation of the DynamicContentHandler interface for the BIPublisher is called oracle.retail.apps.framework.report.bipublisher.handler.BiPublisherDynamicContentHandler.

```
<Item id="bipublisherFolder" visible="#{true}" title="BiPublisher"
type="folder">
  <Items>
    <Item id="bipublisherDynamicContent" title="BIPublisher Dynamic
Menu"
      type="dynamicContent"
      dynamicContentHandler="oracle.retail.apps.framework.report.bipublisher.handler.BiP
ublisherDynamicContentHandler"
      visible="#{securityContext.authenticated}"/>
    </Items>
  </Item>
```

The reports menu model can be customized to add an entry for the dynamic content as shown above. Refer to the section Adding or Modifying an Item in the Reports Menu for more details.

In-Context Launchable Task Flows

Retail applications can expose select task flows that retailers can directly launch into. This feature is referred to as in-context launching in a retail application. Retailers can launch these task flows directly through specific URLs.

Retail Applications will provide information about the various task flows that retailers can in-context launch into, including the URLs and the required parameters.

Retailers can use these URLs in other web pages as links. For example, the URL to a task flow that invokes the Create Transfer Flow in a Retail Application can be added as a link to dashboard reports on a BI server

When the user clicks on a URL to a task flow, the Retail Application will open in a new browser window or tab, depending on the specified target of the URL. The requested task flow will be shown as a UI tab within the Retail Application.

Limitations of In-Context Launch via URLs

The In-Context launch feature will not detect if there is an already opened window or tab for the Retail Application. So when a user clicks on a link to a Retail Application's task flow, for example, a report on a BI server, a new browser window or tab will be opened even though the user already has an existing browser window or tab opened for that same Retail Application.

If a BI dashboard is added on the Retail Application, and if that dashboard has a report that contains links to in-context launchable task flows in the same Retail Application, a new browser window or tab will still be opened.

List of In-Context Launchable Task Flows

The following table lists the in-context launchable task flows.

Source	Parameters in Taskflow	Task Flow Name
Item Header	pmItem pmMode	WEB-INF/oracle/retail/apps/rms/foundation/item/view/itemsetup/itemmaster/flow/MaintainItemFlow.xml
Shipment	pmShipment pmMode	WEB-INF/oracle/retail/apps/rms/inventory/view/shipment/detail/flow/MaintainShipmentDetailFlow.xml
Transfer	pmTsfNo pmMode	WEB-INF/oracle/retail/apps/rms/inventory/view/transfer/setup/header/flow/MaintainTransferHeaderFlow.xml#MaintainTransferHeaderFlow
RTV Maintenance	pmRtvOrderNo pmMode	/WEB-INF/oracle/retail/apps/rms/inventory/view/rtv/detail/flow/MaintainRtvDetailFlow.xml#MaintainRtvDetailFlow

Source	Paramers in Taskfow	Task Flow Name
Transaction Data	pmItem pmLoc pmLocType pmWh pmStore pmTranStartDate pmTranEndDate pmMode pmPostStartDate pmPostEndDate pmDept pmClass pmSubclass pmCreatedDate	/WEB-INF/oracle/retail/apps/rms/inventory/view/stockledger/detail/flow/ViewTransactionDataFlow.xml#ViewTransactionDataFlow
Average Cost Adjustment	pmMode pmItem pmLocation pmLocType	/WEB-INF/oracle/retail/apps/rms/inventory/view/adjustment/cost/flow/MaintainAverageCostAdjustmentFlow.xml#MaintainAverageCostAdjustmentFlow

Pro*C Restart and Recovery

RMS has implemented a restart/recovery process in most of its batch architecture. The general purpose of restart/recovery is to:

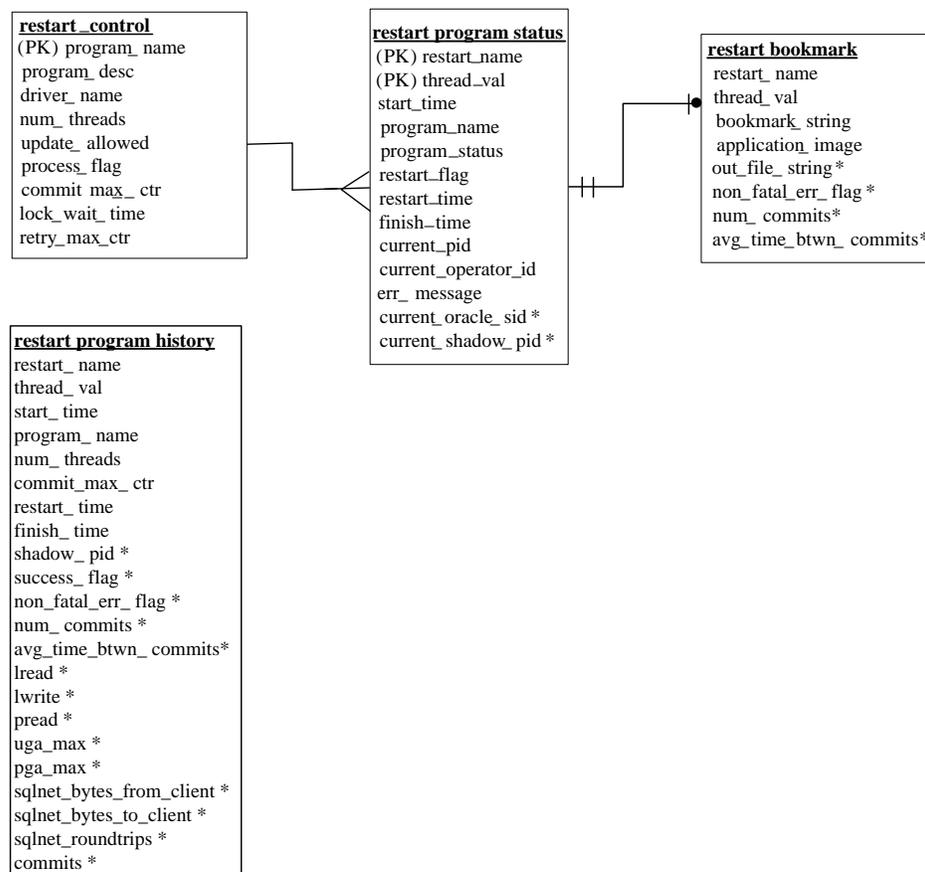
- Recover a halted process from the point of failure.
- Prevent system halts due to large numbers of transactions.
- Allow multiple instances of a given process to be active at the same time.

Further, the RMS restart/recovery tracks batch execution statistics and does not require DBA authority to execute.

The restart capabilities revolve around a program's logical unit of work (LUW). A batch program processes transactions, and commit points are enabled based on the LUW. LUWs consist of a relatively unique transaction key (such as sku/store) and a maximum commit counter. Commit events take place after the number of processed transaction keys meet or exceed the maximum commit counter. For example, after every 10,000 sku/store combinations, a commit occurs. At the time of commit, key data information that is necessary for restart is stored in the restart tables. In the event of a handled or unhandled exception, transactions is rolled back to the last commit point, and upon restart the key information is retrieved from the tables so that processing can continue from the last commit point.

Table of Description and Definition

The RMS restart/recovery process is driven by a set of four tables. These tables are shown in Entity Relationship diagram. For detailed table description, see `RESTART_CONTROL`, `RESTART_PROGRAM_STATUS`, `RESTART_BOOKMARK` and `RESTART_PROGRAM_HISTORY`.



RMS Restart Recovery Entity Relationships Diagram

RESTART_CONTROL

The RESTART_CONTROL table is the master table in the restart/recovery table set. One record exists on this table for each batch program that is run with restart/recovery logic in place. The restart/recovery process uses this table to determine:

- Whether the restart/recovery is table-based or file-based.
- The total number of threads used for each batch program.
- The maximum records processed before a commit event takes place.
- The driver for the threading (multi-processing) logic.
- Time to wait for a lock and the number of lock retry.

RESTART_CONTROL

Column Name	Data Type	Length	Description
(PK) PROGRAM_NAME	varchar2	25	Batch program name.
PROGRAM_DESC	varchar2	120	A brief description of the program function.
DRIVER_NAME	varchar2	25	Driver on query, for example, department (non-updatable).
NUM_THREADS	num	10	Number of threads used for current process.

RESTART_CONTROL

UPDATE_ALLOWED	varchar2	2	Indicates whether user can update thread numbers or if done programmatically.
PROCESS_FLAG	varchar2	1	Indicates whether the process is table-based (T) or file-based (F).
COMMIT_MAX_CTR	num	6	Numeric maximum value for counter before commit occurs.
LOCK_WAIT_TIME	num	6	Contains the number of seconds between the last and the next lock retry.
RETRY_MAX_CTR	num	10	Contains the maximum number of lock retry.

RESTART_PROGRAM_STATUS

The RESTART_PROGRAM_STATUS table is the table that holds record keeping information about current program processes. The number of rows for a program on the status table is equal to its NUM_THREADS value on the RESTART_CONTROL table. The status table is modified during restart/recovery initialization and close logic. For table-based processing, the restart/recovery initialization logic assigns the next available thread to a program based on the program status and restart flag. For file-based processing, the thread value is passed in from the input file name. When a thread is assigned, the PROGRAM_STATUS is updated to prevent the assignment of that thread to another process. Information is logged on the current status of a given thread, as well as record keeping information such as operator and process timing information.

RESTART_PROGRAM_STATUS

(PK)RESTART_NAME	varchar2	50	Program name
(PK)THREAD_VAL	num	10	Thread counter
START_TIME	date		dd-mon-yy hh:mi:ss
PROGRAM_NAME	varchar2	25	Program name
PROGRAM_STATUS	varchar2	25	Started, aborted, aborted in init, aborted in process, aborted in final, completed, ready for start.
RESTART_FLAG	varchar2	1	Automatically set to 'N' after abnormal end, must be manually set to 'Y' for program to restart.
RESTART_TIME	date		dd-mon-yy hh:mi:ss.
FINISH_TIME	date		dd-mon-yy hh:mi:ss.
CURRENT_PID	num	15	Starting program ID.
CURRENT_OPERATOR_ID	varchar2	20	Operator that started the program.
ERR_MESSAGE	varchar2	255	Record that caused program abort and associated error message.
CURRENT_ORACLE_SID	num	15	Oracle SID for the session associated with the current process.

RESTART_PROGRAM_STATUS

CURRENT_SHADOW_PID	num	15	O/S process ID for the shadow process associated with the current process. It is used to locate the session trace file when a process is not finished successfully.
--------------------	-----	----	---

RESTART_PROGRAM_HISTORY

The RESTART_PROGRAM_HISTORY table will contain one record for every successfully completed program thread with restart/recovery logic. Upon the successful completion of a program thread, its record on the RESTART_PROGRAM_STATUS table will be inserted into the history table. Table purging will be at user discretion.

RESTART_PROGRAM_HISTORY

RESTART_NAME	vvarchar2	50	Program name.
THREAD_VAL	Num	10	Thread counter.
START_TIME	Date		dd-mon-yy hh:mi:ss.
PROGRAM_NAME	vvarchar2	25	Program name.
NUM_THREADS	Num	10	Number of threads.
COMMIT_MAX_CTR	num	6	Numeric maximum value for counter before commit occurs.
RESTART_TIME	date		dd-mon-yy hh:mi:ss.
FINISH_TIME	date		dd-mon-yy hh:mi:ss.
SHADOW_PID	num	15	O/S process ID for the shadow process associated with the process. It is used to locate the session trace file.
SUCCESS_FLAG	vvarchar2	1	Indicates whether the process finished successfully (reserved for future use).
NON_FATAL_ERR_FLAG	vvarchar2	1	Indicates whether non-fatal errors have occurred for the process.
NUM_COMMITS	num	12	Total number of commits for the process. The possible last commit when restart/recovery is closed is not counted.
AVG_TIME_BTWN_COMMITS	num	12	Accumulated average time between commits for the process. The possible last commit when restart/recovery is closed is not counted.
LREAD	num		Session logical reads. The sum of "db block gets" plus "consistent gets". This includes logical reads of database blocks from either the buffer cache or process private memory.
LWRITE	num		Session logical writes. The sum of "db block changes" plus "consistent changes".
PREAD	num		Physical reads. Total number of data blocks read from disk.

RESTART_PROGRAM_HISTORY

UGA_MAX	num	Peak UGA (user global area) size for a session.
PGA_MAX	num	Peak PGA (program global area) size for the session.
SQLNET_BYTES_FROM_CLIENT	num	Total number of bytes received from the client over Oracle Net Services.
SQLNET_BYTES_TO_CLIENT	num	Total number of bytes sent to the client from the foreground processes.
SQLNET_ROUNDTRIPS	num	Total number of Oracle Net Services messages sent to and received from the client.
COMMITTS	num	Number of user commits. When a user commits a transaction, the redo generated that reflects the changes made to database blocks must be written to disk. Commits often represent the closest thing to a user transaction rate.

RESTART_BOOKMARK

When a restart/recovery program thread is active, its state is started or aborted, and a record for it exists on the RESTART_BOOKMARK table. Restart/recovery initialization logic inserts the record into the table for a program thread. The restart/recovery commit process updates the record with the following restart information:

- A concatenated string of key values for table processing
- A file pointer value for file processing
- Application context information such as counters and accumulators

The restart/recovery closing process deletes the program thread record if the program finishes successfully. In the event of a restart, the program thread information on this table will allow the process to begin from the last commit point.

RESTART_BOOKMARK

(PK) RESTART_NAME	varchar2	50	Program name.
(PK) THREAD_VAL	num	10	Thread counter.
BOOKMARK_STRING	varchar2	255	Character string of key of last committed record.
APPLICATION_IMAGE	varchar2	1000	Application parameters from the last save point.
OUT_FILE_STRING	varchar2	255	Concatenated file pointers (UNIX sometimes refers to these as stream positions) of all the output files from the last commit point of the current process. It is used to return to the right restart point for all the output files during restart process.
NON_FATAL_ERR_FLAG	varchar2	1	Indicates whether non-fatal errors have occurred for the current process.

RESTART_BOOKMARK

NUM_COMMITS	num	12	Number of commits for the current process. The possible last commit when restart/recovery is closed is not counted.
AVG_TIME_BTWN_COMMITS	num	12	Average time between commits for the current process. The possible last commit when restart/recovery is closed is not counted.

V_RESTART_<X>

Restart views are used for query-based programs that require multi-threading. Separate views (ex v_restart_dept, and v_restart_store) are created for each threading driver, for example, department or store. A join is made to a view based on threading driver to force the separation of discrete data into particular threads. See the threading discussion for more details.

V_RESTART_<X>

DRIVER_NAME	varchar2	Example dept, store, region, and others.
NUM_THREADS	number	Total number of threads in set (defined on restart control).
DRIVER_VALUE	number	Numeric value of the driver_name.
THREAD_VAL	number	Thread value defined for driver_value and num_threads combination.

Restart and Recovery Data Model Design

The RESTART_PROGRAM_STATUS and RESTART_BOOKMARK are separate tables. This is because the initialization process needs to fetch all of the rows associated with RESTART_NAME/schema, but will only update one row. The commit process will continually lock a row with a specific RESTART_NAME and THREAD_VAL. The data involved with these two processes is separated into two tables to reduce the number of hangs that could occur due to locked rows. Even if you allow 'dirty reads' on locked rows, a process will still hang if it attempts to do an update on a locked row. The commit process is only interested in a unique row, so if we move the commit process data to a separate table with row level (not page level) locking, there will not be contention issues during the commit. With the separate tables, the initialization process will now see fewer problems with contention because rows will only be locked twice, at the beginning and end of the process.

Physical Set-Up

The restart/recovery process needs to be as robust as possible in the event of database related failure. The costs outweigh the benefits of placing the restart/recovery tables in a separate database. The tables should, however, be setup in a separate, mirrored table space with a separate rollback segment.

Table and File-Based Restart/Recovery

The restart/recovery process works by storing all the data necessary to resume processing from the last commit point. Therefore, the necessary information is updated on the `RESTART_BOOKMARK` table before the processed data is committed. Query-based and file-based module stores different information on the restart tables, and therefore calls different functions within the restart/recovery API to perform their tasks.

When a program's process is query-based on a module is driven by a driving query that processes the retrieved rows, the information that is stored on the `RESTART_BOOKMARK` table is related to the data retrieved in the driving query. If the program fails while processing, the information that is stored on the restart-tables can be used in the conditional where-clause of the driving query to only retrieve data that has yet to be processed since the last commit event.

File-based processing, needs to store the file location at the time of the last commit point. This file's byte location is stored on the `RESTART_BOOKMARK` table and will be retrieved at the time of a restart. This location information will be used further be used in reopening the file when the data was last committed. Because there is different information being saved to and retrieved from the `RESTART_BOOKMARK` table for each of the different types of processing, different functions need to be called to perform the restart/recovery logic. The query-based processing calls the `RESTART_INIT` or `RETEK_INIT` and `RESTART_COMMIT` or `RETEK_COMMIT` functions while the file-based processing calls the `RESTART_FILE_INIT` and `RESTART_FILE_COMMIT` functions.

In addition to the differences in API function calls, the batch processing flow of the restart/recovery differs between the files. Table-based restart/recovery needs to use a priming fetch logical flow, while the file-based processing usually reads lines in a batch. Table-based processing requires its structure to ensure that the LUW key has changed before a commit event is allowed to occur, while the file-based processing does not need to evaluate the LUW, which can typically be thought of as the type of transaction being processed by the input file.

The following diagram depicts *table*-based Restart/Recovery program flow:

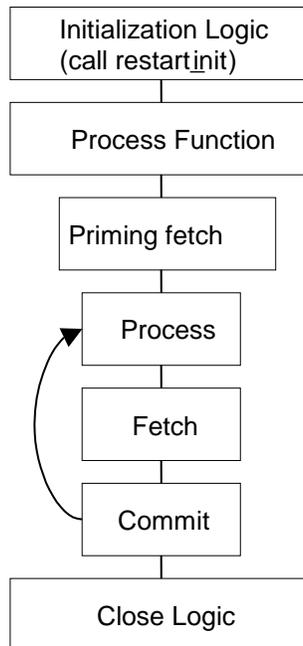
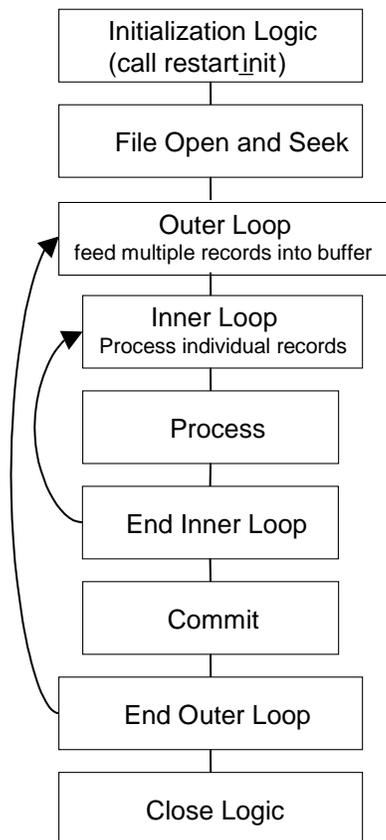


Table-Based Restart/Recovery Program Flow

The following diagram depicts *file*-based Restart/Recovery program flow



File-based Restart/Recovery Program Flow

Initialization logic:

- Variable declarations
- File initialization
- Call `restart_init()` or `restart_file_init()` function - determines start or restart logic
- First fetch on driving query

Start logic: initialize counters/accumulators to start values

Restart logic:

- Parse `application_image` field on bookmark table into counters/accumulators
- Initialize counters/accumulators to values of parsed fields

Process/commit loop:

- Process updates and manipulations
- Fetch new record
- Create varchar from counters/accumulators to pass into `application_image` field on `restart_bookmark` table
- Call `restart_commit()` or `restart_file_commit()`

Close logic:

- Reset pointers
- Close files/cursors
- Call `Restart_close()`

API Functional Descriptions

RESTART_INIT

An initialization functions for table-based batch processing.

The batch process gathers the following information from the restart control tables:

- Total number of threads for a program and thread value assigned to current process.
- Number of records to loop through in driving cursor before commit (LUW).
- Start string - bookmark of last commit to be used for restart or a null string if current process is an initial start and initializes the restart record-keeping (`restart_program_status`).
- Program status is changed to 'started' for the first available thread.
- Operational information is updated: operator, process, `start_time`, and bookmarking (`restart_bookmark`) tables.
- On an initial start, a record is inserted.
- On restart, the start string and application context information from the last commit is retrieved.
- Lock wait time and maximum retry for locking.

RESTART_FILE_INIT

An initialization functions for file-based batch processing. It is called from program modules.

- The batch process gathers the following information from the restart control tables:
- Number of records to read from file for array processing and for commit cycle.
- File start point- bookmark of last commit to be used for restart or 0 for initial start.

- The process initializes the restart record-keeping (`restart_program_status`):
- Program status is changed to 'started' for the current thread.
- Operational information is updated: operator, process, `START_TIME`.
- The process initializes the restart bookmarking (`restart_bookmark`) tables:
- On an initial start, a record is inserted.
- On restart, the file starting point information and application context information from the last commit is retrieved.

RESTART_COMMIT

A function that commits the processed transaction for a given number of driving query fetches. It is called from program modules.

The process updates the `RESTART_BOOKMARK` start string and application image information if a commit event has taken place:

- The current number of driving query fetches is greater than or equal to the maximum set in the `restart_program_status` table (and fetched in the `restart_init` function).
- The bookmark string of the last processed record is greater than or equal to the maximum set in the `restart_program_status` table (and fetched in the `restart_init` function).
- The bookmark string increments the counter.
- The bookmark string sets the current string to be the most recently fetched key string.

RESTART_FILE_COMMIT

A function that commits processed transactions after reading a number of lines from a flat file. It is called from program modules.

The process updates the `RESTART_BOOKMARK` table:

- `Start_string` is set to the file pointer location in the current read of the flat file.
- Application image is updated with context information.

RESTART_CLOSE

A function that updates the restart tables after program completion.

The process determines whether the program was successful. If the program finished successfully:

- The `restart_program_status` table is updated with finish information and the status is reset.
- The corresponding record in the `RESTART_BOOKMARK` table is deleted.
- The `restart_program_history` table has a copy of the `restart_program_status` table record inserted into it.
- The `restart_program_status` is re-initialized.

If the program ends with errors:

- The transactions are rolled back.
- The `program_status` column on the `restart_program_status` table is set to 'aborted in *' where * is one of the three main functions in batch: `init`, `process` or `final`.
- The changes are committed.

PARSE_ARRAY_ARGS

This function parses a string into components and places results into multidimensional array. It is only called within API functions and will never be called in program modules.

The process is passes a string to parse, and a pointer to an array of characters.

The first character of the passed string is the delimiter.

RESTART_FILE_WRITE

This function will append output in temporary files to final output files when a commit point is reached. It is called from program modules.

RESTART_CAT

This function contains the logic that appends one file to another. It is only called within the restart/recovery API functions and never called directly in program modules.

Restart Headers and Libraries

The RESTART.H and the STD_ERR.H header files are included in RETEK.H to utilize the restart/recovery functionality.

RESTART.H

This library header file contains constant, macro substitutions, and external global variable definitions as well as restart/recovery functions prototypes.

The global variables that are defined include:

- The thread number assigned to the current process.
- The value of the current process's thread maximum counter:
 - For table-based processing, it is equal to the number of iterations of the driving query before a commit can take place.
 - For file-based processing, it is equal to the number of lines that is read from a flat file and processed using a structured array before a commit can take place.
- The current count of driving query iterations used for table-based processing or the current array index used in file-based processing.
- The name assigned to the program/logical unit of work by the programmer. It is the same as the restart_name column on the restart_program_status, restart_program_history, and RESTART_BOOKMARK tables.

STD_REST.H

This library header file contains standard restart variable declarations that are used in the program modules.

The variable definitions that are included are:

- The concatenated string value of the fetched driving query key that is currently being processed.
- The concatenated string value of the fetched driving query key that will be processed next.
- The error message passed to the restart_close function and updated to restart_program_status.
- Concatenated string of application context information, for example, counters and accumulators.

- The name of the threading driver, for example, department, store, warehouse and others.
- The total number of threads used by this program.
- The pointer to pass to initialization function to retail number of threads value.

Updating Restart Headers and Libraries

Restart/recovery performs the following, among other capabilities:

- Organizes global variables associated with restart recovery.
- Allows the batch developer full control of restart recovery variables parameter passing during initialization.
- Removes temporary write files to speed up the commit process.
- Moves more information and processing from the batch code into the library code.
- Adds more information into the restart recovery tables for tuning purposes.

retk_2.h

This library header file is included by all C code within Retail and serves to centralize system includes, macro defines, globals, function prototypes, and, especially, structs for use in the new restart/recovery library.

The globals used by the old restart/recovery library are all discarded. Instead, each batch program declares variables needed and calls `retk_init()` to get them populated from restart/recovery tables. Therefore, only the following variables are declared:

- `gi_no_commit`: flag for `NO_COMMIT` command line option (used for tuning purposes).
- `gi_error_flag`: fatal error flag.
- `gi_non_fatal_err_flag`: non-fatal error flag.

In addition, a `rtk_file` struct is defined to handle all file interfaces associated with restart/recovery. Operation functions on the file struct are also defined.

```
#define NOT_PAD          1000 /* Flag not to pad thread_val */
#define PAD              1001 /* Flag to pad thread_val at the end */
#define TEMPLATE        1002 /* Flag to pad thread_val using filename template
*/
#define MAX_FILENAME_LEN 50
typedef struct
{
    FILE* fp; /* File pointer */
    char filename[MAX_FILENAME_LEN + 1]; /* Filename */
    int pad_flag; /* Flag whether to pad thread_val to filename */
} rtk_file;

int set_filename(rtk_file* file_struct, char* file_name, int pad_flag);
FILE* get_FILE(rtk_file* file_struct);
int rtk_print(rtk_file* file_struct, char* format, ...);
int rtk_seek(rtk_file* file_struct, long offset, int whence);
```

The parameters that `retk_init()` needs to populate passed using a format known to `retk_init()`. A struct is defined here for this purpose. An array of parameters of this struct type is needed in each batch program. Other requirements have to be initialized at each batch program.

- The lengths of name, type and sub_type should not exceed the definitions here.
- Type can only be: "int", "uint", "long", "string", or "rtk_file".
- For type "int", "uint" or "long", use "" as sub_type.

- For type "string", sub_type can only be "S" (start string) unless the string is the thread value or number of threads, in which case use "" as sub_type or "I" (image string).
- For type "rtk_file", sub_type can only be "I" (input) or "O" (output).

```
#define NULL_PARA_NAME      51
#define NULL_PARA_TYPE     21
#define NULL_PARA_SUB_TYPE  2
typedef struct
{
    char name[NULL_PARA_NAME];
    char type[NULL_PARA_TYPE];
    char sub_type[NULL_PARA_SUB_TYPE];
} init_parameter;
```

INT RETEK_INIT(INT NUM_ARGS, INIT_PARAMETER *parameter, ...)

RETEK_INIT initializes restart/recovery (for both table and file-based):

- Pass in NUM_ARGS as the number of elements in the INIT_PARAMETER array, and then the INIT_PARAMETER array, then variables a batch program needs to initialize in the order and types defined in the INIT_PARAMETER array. Note that all int, uint and long variables need to be passed by reference.
- Get all global and module level values from databases.
- Initialize records for RESTART_PROGRAM_STATUS and RESTART_BOOKMARK.
- Parse out user-specified initialization variables (variable arg list).
- Return NO_THREAD_AVAILABLE if no qualified record in RESTART_CONTROL or RESTART_PROGRAM_STATUS.
- Commit work.

INT RETEK_COMMIT(INT NUM_ARGS, ...)

RETEK_COMMIT checks and commits if needed (for both table and file-based):

- Pass in num_args, then variables for start_STRING first, and those for image string (if needed) second. The num_args is the total number of these two groups. All are string variables and are passed in the same order as in retek_init().
- Concatenate start_string either from passed in variables (table-based) or from ftell of input file pointers (file-based).
- Check if commit point is reached (counter check and, if table-based, start string comparison).
- If reached, concatenated image_string from passed in variables (if needed) and call internal_commit() to get out_file_string and update RESTART_BOOKMARK.
- If table-based, increment pl_current_count and update ps_cur_string.

INT COMMIT_POINT_REACHED(INT NUM_ARGS, ...)

COMMIT_POINT_REACHED checks if the commit point is reached (for both table- and file-based). The difference between this function and the check in RETEK_COMMIT() is that here the PL_CURRENT_COUNT and PS_CUR_STRING are not updated. This checking function is designed to be used with RETEK_FORCE_COMMIT(), and the logic to ensure integrity of LUW exists in user batch program. It can also be used together with RETEK_COMMIT() for extra processing at the time of commit.

- Pass in num_args, then all string variables for start_string in the same order as in retek_init(). The num_args is the number of variables for start_string. If no start_string (as in file-based), pass in NULL.

- For table-based, if `pl_curren_count` reaches `pl_max_counter` and if newly concatenated bookmark string is different from `ps_cur_string`, return 1; otherwise return 0.
- For file-based, if `pl_curren_count` reaches `pl_max_counter` return 1; otherwise return 0.

INT RETEK_FORCE_COMMIT(INT NUM_ARGS, ...)

RETEK_FORCE_COMMIT always commits (for both table and file-based):

- Pass in `num_args`, then variables for `start_string` first, and those for image string (if needed) second. The `num_args` is the total number of these two groups. All are string variables and are passed in the same order as in `retek_init()`.
- Concatenate `start_string` either from passed in variables (table-based) or from `ftell` of input file pointers (file-based).
- Concatenated `image_string` from passed in variables (if needed) and call `internal_commit()` to get `out_file_string` and update `RESTART_BOOKMARK`.
- If table-based, increment `pl_current_count` and update `ps_cur_string`.

INT RETEK_CLOSE(VOID)

RETEK_CLOSE closes restart/recovery (for both table and file-based):

- If `gi_error_flag` or `NO_COMMIT` command line option is TRUE, rollback all database changes.
- Update `RESTART_PROGRAM_STATUS` according to `gi_error_flag`.
- If no `gi_error_flag`, insert record into `RESTART_PROGRAM_HISTORY` with information fetched from `RESTART_CONTROL`, `RESTART_PROGRAM_BOOKMARK` and `RESTART_PROGRAM_STATUS` tables.
- If no `gi_error_flag`, delete `RESTART_BOOKMARK` record.
- Commit work.
- Close all opened file streams.

INT RETEK_REFRESH_THREAD(VOID)

You must refresh a program's thread, so that it can be run again.

- Updates the `RESTART_PROGRAM_STATUS` record for the current program's `PROGRAM_STATUS` to be 'ready for start'.
- Deletes any `RESTART_BOOKMARK` records for the current program.
- Commits work.

VOID INCREMENT_CURRENT_COUNT(VOID)

INCREMENT_CURRENT_COUNT increases `PL_CURRENT_COUNT` by 1.

Note: This is called from `GET_RECORD()` of `infrface.pc` for file-based input/output.

INT PARSE_NAME_FOR_THREAD_VAL(CHAR* NAME)

PARSE_NAME_FOR_THREAD_VAL parses thread value from the extension of the specified file name.

INT IS_NEW_START(VOID)

IS_NEW_START checks if current run is a new start; if yes, return 1; otherwise 0.

Query-Based Commit Thresholds

The restart capabilities revolve around a program's logical unit of work (LUW). A batch program processes transactions and enables commit points based on the LUW. An LUW is comprised of a transaction key (such as item-store) and a maximum commit counter. Commit events occur after a given number of transaction keys are processed. At the time of the commit, key data information that is necessary for restart is stored in the restart table. In the event of a handled or un-handled exception, transactions will be rolled back to the last commit point. Upon restart the restart key information will be retrieved from the tables so that processing can resume with the unprocessed data.

Pro*C Multi-Threading

Processing multiple instances of a given program can be accomplished through “threading”. This requires driving cursors to be separated into discrete segments of data to be processed by different threads. This is accomplished through a stored procedure that separates threading mechanisms (for example, departments or stores) into particular thread given value (for example, department 1001) and the total number of threads for a given process.

File-based processing will not “thread” its processing. The same data file will never be acted upon by multiple processes. Multi-threading is accomplished by dividing the data into separate files using a separate process for each file. The thread value is related to the input file. This is necessary to ensure that the appropriate information can be tied back to the relevant file in the event of a restart.

The store length in RMS is ten digits. Therefore, thread value which is based on the store number must allow ten digits. As the thread value is declared as a ‘C’ variable of type integer (long), the system restricts thread values to nine digits.

This does not mean that you cannot use ten digit store numbers. It means that if you do use ten digit store numbers you cannot use them as thread values.

Threading Description

The use of multiple threads or processes in Oracle Retail batch processing increases efficiency and decrease processing time. The design of the threading process allows maximum flexibility to the end user in defining the number of processes over which a program is divided.

Originally, the threading function was used directly in the driving queries. This was a slow process. Instead of using the function call directly in the driving queries, the designs call for joining driving query tables to a view (for example, v_restart_store) that includes the function.

Threading Function for Query-Based

A stored procedure has been created to determine thread values. Restart_thread_return returns a thread value derived from a numeric driver value, such as department number, and the total number of threads in a given process. Retailers should be able to determine the best algorithm for their design, and if a different means of segmenting data is required, then either the restart_thread_return function can be altered, or a different function can be used in any of the views in which the function is contained.

Currently the restart_thread_return function is a very simple modulus routine:

```
CREATE OR REPLACE FUNCTION RESTART_THREAD_RETURN(in_unit_value NUMBER,
                                                in_total_threads NUMBER)
RETURN NUMBER IS
  ret_val NUMBER;
BEGIN
  ret_val := MOD(ABS(in_unit_value),in_total_threads) + 1;
  RETURN ret_val;
END;
```

Restarting View for Query-Based

Each restart view will have four elements:

- The name of the threading mechanism, *driver_name*.
- The total number of threads in a grouping, *num_threads*.
- The value of the driving mechanism, *driver_value*.
- The thread value for that given combination of *driver_name*, *num_threads*, and *driver_value*, *thread_val*.

The view will be based on the *restart_control* table and an information table such as *DEPS* or *STORES*. A row will exist in the view for every driver value and every total number of threads value. Therefore, if a retailer were to always use the same number of threads for a given driver (dept, store, and others), then the view would be relatively small. As an example, if all of a retailer's programs threaded by department have a total of 5 threads, then the view will contain only one value for each department. For example, if there are 10 total departments, 10 rows will exist in *v_restart_dept*. However, if the retailer wants to have one of the programs to have ten threads, then there will be 2 rows for every department: one for five total threads and one for ten total threads (for example, if 10 total departments, 20 rows will exist in *v_restart_dept*). Obviously, retailers should be advised to keep the number of total thread values for a thread driver to a minimum to reduce the scope of the table joins of the driving cursor with the view.

Below is an example of how the same driver value can result in differing thread values. This example uses the *restart_thread_return* function as it currently is written to derive thread values.

Driver_name	num_threads	driver_val	thread_val
DEPT	1	101	1
DEPT	2	101	2
DEPT	3	101	3
DEPT	4	101	2
DEPT	5	101	2
DEPT	6	101	6
DEPT	7	101	4

Below is an example of what a distribution of stores might look like given 10 stores and 5 total threads:

Driver_name	num_threads	driver_val	thread_val
STORE	5	1	2
STORE	5	2	3
STORE	5	3	4
STORE	5	4	5
STORE	5	5	1
STORE	5	6	2
STORE	5	7	3
STORE	5	8	4

Driver_name	num_threads	driver_val	thread_val
STORE	5	9	5
STORE	5	10	1

View syntax:

The following is an example of the syntax needed to create the view for the multi-threading join, created with script (see threading discussion for details on RESTART_THREAD_RETURN function):

```
create or replace view v_restart_store as
  select rc.driver_name driver_name,
         rc.num_threads num_threads,
         s.store driver_value,
         restart_thread_return(s.store, rc.num_threads) thread_val
  from restart_control rc, store s
  where rc.driver_name = 'STORE'
```

There is a different threading scheme used within Oracle Retail Sales Audit (ReSA). Because ReSA needs to run 24 hours a day and seven days a week, there is no batch window. This means that there may be batch programs running at the same time that there are online users. ReSA solved this concurrency problem by creating a locking mechanism for data that is organized by store days. These locks provide a natural threading scheme. Programs that cycle through all of the store day data attempt to lock the store day first. If the lock fails, the program simply goes on to the next store day. This has the affect of automatically balancing the workload between all of the programs executing.

Thread Scheme Maintenance

All program names will be stored on the restart_control table along with a functional description, the query driver (department, store, class, and others) and the user-defined number of threads associated with them. Users should be able to scroll through all programs to view the name, description, and query driver, and if the update_allowed flag is set to true, to modify the number of threads (update is set to true).

File-Based

File-based processing does not truly “multi-thread” and therefore the number of threads defined on restart_control will always be one. However, a restart_program_status record needs to be created for each input file that is to be processed for the program module. Further, the thread value that is assigned must be part of the input file name. The restart_parse_name function included in the program module will parse the thread value from the program name and use that to determine the availability and restart requirements on the RESTART_PROGRAM_STATUS table.

Refer to the beginning of this multi-threading section for a discussion of limits on using large (greater than nine digits) thread values.

Query-Based

When the number of threads is modified in the `restart_control` table, the form must first validate that no records for that program are currently being processed in the `restart_program_status_table` (that is, all records = 'Completed'). The program must insert or delete rows depending on whether the new thread number is greater than or less than the old thread number. In the event that the new number is less than the previous number, all records for that `program_name` with a thread number greater than the new thread number will be deleted. If the new number is greater than the old number, new rows will be inserted. A new record is inserted for each `RESTART_NAME/THREAD_VAL` combination.

For example if the batch program SALDLY has its number of processes changed from 2 to 3, then an additional row (3) will be added to the `restart_program_status` table. In the same way, if the number of threads was reduced to 1 in this example, rows 2 and 3 would be deleted.

Row	RESTART_NAME	THREAD_VA	program_name
1	SALDLY	1	SALDLY
2	SALDLY	2	SALDLY

`RESTART_PROGRAM_STATUS` table after insert:

Row	RESTART_NAME	THREAD_VA	program_name
1	SALDLY	1	SALDLY
2	SALDLY	2	SALDLY
3	SALDLY	3	SALDLY

`RESTART_PROGRAM_STATUS` table after delete:

Row	RESTART_NAME	THREAD_VA	program_name
1	SALDLY	1	SALDLY

Users should also be able to modify the `commit_max_ctr` column in `restart_program_status` table. This controls the number of iterations in driving query or the number of lines read from a flat file that determine the logical unit of work (LUW).

Batch Maintenance

Users must be able to view the status of all records in `restart_program_status` table. This is where the user gets to view error messages from aborted programs, and statistics and histories of batch runs. The only fields that are modified will be `program_status` and `restart_flag`. The user should be able to reset the `restart_flag` to 'Y' from 'N' on records with a status of aborted, started records to aborted in the event of an abend (abnormal termination), and all records in the event of a restore from tape/re-run of all batch.

Scheduling and Initialization of Restart Batch

Before any batch with restart/recovery logic is run, an initialization program must be run to update the status in the `restart_program_status` table. This program must update the `program_status` to 'ready for start' wherever a record's `program_status` is 'completed.' This leaves unchanged all programs that ended unsuccessfully in the last batch run.

Pre-Processing and Post-Processing

Due to the nature of threading algorithm, individual programs needs a pre or a post program run to initialize variables or files before any of the threads have run or to update final data when all the threads are run. The decision was made to create pre-programs and post-programs in these cases rather than let the restart/recovery logic decide whether the currently processed thread is the first thread to start or the last thread to end for a given program.

ksh Driven Batch Programs

For ksh driven batch programs that call PL/SQL for its main processing logic, multi-threading is also supported. An example of this type of batch job is ksh script stockcountupload.ksh calling PL/SQL package CORESVC_STOCK_UPLOAD_SQL. The threading configuration for each program is defined in table RMS_PLSQL_BATCH_CONFIG (instead of RESTART_CONTROL for the ProC programs). Column MAX_CONSURRENT_THREAD holds the maximum number of concurrent threads. MAX_CHUNK_SIZE defines the commit size within each thread, similar to the RESTART_CONTROL.COMMIT_MAX_CTR column.

Pro*C Array Processing

Oracle Retail batch architecture uses array processing to improve performance. Instead of processing SQL statements using scalar data, data is grouped into arrays and used as bind variables in SQL statements. This improves performance by reducing the server/client and network traffic.

Array processing is used in selecting, inserting, deleting, and updating statements. Oracle Retail typically does not statically define the array sizes, but uses the restart maximum commit variable as a sizing multiple. The user must remember this when defining the system's maximum commit counters.

Remember, when using array processing in Oracle, it does not allow a single array operation to be performed for more than 32000 records in one step. The Oracle Retail restart/recovery libraries have been updated to define macros for this value: `MAX_ORACLE_ARRAY_SIZE`.

All batch programs that use array processing need to limit the size of their array operations to `MAX_ORACLE_ARRAY_SIZE`.

If the commit max counter is used for array processing size, check it after the call to `restart_init()` and, if necessary, reset it to the maximum value if greater. If `retek_init()` is used to initialize, check the returned commit max counter and reset it to the maximum size if it is greater. In case of `retek_init()`, reset the library's internal commit max counter by calling `extern int limit_commit_max_ctr(unsigned int new_max_ctr)`.

If some other variable is used for sizing the array processing, the actual array-processing step must be encapsulated in a calling loop that performs the array operation in sub segments of the total array size where each sub-segment is at most `MAX_ORACLE_ARRAY_SIZE` large. Currently all Oracle Retail batch programs are implemented in the similar way.

Pro*C Input and Output Formats

Oracle Retail batch processing utilizes input from both tables and flat files. Further, the outcome of processing can both modify data structures and write output data. Interfacing Oracle Retail with external systems is the main use of file based I/O.

General Interface Discussion

To simplify the interface requirements, Oracle Retail requires that all in-bound and out-bound file-based transactions adhere to standard file layouts. There are two types of file layouts, detail-only and master-detail, which are described in the sections below.

An interfacing API exists within Oracle Retail to simplify the coding and the maintenance of input files. The API provides functionality to read input from files, ensure file layout integrity, and write and maintain files for rejected transactions.

Standard File Layouts

The RMS interface library supports two standard file layouts; one for master/detail processing, and one for processing detail records only. True sub-details are not supported within the RMS base package interface library functions.

A 5-character identification code or record type identifies all records within an I/O file, regardless of file type. The following includes the valid record type values:

- FHEAD – File Header
- FDETL – File Detail
- FTAIL – File Tail
- THEAD – Transaction Header
- TDETL – Transaction Detail
- TTAIL – Transaction Tail

Each line of the file must begin with the record type code followed by a 10-character record ID.

Detail-Only Files

File layouts have a standard file header record, a detail record for each transaction to be processed, and a file trailer record. Valid record types are FHEAD, FDETL, and FTAIL.

Example:

```
FHEAD0000000001STKU1996010100000019960929
FDETL0000000002SKU100000040000011011
FDETL0000000003SKU100000050003002001
FDETL0000000004SKU100000050003002001
FTAIL00000000050000000003
```

Master and Detail Files

File layouts consists of:

- Standard file header record
- Set of records for each transaction to be processed
- File trailer record.

The transaction set consists of:

- Transaction set header record
- Transaction set detail for detail within the transaction
- Transaction trailer record

Valid record types are FHEAD, THEAD, TDETL, TTAIL, and FTAIL.

Example:

```
FHEAD0000000001RTV 19960908172000
THEAD000000000200000000000001199609091202000000000003R
TDETL000000000300000000000001000001SKU10000012
TTAIL0000000004000001
THEAD000000000500000000000002199609091202001215720131R
TDETL000000000600000000000002000001UPC400100002667
TDETL0000000007000000000000020000021UPC400100002643 0
TTAIL0000000008000002
FTAIL0000000009000000007
```

Record Name	Field Name	Field Type	Default Value	Description
File Header	File Type Record Descriptor	Char(5)	FHEAD	Identifies file record type.
	File Line Identifier	Number(10)	Specified by external system	Line number of the current file.
	File Type Definition	Char(4)	n/a	Identifies transaction type.
	File Create Date	Date	Create date	Date file was written by external system.
Transaction Header	File Type Record Descriptor	Char(5)	THEAD	Identifies file record type.
	File Line Identifier	Number(10)	Specified by external system	Line number of the current file.
	Transaction Set Control Number	Char(14)	Specified by external system	Used to force unique transaction check.
	Transaction Date	Char(14)	Specified by external system	Date the transaction was created in external system.
Transaction Detail	File Type Record Descriptor	Char(5)	TDETL	Identifies file record type.
	File Line Identifier	Number(10)	Specified by external system	Line number of the current file.

Record Name	Field Name	Field Type	Default Value	Description
	Transaction Set Control Number	Char(14)	Specified by external system	Used to force unique transaction check.
	Detail Sequence Number	Char(6)	Specified by external system	Sequential number assigned to detail records within a transaction.
Transaction Trailer	File Type Record Descriptor	Char(5)	TTAIL	Identifies file record type.
	File Line Identifier	Number(10)	Specified by external system	Line number of the current file.
	Transaction Detail Line Count	Number(6)	Sum of detail lines	Sum of the detail lines within a transaction.
File Trailer	File Type Record Descriptor	Char(5)	FTAIL	Identifies file record type.
	File Line Identifier	Number(10)	Specified by external system	Line number of the current file.
	Total Transaction Line Count	Number(10)	Sum of all transaction lines	All lines in file less the file header and trailer records.

RETL Program Overview for the RMS-RPAS Interface

This chapter covers information about the Oracle Retail Extract Transform and Load (RETL) program overview for the RMS and the RPAS interface. The RETL architecture is mentioned along with the RETL program overview.

Oracle Retail ETL Architecture

RMS works with the Oracle Retail Extract Transform and Load (RETL) framework. This architecture utilizes a high performance data processing tool that allows database batch processes to take advantage of parallel processing capabilities.

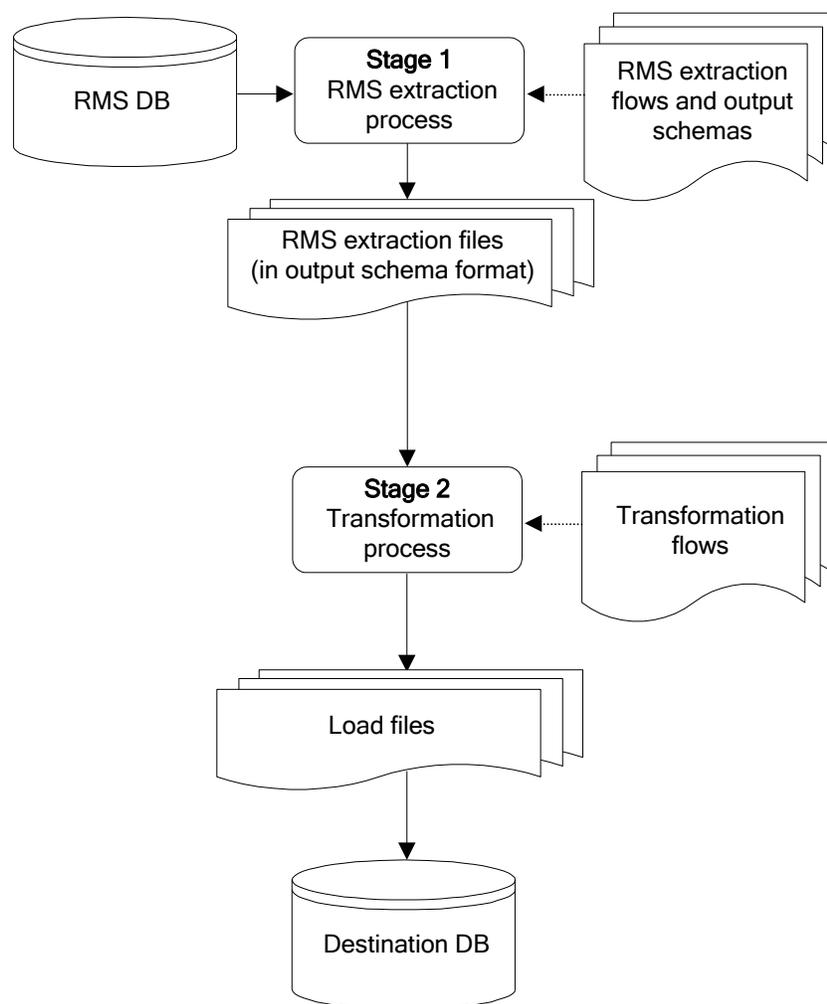
The RETL framework runs and parses through the valid operators composed in XML scripts. For more information on RETL tool, see *RETL Programmer's Guide*.

The figure [The two stages of RETL processing](#) illustrates the extraction processing architecture. Instead of managing the change captures as they occur in the source system during the day, the process involves extracting the current data from the source system. The extracted data is output to flat files. These flat files are then available for consumption by products such as RPAS.

The target system has its own way of completing the transformations and loading the necessary data into the system, where it can be used for further processing.. For more information on transformation and loading, see *RPAS documentation*.

The architecture relies upon two distinct stages, shown in the figure [The two stages of RETL processing](#).

- **Stage 1-** Extraction from the RMS database using well-defined flows specific to the RMS database. The resulting output is comprised of data files written in a well-defined schema file format. This stage includes no destination specific code.
- **Stage 2-** Introduces a flow specific to the destination. In this case, flows for RPAS are designed to transform the data so that RPAS can import the data properly.



The two stages of RETL processing

RETL Program Overview

This section summarizes the RETL program features utilized in the RMS extractions and loads. Installation information about the RETL tool is available in the latest RETL Programmer's Guide.

Configuration

Version of RETL

Before trying to configure and run RMS RETL, install RETL version 13.0 or later, which is required for RMS RETL to run. For thorough installation procedure, see *RETL Programmer's Guide*.

RETL User and Permissions

The permissions are set up as per the RETL Programmer's Guide. RMS RETL reads and writes data files, you can also create, delete, update and insert into tables. The extraction fails, if it is not set up properly.

Environment Variables

See the RETL Programmer's Guide for RETL environment variables that must be set up for your version of RETL. You will need to set `RDF_HOME` to your base directory for RMS RETL. This is the top level directory that you select during the installation process. Add `export RDF_HOME=<base directory for RMS RETL>` in `profile`.

`rmse_rpas_config.env` Settings for RPAS

There are several constants that must be set in `rmse_rpas_config.env` depending upon a retailer's preferences and the local environment. This is summarized in the table below.

Constant Name	Default Value	Alternate Value	Description
<code>DATE_TYPE</code>	<code>vdate</code>	<code>current_date</code>	Determines whether the date used in naming the error, log, and status files is the current date or the <code>VDATE</code> value found in the <code>PERIOD</code> table.
<code>DBNAME</code>	<code>rtkdev01</code>	Depends on installation	The database schema name.
<code>RMS_OWNER</code>	<code>RPASINT</code>	Depends on installation	The username of the RMS database schema owner.
<code>BA_OWNER</code>		Depends on installation	The username of the RMS batch user (not currently used by RMS-RPAS).
<code>CONN_TYPE</code>	<code>thin</code>	<code>oci</code>	The way in which RMS connects to the database.
<code>DBHOST</code>	<code>mospdev17</code>	Depends on installation	The computer hardware node name.
<code>DBPORT</code>	<code>1524</code>	Depends on installation	The port on which the database listener resides.
<code>LOC_ATTRIBUTES_ACTIVE</code>	<code>False</code>	<code>True</code>	Determines whether <code>rmse_rpas_attributes.ksh</code> is run or not.
<code>PROD_ATTRIBUTES_ACTIVE</code>	<code>False</code>	<code>True</code>	Determines whether <code>rmse_rpas_attributes.ksh</code> is run or not.
<code>DIFFS_ACTIVE</code>	<code>True</code>	<code>False</code>	Determines whether <code>rmse_rpas_merchhier.ksh</code> generates data files that contain diff allocation information.

Constant Name	Default Value	Alternate Value	Description
ISSUES_ACTIVE	True	False	If set to 'True', rmse_rpas_stock_on_hand also extracts stock at the warehouse level. If set to 'False', rmse_rpas_stock_on_hand extracts stock at the store level only.
LOAD_TYPE	CONVENTIONAL	DIRECT	Data loading method to be used by SQL*Loader (Direct may be faster than conventional).
DB_ENV	ORA	DB2, TERA	Database type (Additional changes to the software may be needed if a database other than Oracle is selected).
NO_OF_CPUS	4	Depends on installation	Used in parallel database query hints to improve performance.
LANGUAGE	en	Various	En = English
RFX_OPTIONS	-c \$RDF_HOME/ rfx/etc/rfx.conf -s SCHEMAFILE	-c \$RDF_ HOME/ rfx/etc/rfx .conf	Processing speed may be increased for some extractions if the -s SCHEMAFILE option is omitted.

You must set up the wallet alias in the `rmse_aip_config.env`. The wallet and wallet alias creation is required to use programs in a secured mode. The following variables must be setup `RETL_WALLET_ALIAS`, `ORACLE_WALLET_ALIAS`, `SQLPLUS_LOGON`.

Be sure to review the environmental parameters in the `rmse_rpas_config.env` file before executing batch modules.

Steps to Configure RETL

1. Log in to the UNIX server with a UNIX account that will run the RETL scripts.
2. Change directories to `<base_directory>/rfx/etc`.
3. Modify the constants from the table above in the `rmse_rpas_config.env` script as needed.

Program Return Code

RETL programs use a return code to indicate successful completion. If the program successfully runs, a zero (0) is returned. If the program fails, no number is returned.

Program Status Control Files

To prevent a program from running while the same program is already running against the same set of data, the code utilizes a program status control file. At the beginning of each module, `rmse_rpas_config.env` is run. This script checks for the existence of the program status control file. If the file exists, then a message stating, '`${PROGRAM_NAME}` has already started', is logged and the module exits. If the file does not exist, a program status control file is created and the module executes.

If the module fails at any point, the program status control file is not removed, and the user is responsible for removing the control file before running the module again.

File Naming Conventions

The name and directory of the program status control file is set in the configuration script (`rmse_rpas_config.env`). The directory defaults to `$RDF_HOME/error`.

The naming convention for the program status control file itself defaults to the following dot separated file name:

- The program name.
- The program status is mentioned under the 'status' parameter.
- The business virtual date for which the module was run.

For example, a program status control file for the `rmse_rpas_daily_sales.ksh` program is named as follows for a batch run on the business virtual date of January 5, 2001:

```
$RDF_HOME/error/rmse_rpas_daily_sales.status.20010105
```

Restart and Recovery

Because RETL processes all records as a set, as opposed to one record at a time, the method for restart and recovery must be different from the method that is used for Pro*C. The restart and recovery process serves the following two purposes:

1. It prevents the loss of data due to program or database failure.
2. It increases performance when restarting after a program or database failure by limiting the amount of reprocessing that needs to occur.

The RMS extract (RMSE) modules extract from a source transaction database or text file and write to a text file. The RMS loads module import data from flat files, performs necessary transformations, and then loads the data into the applicable RMS table.

Most modules use a single RETL flow and do not require the use of restart and recovery. If the extraction process fails for any reason, the problem can be fixed, and the entire process is started again without the loss of data. No RMS to RPAS extraction programs has any restart/recovery capability. The single RMS load program, `rmsl_rpas_forecast.ksh`, takes a text file as its input, and the following choices are available, that enables the program to complete the load in the event of an error:

- a. Re-run the program with the entire input file.
- b. Re-run the program with only the input records that were not processed successfully the first time.

Message Logging

Message logs are written daily in a format as described in this section.

Daily Log File

Every RETL program writes a message to the daily log file when it starts and when it finishes. In some cases, progress messages are also written. The name and directory of the daily log file is set in the configuration script (`rmse_rpas_config.env`). The directory defaults to `$RDF_HOME/log`. All log files are encoded UTF-8.

The naming convention of the daily log file defaults to the following “dot” separated file name:

- The business virtual date for which the modules are run.
- The log file format should be `.log`.

For example, the location and the name of the log file for the business virtual date of January 5, 2001 would be the following:

```
$RDF_HOME/log/20010105.log
```

Format

As the following examples illustrate, every message written to a log file has the name of the program, a timestamp, and either an informational or error message. For example:

```
rmse_rpas_item_retail 17:09:07: Program started ...
rmse_rpas_item_retail 17:09:12: Program completed successfully
```

Some error messages are also written to the log file, such as `'No output file specified'`.

Program Error File

In addition to the daily log file, each program also writes its own detailed flow and error messages. Rather than clutter the daily log file with these messages, each program writes out its errors to a separate error file unique to each execution.

If a program finishes unsuccessfully, a message is usually written in the error file that indicates where the problem occurred in the process.

The name and directory of the program error file is set in the applicable configuration file (`rmse_rpas_config.env`). The directory defaults to `$RDF_HOME/error`. All errors and *all routine processing messages* for a given program on a given day go into this error file (for example, it contains both the `stderr` and `stdout` produced during execution of the program).

The naming convention for the program's error file defaults to the following “dot” separated file name:

- The program name
- The business virtual date for which the module was run

For example, all errors and detailed log information for the `rms_item_master.ksh` program would be placed in the following file for the batch run on the business virtual date of January 5, 2001:

```
$MMHOME/error/rms_item_master.20010105
```

RMSE Reject Files

RMSE extract modules may produce a reject file if they encounter data related problems, such as the inability to find data on required lookup tables. The module tries to process all data and then indicates that records were rejected so that all data problems can be identified in one pass and corrected; then, the module can be re-run to successful completion. If a module does reject records, the reject file is not removed, and the user is responsible for removing the reject file before re-running the module. The records in the reject file consist of the rejected records.

The name and directory of the reject file are defined in the applicable configuration script (`rmse_rpas_config.env`). The directory defaults to `$RDF_HOME/data`.

Note: A directory specific to reject files can be created. The `rmse_rpas_config.env` script would need to be changed to define the reject directory constant such that it would point to that directory.

The naming convention for the reject file defaults to the following “dot” separated file name:

- The program name.
- The first filename, if one is specified on the command line.
- The file format should be as `‘.rej’`.
- The business virtual date for which the module was run.

Schema Files Overview

RETL uses schema files to specify the format of incoming or outgoing datasets. The schema file defines each column’s data type and format, which is then used within RETL to format/handle the data. For more information about schema files, see the latest RETL Programmer’s Guide. Schema file names are hard-coded within each module because they do not change on a day-to-day basis. All schema files end with `“.schema”` and are placed in the `“$RDF_HOME/rfx/schema”` directory.

Command Line Parameters

The only programs or scripts that allows the command line parameters (or arguments) are:

- `rmse_rpas_config.env` script
- `pre_rmse_rpas.ksh`
- `rmse_rpas.ksh` programs.

All of the command line parameters for these modules are optional and are described in section [rmse_rpas_config.env](#), the square brackets indicate that the parameter is optional.

rmse_rpas_config.env

Usage: \$RDF_HOME/rfx/etc/rmse_rpas_config.env [-t \$*] [-r \$*] [-s \$*] [-v \$* | -c \$*]

Description of Command Line Options

Note: See the end of this description for an explanation of the need for the '\$*' that appears after each command line option.

- **-t:** This option causes rmse_rpas_config.env to skip the initializing of the environment variables that obtains the values from the '.txt' files, except for VDATE which is initialized with the date found in the vdate.txt file. This option is utilized by pre_rmse_rpas.ksh, rmse_rpas.ksh, rdft.ksh and outage.ksh when they call rmse_rpas_config.env.
- **-r:** This option prevents the redirection of all output (stdout and stderr) to the error file. This can be useful during debugging and maintenance. This option can also be utilized by rmse_rpas.ksh, rdft.ksh and outage.ksh when they call rmse_rpas_config.env.
 - The '-t' and '-r' options must be followed by '\$*' on the line which invokes this script. This step is necessary in order to preserve the command line arguments or options that may have been present on the command line for the RETL script that invokes this script. However, the '\$*' must only appear once if both options are used.
- **-s:** This option causes rmse_rpas_config.env to skip the STATUS_FILE test. This is also useful during maintenance and debugging.
- **-v:** If DATE_TYPE (in rmse_rpas_config.env) is set to 'vdate', this option prevents the normal exit with an error message when the vdate.txt file is empty or non-existent; instead, it uses the current date to derive FILE_DATE. However, if DATE_TYPE is set to 'vdate', and vdate.txt actually does exist and is non-empty, the date in vdate.txt continues to be used even if this option is set. If DATE_TYPE is set to 'current_date', this option has no effect.
- **-c:** This option overrides the DATE_TYPE switch setting and causes the current date to be used to derive FILE_DATE regardless of what DATE_TYPE is set to. This option is utilized by pre_rmse_rpas.ksh when it calls rmse_rpas_config.env, if it is run with the -c option on its command line. The '-c' option is normally only used when rmse_rpas_config.env is called from pre_rmse_rpas.ksh.
 - If only one command line option is used, it must be followed by '\$*'. But if more than one option is specified, then '\$*' must be entered on the command line only once after all options have been entered. The '\$*' is necessary in order to preserve the command line arguments or options (if there are any) that are present on the command line that is used to execute the RETL script which invokes this script.
 - If more than one option is specified, options must appear on the command line in the same order as shown on the "Usage" line, above.

pre_rmse_rpas.ksh

- Usage: pre_rmse_rpas.ksh [-c]
- The '-c' option is used to specify what option is to be placed on the rmse_rpas_config.env command line when it is called by this program. It is usually used the first time that pre_rmse_rpas.ksh is run at a new installation or if the state of the vdate.txt file is unknown. This option is passed directly to rmse_rpas_config.env when it is called by pre_rmse_rpas.ksh. No other use is made of this parameter by pre_rmse_rpas.ksh.
- This option causes rmse_rpas_config.env to use the current date to initialize FILE_DATE instead of possibly setting it to VDATE, which is obtained from the vdate.txt file. (FILE_DATE is the date that is used to name the error, log, and status files).
- The current date is used regardless of how DATE_TYPE is set in rmse_rpas_config.env. By using the '-c' option, there is no need to manually set up the vdate.txt file before running this script.
- The normal mode for pre_rmse_rpas.ksh (without the -c option) is that when it calls rmse_rpas_config.env, FILE_DATE is set to VDATE or the current date, depending on how DATE_TYPE is set in rmse_rpas_config.env. If DATE_TYPE is set to 'vdate', and if the vdate.txt file does not exist or is empty, rmse_rpas_config.env (and this program) exits with an error message.
- The use of this option does not affect what date is used by any of the other RETL scripts that run after this script is done. After pre_rmse_rpas.ksh has run, when the other RETL scripts are run, they call rmse_rpas_config.env with no options on the command line, and their files are named using VDATE or the current date, depending on how DATE_TYPE is set in rmse_rpas_config.env.

rmse_rpas.ksh

- Usage: rmse_rpas.ksh [-c]
- The presence of the '-c' option causes FILE_DATE in rmse_rpas_config.env to be set to the current date instead of possibly using VDATE (which gets its value from the vdate.txt file), but only when it is called by rmse_rpas.ksh and pre_rmse_rpas.ksh (pre_rmse_rpas.ksh is invoked by rmse_rpas.ksh). It has no effect when other extract programs call rmse_rpas_config.env, at the time that they are invoked by rmse_rpas.ksh. This option is passed directly to rmse_rpas_config.env and pre_rmse_rpas.ksh when they are called by rmse_rpas.ksh. No other use is made of this parameter by rmse_rpas.ksh.

Note: During the rmse_rpas_config.env configuration to connect to the JDBC, enter the connection details as follows:
 jdbc:oracle:thin:@<DB host server>:<PORT ID>:<connection string>. For example,
 oracle:thin:@msp52417.us.oracle.com:1521:dolsp01app.

RMSE I/O File Names

Most of the output path/filenames have the format, \$DATA_DIR/(RMSE_RPAS_program name).dat. Similarly, the schema format for the records in these files are specified in the file - \$SCHEMA_DIR/(RMSE_RPAS_program name).schema.

Typical Run and Debugging Situations

The following examples illustrate typical run and debugging situations for programs. The log, error, etc. file names referenced below assumes that the module is run on the business virtual date of March 9, 2001. For more information on naming convention, see [File Naming Conventions](#) section.

For example:

To run `rmse_rpas_stores.ksh`:

1. Change directories to `$RDF_HOME/rfx/src`.
2. At a UNIX prompt (\$) enter:

```
$rmse_rpas_stores.ksh
```

If the module runs successfully, the following results:

3. **Log file:** Today's log file, `20010309.log`, contains the messages "Program started ..." and "Program completed successfully" for `rmse_rpas_stores`.
4. **Data:** The `rmse_rpas_stores.dat` file exists in the data directory and contains the extracted records.
5. **Schema:** The `rmse_rpas_stores.schema` file exists in the schema directory and contains the definition of the data file in #2 above.
6. **Error file:** The program's error file, `rmse_rpas_stores.20010309`, contains the standard RETL flow (ending with "All threads complete" and "Flow ran successfully") and no error messages.
7. **Program status control:** The program status control file, `rmse_rpas_stores.status.20010309`, will not exist.
8. **Reject file:** The reject file, `rmse_rpas_stores.rej.20010309`, will not exist.

If the module does *not* run successfully, the following results:

9. **Log file:** Today's log file, `20010309.log`, does not contain the "Program completed successfully" message for `rmse_rpas_stores`.
10. **Data:** The `rmse_rpas_stores.dat` file may exist in the data directory but may not contain all the extracted records.
11. **Schema:** The `rmse_rpas_stores.schema` file exists in the schema directory and contains the definition of the data file in #2 above.
12. **Error file:** The program's error file, `rmse_rpas_stores.20010309`, may contain one or more error messages.
13. **Program status control:** The program status control file, `rmse_rpas_stores.status.20010309`, exists.
14. **Reject file:** The reject file, `rmse_rpas_stores.status.20010309`, does not exist because this module does not reject records.

To re-run the module, perform the following actions:

15. Determine and fix the problem causing the error.
16. Remove the program's status control file.
17. Change directories to `$RDF_HOME/rfx/src`. At a UNIX prompt, enter:
`$rmse_rpas_stores.ksh`

RPAS/AIP Configuration

This section covers information about the configuration of RPAS/AIP. It gives an overview of handling installation and configuration in RMS ETL.

RETL Program Overview for the RMS-Time-Phased Inventory Planning Tool Interface

This section summarizes the RETL program features utilized in the RMS Extractions (RMSE) for the RMS-time-phased inventory planning tool integration. Starting with RMS version 11, the RMS extract for a time-phased inventory planning tool is separate from the RMS extracts for RPAS. In prior RMS version, time-phased inventory planning tool and RPAS had common RETL extracts.

More installation information about the RETL tool is available in the latest RETL Programmer's Guide.

Note: In this section, some examples refer to RETL programs that are not related to RMS or are related to other versions of RMS than this document addresses. Such examples are included for illustration purposes only.

Installation

Select a directory where you would like to install RMS ETL. This directory (also called MMHOME) is the location from which the RMS ETL files are extracted.

The following code tree is utilized for the RETL framework during the extractions, transformations, and loads and is referred to in this documentation.

```
<base directory (MMHOME)>
  /data
  /error
  /log
  /rfx
      /bookmark
      /etc
      /lib
      /schema
      /src
```

Configuration

RETL

Before trying to configure and run RMS RETL, install RETL version 13.0 or later, which is required to run RMS ETL. For thorough installation process, see the latest RETL Programmer's Guide.

RETL user and permissions

RMS ETL is installed and run as the RETL user. Additionally, the permissions are set up as per the RETL Programmer's Guide. RMS ETL reads data, creates, deletes and updates tables. If these permissions are not set up properly, extractions fail.

Environment Variables

To set up RETL environment variable for the RETL version you have installed, see RETL Programmers Guide. You will need to set MMHOME to your base directory for RMS RETL. This is the top level directory that you selected during the installation process (see the section, 'Installation', above). In your .kshrc, you should add a line such as the following:

```
export MMHOME=<base directory for RMS ETL>
```

rmse_aip_config.env Settings

There variables you must be change depending upon your local settings:

For example:

```
export DBNAME=int9i
export RMS_OWNER=steffej_rms1011
export BA_OWNER=rmsint1011
```

You must set up the wallet alias in the `rmse_aip_config.env`. The wallet and wallet alias creation is required to use programs in a secured mode. The following variables must be setup `RETL_WALLET_ALIAS`, `ORACLE_WALLET_ALIAS`, `SQLPLUS_LOGON`.

Make sure to review the environmental parameters in the `rmse_aip_config.env` file file before executing batch modules.

Steps to Configure RETL

1. Log in to the UNIX server with a UNIX account that will run the RETL scripts.
2. Change directories to `<base_directory>/rfx/etc`.
3. Modify the `rmse_aip_config.env` script.

For example:

- a. Change the `DBNAME` variable to the name of the RMS database.
- b. Change the `RMS_OWNER` variable to the username of the RMS schema owner.
- c. Change the `BA_OWNER` variable to the username of the RMSE batch user.

Note: During the `rmse_aip_config.env` configuration to connect to the JDBC, enter the connection details as follows: `jdbc:oracle:thin:@<DB host server>:<PORT ID>:<connection string>`. For example, `oracle:thin:@msp52417.us.oracle.com:1521:dolsp01app`.

Program Return Code

RETL program uses one return code to indicate successful completion. If the program successfully runs, a zero (0) is returned. If the program fails then, zero is not returned.

Program Status Control Files

To prevent a program from running while the same program is already running against the same set of data, the code utilizes a program status control file. At the beginning of each module, `rmse_aip_config.env` is run. These files check for the existence of the program status control file. If the file exists, then a message stating, `'${PROGRAM_NAME} has already started'`, is logged and the module exits. If the file does not exist, a program status control file is created and the module executes.

If the module fails at any point, the program status control file is not removed, and the user is responsible for removing the control file before re-running the module.

File Naming Conventions

The naming convention of the program status control file allows a program whose input is a text file to be run multiple times at the same time against different files.

The name and directory of the program status control file is set in the applicable configuration file (rmse_aip_config.env). The directory defaults to \$MMHOME/error. The naming convention for the program status control file itself defaults to the following dot separated file name:

- The program name.
- The status of the program name should be specified in 'status'.
- The business virtual date for which the module was run.

For example, a program status control file for one program is named as follows for the batch run of January 5, 2001:

```
$MMHOME/error/rmse_aip_banded_item.status.20010105
```

Restart and Recovery

Because RETL processes all records as a set, as opposed to one record at a time, the method for restart and recovery must be different from the method that is used for Pro*C. The restart and recovery process serves the following two purposes:

1. It prevents the loss of data due to program or database failure.
2. It increases performance when restarting after a program or database failure by limiting the amount of reprocessing that needs to occur.

The RMS Extract (RMSE) module extracts a source transaction database or text file and writes to a text file. The RMS Load (RMSL) modules import data from flat files, performs transformation if necessary and then loads the data into the applicable RMS tables.

Most modules use a single RETL flow and do not require the use of restart and recovery. If the extraction process fails for any reason, the problem is fixed, and the entire process runs from the beginning without the loss of data. For a module that takes a text file as its input, the following two choices are available that enables the module to be re-run from the beginning:

3. Re-run the module with the entire input file.
4. Re-run the module with only the records that were not processed successfully the first time and concatenate the resulting file with the output file from the first time.

To limit the amount of data that needs to be re-processed, more complicated modules that require the use of multiple RETL flows utilize a bookmark method for restart and recovery. This method allows the module to be restarted at the point of last success and run to completion. The bookmark restart/recovery method incorporates the use of a bookmark flag to indicate which step of the process should be run next. For each step in the process, the bookmark flag is written to and read from a bookmark file.

Note: If the fix for the problem causing the failure requires changing data in the source table or file, then the bookmark file must be removed and the process must be re-run from the beginning in order to extract the changed data.

Message Logging

Message logs are written daily in a format described in this section.

Daily Log File

Every RETL program writes a message to the daily log file when it starts and when it finishes. The name and directory of the daily log file is set in the configuration file (`rmse_aip_config.env`). The directory defaults to `$MMHOME/log`. All log files are encoded UTF-8.

The naming convention of the daily log file defaults to the following “dot” separated file name:

- The business virtual date for which the modules are run
- `’.log’`

For example, the location and the name of the log file for the business virtual date of January 5, 2001 would be the following:

```
$MMHOME/log/20010105.log
```

Format

As the following examples illustrate, every message written to a log file has the name of the program, a timestamp, and either an informational or error message:

```
aipt_item 17:07:43: Program started ...
aipt_item 17:07:50: Program completed successfully
rmse_aip_item_master 17:08:53: Program started ...
rmse_aip_item_master 17:08:59: Program completed successfully
rmse_item_retail 17:09:07: Program started ...
rmse_item_retail 17:09:12: Program completed successfully
```

If a program finishes unsuccessfully, an error file is usually written that indicates where the problem occurred in the process. There are some error messages written to the log file, such as `’No output file specified’`, that require no further explanation written to the error file.

Program Error File

In addition to the daily log file, each program also writes its own detail flow and error messages. Rather than clutter the daily log file with these messages, each program writes out its errors to a separate error file unique to each execution.

The name and directory of the program error file is set in the applicable configuration file (`rmse_aip_config.env`). The directory defaults to `$MMHOME/error`. All errors and *all routine processing messages* for a given program on a given day go into this error file (for example, it will contain both the `stderr` and `stdout` from the call to RETL). All error files are encoded UTF-8.

The naming convention for the program’s error file defaults to the following “dot” separated file name:

- The program name
- The business virtual date for which the module was run

For example, all errors and detail log information for the `rms_aip_item_master` program would be placed in the following file for the batch run of January 5, 2001:

```
$MMHOME/error/rms_aip_item_master.20010105
```

RMSE and Transformation Reject Files

RMSE extract and transformation modules may produce a reject file if they encounter data related problems, such as the inability to find data on required lookup tables. The module tries to process all data and then indicates that records were rejected so that all data problems can be identified in one pass and corrected; then, the module can be re-run to successful completion. If a module does reject records, the reject file is not removed, and the user is responsible for removing the reject file before re-running the module.

The record in the reject file contains an error message and key information from the rejected record. The following example illustrates a record that is rejected due to problems within the currency conversion library:

```
Currency Conversion Failed|101721472|20010309
```

The following example illustrates a record that is rejected due to problems looking up information on a source table:

```
Unable to find item_master record for Item|101721472
```

The name and directory of the reject file is set in the applicable configuration file (`rmse_config.env` or `config.env`). The directory defaults to `$MMHOME/data`.

Note: A directory specific to reject files can be created. The `rmse_config.env` and/or `config.env` file would need to be changed to point to that directory.

The naming convention for the reject file defaults to the following “dot” separated file name:

- The program name
- The first filename, if one is specified on the command line
- ‘rej’
- The business virtual date for which the module was run

For example, all rejected records for the `slsildmex` program would be placed in the following file for the batch run of January 5, 2001:

```
$MMHOME/data/slsildmex.slsildmdm.txt.rej.20010105
```

Schema Files Overview

RETL uses schema files to specify the format of incoming or outgoing datasets. The schema file defines each column’s data type and format, which is then used within RETL to format/handle the data. For more information about schema files, see the latest RETL Programmer’s Guide. Schema file names are hard-coded within each module since they do not change on a day-to-day basis. All schema files end with “.schema” and are placed in the “rfx/schema” directory.

Command Line Parameters

In order for each RETL module to run, the input/output data file paths and names may need to be passed in at the UNIX command line.

RMSE and Transformation

Most RMSE and transformation modules do not require the passing in of any parameters. The output path/filename defaults to `$DATA_DIR/(RMSE and transfer program name).dat`. Similarly, the schema format for the records in these files are specified in the file - `$SCHEMA_DIR/(RMSE program name).schema`.

Scripts that need Parameter to Run

The scripts below are run on a full snapshot basis. The parameter is F (for full snapshot).

- `rmse_aip_store_cur_inventory.ksh`
- `rmse_aip_wh_cur_inventory.ksh`

Typical Run and Debugging Situations

The following examples illustrate typical run and debugging situations for types of programs. The log, error, and other file names referenced below assume that the module is run on the business virtual date of March 9, 2001. See the previously described naming conventions for the location of each file.

For example:

To run `rmse_aip_store.ksh`:

1. Change directories to `$MMHOME/rfx/src`.
2. At a UNIX prompt enter:

```
%rmse_aip_store.ksh
```

If the module runs successfully, the following results:

- **Log file:** Today's log file, `20010309.log`, contains the messages "Program started ..." and "Program completed successfully" for `rmse_aip_store`.
- **Data:** The `rmse_aip_store.dat` file exists in the data directory and contains the extracted records.
- **Schema:** The `rmse_aip_store.schema` file exists in the schema directory and contains the definition of the data file in #2 above.
- **Error file:** The program's error file, `rmse_aip_store.20010309`, contains the standard RETL flow (ending with "All threads complete" and "Flow ran successfully") and no additional error messages.
- **Program status control:** The program status control file, `rmse_aip_store.status.20010309`, does not exist.
- **Reject file:** The reject file, `rmse_aip_store.rej.20010309`, does not exist because this module does not reject records.
- If the module does *not* run successfully, the following results:
 - **Log file:** Today's log file, `20010309.log`, does not contain the "Program completed successfully" message for `rmse_stores`.
 - **Data:** The `rmse_aip_store.dat` file may exist in the data directory but may not contain all the extracted records.
 - **Schema:** The `rmse_aip_store.schema` file exists in the schema directory and contains the definition of the data file in #2 above.
 - **Error file:** The program's error file, `rmse_aip_store.20010309`, may contain an error message.
 - **Program status control:** The program status control file, `rmse_aip_store.status.20010309`, exists.
 - **Reject file:** The reject file, `rmse_aip_store.status.20010309`, does not exist because this module does not reject records.

To re-run the module, perform the following actions:

1. Determine and fix the problem causing the error.
2. Remove the program's status control file.
3. Change directories to \$MMHOME/rfx/src. At a UNIX prompt, enter:
%rmse_aip_store.ksh.

Internationalization

Internationalization is the process of creating software that can be translated more easily. Changes to the code are not specific to any particular market. RMS is internationalized to support multiple languages.

This section describes configuration settings and features of the software that ensure that the base application can handle multiple languages.

Translation

Translation is the process of interpreting and adapting text from one language into another. Although the code itself is not translated, components of the application that are translated may include the following:

- Graphical user interface (GUI)
- Error messages
- Reports
- Spreadsheets

The following components are not translated:

- Documentation (online help, release notes, installation guide, user guide, operations guide)
- Batch programs and messages
- Log files
- Configuration tools
- Demonstration data
- Training materials

The user interface for RMS is translated into:

- Arabic
- Chinese (simplified)
- Chinese (traditional)
- Croatian
- Dutch
- English
- French
- German
- Greek
- Hungarian
- Italian
- Japanese
- Korean
- Polish
- Portuguese (Brazilian)
- Russian

- Spanish
- Swedish
- Turkish

Translation Support for Multiple Entities

Several of the RMS entities supports translation in which the name or description or some text field for that entity can be maintained in other supported languages in addition to the data integration language for that installation. There are more than 100 tables for which a corresponding translation tables exists. The general naming convention for the translation table is to suffix the entities table name with `_TL`. For example, `ITEM_MASTER_TL` is the translation table for `ITEM_MASTER`.

The entities in the translation table are loaded based on the usage of the entities. The entities which are populated during install time also contain the seed script in all supported languages. This includes tables such as `UOM_LANG`, `CODE_DETAIL`.

The entities which are user created will have the capability to maintain the translation values for individual entities. This includes entities such as items, merchandise hierarchy. The translation support is “inbuilt” in both user interface for entities maintained via UI and also in spreadsheet as a separate worksheet for entities maintained via spreadsheet upload.

RMS User Interface Language Display Settings

The user can select a preferred language through the RMS UI user preference menu. The language selected in user preference can be defaulted for all subsequent logins for that user or can be configured only for that session only. The default language for the user if the user preference is not set is the browser default language.

Note: During installation, the primary language for the installation is English. This can be changed to any one of the supported language during installation or even after installation. The primary language is also referred as Data Integration language. All the inbound and outbound integration with RMS will always have the names, descriptions, and so on in the data integration language only even though those values are maintained in multiple languages.

For every user session, the application user language is updated to Database `SYS_CONTEXT` with the namespace `RETAIL_CTX` and the parameter `APP_ISO_LANGUAGE`. The backend packages and database views refers to the application user language and for the entities supporting language entries, the data is fetched in the application’s user language.

Maintenance of Translation Values through the User Interface

The data presented in the screen, in which the entities supporting translation is maintained, changes depending on the user language configuration and the mode in which the user launches the screen.

The values loaded in the base table of the entity are always maintained in the data integration language for the installation. All users, irrespective of the configured language, maintain the name or description field for that entity during Create or Edit mode in data integration language only.

If the user enters an entity page in View mode, the name or description field for that entity is visible in the user language if entries exist in the entities translation table for the user language. If the user language entry is not same as data integration language and there are no entries in entities translation table, the data in the data integration language is presented to the user.

The entities which support translation and are maintained through the user interface have two ways to maintain the translation entry through the UI.

1. In the main entity page, the user is presented with a globe icon. Clicking on the globe icon opens a translation popup in which the user is able to view and maintain other language entries for that entity and the columns that support translation. For example, in the case of the Item screen, the user is allowed to maintain translation values in all supported language for item description, short description and secondary description. The language entries for item are maintained in the `item_master_tl` table.
2. A generic Translation screen is available on the left navigation link to allow for the maintenance of data across multiple entities through a single screen. The user can choose the entity for which translation entries have to be maintained, and all the data for that entity will be presented to the user. The user can choose to filter to identify the specific record, and for that record, the translation entry in all supported language can be maintained.

The two translation screens above get dynamically rendered based on metadata of the translation tables.

The metadata tables used by the translation screens are

- `TRANSLATE_METADATA` to hold the table to column mapping and the variable mapping to interact with the PLSQL package where the updates to the database table are implemented.
- `TRANSLATE_METADATA_TABLE` to hold the list of tables supported by the generic translation screen.

Maintenance of Translation Values through Spreadsheet Uploads

The spreadsheet upload process supports maintenance of the translation entry for entities having translation support and maintained via spreadsheet uploads. The spreadsheet has an additional worksheet with the default name of the worksheet suffixed with `_TL` to indicate that it is a translation worksheet. The maintenance of names or description columns in primary language occurs via the worksheet for the base entity. The maintenance of the translatable columns in the non-data integration language occurs using the translation worksheet.

Integrating RMS with Store Inventory Management

Oracle Retail Store Inventory Management (ORSIM or SIM) is a Java based application with multitier architecture. SIM assists store operation, tracking item and item inventory with all the integrated systems.

SIM helps store personnel in performing the following in-store operations:

- Receiving merchandise from the warehouse or directly from the vendor.
- Replenishing and order stock.
- Requesting and implementing price changes.
- Managing physical inventories and performing stock count.
- Lookup for the detailed information about merchandise items, suppliers, containers, and customer orders.
- Transferring or returning stock.

SIM function includes administration, shipping, receiving, inventory management, lookups and reporting.

The administration function is performed either by the SIM administrator or the Manager. The function includes:

- Setup and technical maintenance of SIM.
- Security setup to define SIM users and their roles.
- Setup and maintenance of serial numbers that are Unique Identification Numbers (UIN) based.

The usage of serial numbers for item is an optional feature.

The inventory management function helps to maintain an accurate perpetual store Inventory for the Stock Counts, Sequencing, Pick Lists, Item Requests, Store Orders, Price Changes, and Ticketing.

The inventory management system provides detailed information about inventory items, suppliers, containers and customer orders which are created in RMS. You can check for the related information in parallel. For example, when you are checking for an item, the supplier information of the item can also be viewed.

Using inventory management, you can perform lookup along with the other SIM tasks. For example, lookup for an item while preparing an item request or lookup for a supplier information when preparing the store order.

SIM includes many standard reports, which can be customized as per your requirement. SIM uses Oracle BI Publisher as an interface for the SIM reports. The SIM reports are custom designed as per the organizational requirement. The BI Publisher interface is also customized to organize and present the reports available to the SIM users.

All the location, item and supplier information are created in RMS whether it is ranged or not ranged to a store, flows to SIM using the RIB Adapter.

Supplier

Supplier is created in RMS. The details of the supplier are sent to SIM through RIB.

RIB Validation: After successful execution of the batches, check for the stores in the RIB.

- VENDOR_PUBLISHER in RIB-RMS
- VENDOR_SUBSCRIBER in RIB-SIM

When RIB message shows as succeeded, the vendor will be available in the SIM.

Merchandise Hierarchy

Merchandise hierarchy includes Department, Class, and Sub-Class which are created in RMS. After they are created successfully in RMS, the details are sent to SIM through RIB.

RIB Validation: After successful execution of the batches, check for the stores in the RIB.

- MERCHHIER_PUBLISHER in RIB-RMS
- MERCHHIER_SUBSCRIBER in RIB-SIM

When RIB message shows as succeeded, the Department, Class and Subclass is available in SIM.

Warehouse

When a warehouse is created in the RMS, the details of warehouse are sent to SIM through RIB.

RIB Validation: After successful execution of the batches, check for the stores in the RIB.

- WH_PUBLISHER in RIB-RMS
- WH_SUBSCRIBER in RIB-SIM

When RIB message shows as succeeded, warehouse will be available in SIM.

SIM Store

The SIM Store function allows you to set operating parameters for stores managed with SIM. The Store is created in RMS and flows to SIM application.

RIB Validation: After successful creation of the stores, check for the stores message in the RIB.

- STORES_PUBLISHER in RIB-RMS
- STORE_SUBSCRIBER in RIB-SIM

When RIB message shows as succeeded, the store is available in SIM.

Inventory Adjustment Reason

You can add, change, and delete reason codes used for inventory adjustments. In addition to showing the reasons for inventory adjustments, the reason code specifies how inventory adjustments affect stock on hand, unavailable inventory, or customer order reserve inventory.

Note: The Reason Codes created in SIM must be mapped to the RMS Inventory Adjustment Reason Codes.

Adding Inventory Adjustment Reason in SIM

To add an Inventory Adjustment Reason in SIM:

1. Navigate and select **Admin > Setup > Inv. Adj. Reason.** >The Inventory Adjustment Reason Maintenance window opens.
2. Click **Add.**
3. Enter ID: 123.
4. Description: Test Inventory Reason Code.
5. The UI must be check.
6. Select **Stock on Hand** from the Disposition dropdown list.
7. Click **Done.**
8. Validate the database in the INV_ADJUSTMENT_REASON table.
9. When the Inventory reason code is set up in SIM, it must be mapped to RMS.

Mapping Inventory Adjustment Reason in RMS

To map Inventory Adjustment Reasons in RMS:

1. Navigate to Foundation Data → Data Loading → Upload.
2. In Upload Data, select Inventory from the Template Type choice list.
3. Select Inventory Status and Adjustment Reasons from the Template list of values. The Process Description field gets auto-populated.
4. Click on the Browse button and select the data file with the Inventory Adjustment Reason values.
5. Click **Upload** to upload the values in the file into RMS. An information message will appear to confirm that the upload was processed successfully. Click OK on the information message.
6. Click **Done** to close the screen.

Database Validation

Functionality	SIM Table	RMS Table
Inventory Adjustment Reason	INV_ADJUSTMENT_REAS ON	INV_ADJ_REASON

Diff ID

To create a Diff ID in RMS, navigate and select **Items > Diffs > Diff Group/Diff IDs.**

RIB Validation: Validate the details in RIB.

- Diffs Publisher in RIB-RMS
- Diffs Subscriber in RIB-SIM

When the Diff ID is available in both publisher and subscriber, it will be available in SIM.

Item With/Without UIN and Item Locations

You can create the item with diff or without diff in RMS. To create you must navigate and select **Item> Items.** Select the items to be created in RMS. If you have associated the stores to the item, then you can validate the item availability in SIM.

RIB Validation: Validate the details in RIB.

- Items Publisher (for item) and ItemLoc Publisher (For location) in RIB-RMS.
- Items Subscriber and ItemLoc Subscriber in RIB-SIM.
- When the item and location are available in publisher and subscriber, it will be available in SIM.

Database Validation

Functionality	SIM Table	RMS Table
Item	ITEM	ITEM_MASTER
Country of Manufacture	SUPPLIER_ITEM_MANUFACTURE	ITEM_SUPP_MANU_COUNTRY
Item Supplier Relationship	SUPPLIER_ITEM	ITEM_SUPPLIER
Item Supplier Relationship	SUPPLIER_ITEM_COUNTRY	ITEM_SUPP_COUNTRY
Item Location Relationship	STORE_SEQUENCE_ITEM	ITEM_LOC

SIM GUI Item Look up

To view SIM GUI item lookup:

1. Log in to SIM and select the associated store.
2. Navigate and select **Lookups > Item Lookup**.
3. Enter the item created in RMS and search.
4. You get the details with the diff (optional) available in the Merchandise Hierarchy block.

Transactions

There are different types of transactions created and maintained in SIM.

1. Purchase Order
2. Transfer
3. Return to Warehouse
4. Return to Vendor
5. Store Order
6. Inventory Adjustment
7. Stock Count
8. Price Change

Purchase Order

To create a purchase order in RMS for a store you must navigate and select **Orders → Create Order**. When the order is created and submitted, approve the Purchase Order (PO). The order details is be sent to SIM successfully.

RIB Validation: When the purchase order is create, make sure to verify the order number for the following:

- ORDER_PUBLISHER in RIB-RMS
- ORDER_SUBSCRIBER in RIB-SIM

Receiving the Order in SIM

The Purchase Order is received with different combinations as given below:

1. Receive with Exact Quantity.
2. Under Receive Quantity.
3. Over Receive Quantity.
4. Receive with Damage Quantity.
5. PO On The FLY – In this scenario, the Purchase Order is created in SIM and not RMS. This can be done based on the item and supplier's detail, when we have the details in SIM a PO is displayed.

Transfers

Transfer implies to movement of goods from one location to another. Accordingly to SIM, transfer is either store to store transfer or warehouse to store transfer. Transfers can be generated in SIM also.

Store To Store Transfer

The Store to Store transfer is created for the movement of goods from Store A to Store B.

RIB Validation:

- Verify the Stock-Order/Transfer in Transfer_Publisher in RIB-RMS.
- Verify the Stock-Order/Transfer in StockOrder_Subscriber in RIB-SIM.

Transfer Receiving

To receive a transfer, select the receiving store in the SIM login form. For dispatching transfer from store to store:

1. Log in to the correct **FROM** store.
2. Navigate and select > **Shipping/Receiving > Transfers**.
3. Double-click the record and update the quantity column with the unit which needs to be transferred. The transfer functionality supports both overage and underage.
4. Click **Dispatch** and accept the warnings.

RIB Validation: When the dispatch is completed, a message is sent to RMS.

- ASNOUT_PUBLISHER in RIB SIM
- ASNOUT_SUBSCRIBER in RIB RMS

To receive the transfer from store to store:

5. Log in to the correct **TO** store.
6. Navigate and select **Shipping/Receiving > Transfers**.
7. Double-click the record and updates the quantity column with the unit which needs to be transferred. The transfer functionality supports overage and underage.
8. Click **Receive All** and accept the warning.

RIB Validation: When the receiving is done, a message is sent to RMS. Verify the order number in:

- Receiving_Publisher in RIB-SIM
- Receiving_Subscriber in RIB-RMS

RMS Database Validation

- Stock on Hand gets updated in ITEM_LOC_SOH table in RMS. Received quantity is deducted from FROM store and added in TO store.
- Tran code 37 and 38 is posted in TRAN_DATA table.

Warehouse To Store Transfer

In this case of transfer, the goods are moved from warehouse to another store. The user must login to the receiving store to receive the transfer. Before receiving in SIM, the transfer must to be shipped from the corresponding warehouse.

To receive the transfer from Warehouse to Store:

1. Log in to the correct **TO** store.
2. Navigate and select Shipping/Receiving > Warehouse Delivery.
3. Double-click the container ID which has received from Warehouse.
4. Edit quantities or record damages.
5. Click **Receive**. The status changes to Received.
6. Click **Confirm** and accept the warning message.

RIB Validation: When the transfer is received, a message is sent to RMS. Verify the order number in:

- Receiving_Publisher in RIB-SIM
- Receiving_Subscriber in RIB-RMS

RMS Database Validation:

- Stock on Hand gets updated in ITEM_LOC_SOH table in RMS. Received quantity is deducted from FROM Warehouse and added in TO store.
- Tran code 30 and 32 is posted in TRAN_DATA table.

SIM Database Validation:

- STORE_ITEM_STOCK is increased in SOH for the store.

Transfer On The Fly

You can create the transfers from one store to another. No other transfers are possible.

1. Navigate and select Shipping/**Receiving** > **Transfer** > **Create Transfer**.
2. Select **Transfer To store** from the dropdown.
3. Enter the item and shipped quantity.
4. Click **BOL** and select the pickup date.
5. Click **Dispatch** and accept the warning message.

To receive the transfer on the FLY:

6. The user must log in to the correct TO store.
7. Navigate and select Shipping/Receiving > Transfers.
8. Double-click on the record and updates the quantity column with the units which needs to be transferred. The transfer functionality supports overage and underage.
9. Click **Receive All** and accept the warning in the pop up.
10. Click **Done** to save changes and confirm the transfer.

RIB Validation: When receiving is done, message is sent to RMS. Verify the order number in:

- Receiving_Publisher in RIB-SIM
- Receiving_Subscriber in RIB-RMS

RMS Database Validation

- Stock on Hand is updated in ITEM_LOC_SOH table in RMS. The received quantity is deducted from FROM store and added in TO store.
- Tran code 30, 32 and 22 is posted in TRAN_DATA table.

Note:

- The Reason Codes created in SIM must be mapped to the RMS Inventory Adjustment Reason Codes.
 - If the external ID is a number and the user is external it implies that the transfer is generated in RMS.
 - If a transfer request is initiated in SIM this message is not published in RMS or any other integrated application.
 - A transfer handles damaged quantities which in turn are put in the non-sellable bucket in RMS.
 - Negative receiving though cannot be done nor can be transferred.
 - We can also receive multiple times against the same transfer for the expected quantity. This action will be allowed as long as the transfer is open and not restricted by expected quantity.
 - While transferring we can also add items which were not part of the original transfer but we cannot remove the original item while we transfer as long as we have stock on hand.
 - Non ranged items can also be part of the transfer. These are items which are not ranged to that particular location but it is ranged by raising the transfer.
-
-

Return to Warehouse

You can create, edit, and dispatch returns from the store to a company-owned warehouse, or directly to a vendor. If there is unavailable stock for a returned item, you have an option to use items from unavailable stock for the return. A completed (dispatched) return decreases available Stock on Hand.

The store to warehouse transfer is created in RMS, which is displayed as a return to warehouse in SIM. A similar transfer is created in SIM.

Note: Unless the return is dispatched from SIM, WMS cannot see the transaction.

Creating Return to Warehouse in SIM

To create a return to Warehouse in SIM:

1. Select the correct store from the dropdown in the Login page.
2. Navigate and select Shipping/Receiving > Returns.
3. Click **Create**.
4. In the Return Type field, select **Warehouse** from the dropdown.
5. Enter warehouse, inventory status, authorization number, item details, reason to return and quantity.
6. Click Dispatch.
7. Click **Done** to save the changes.

RIB Validation: After dispatch is cleared, a message is sent to RMS.

Verify the order number in:

- RTV_Publisher in RIB-SIM
- RTV_Subscriber in RIB-RMS

A transfer (RTW) is created in RMS with **APPROVED** status.

RMS/SIM Database Validation

- Stock on Hand gets updated in ITEM_LOC_SOH table in RMS. Inventory must be reduced from SIM and RMS by correct quantity.
- Tran code 30 and 32 is posted in TRAN_DATA table.

Creating Return to Warehouse in RMS

When initiated by RMS, a warehouse to store transfer is created in RMS, and is returned back to warehouse then it is called Return to Warehouse.

RMS functionality includes, create a transfer, return to warehouse in RMS and APPROVE it.

RIB Validation: When the dispatch is completed, a message is sent to RMS.

- RTV_Publisher in RIB-RMS
- RTV_Subscriber in RIB-SIM
- Transfer_Publisher in RIB-RMS

Dispatching in SIM

To dispatch to SIM:

1. Select the correct store from the login form.
2. Navigate and select Shipping/Receiving > Returns.
3. Select the transfer you need to return. The user ID will be external along with the external ID number which indicates that the transfer originated in RMS.
4. Double-click on the **Record**.
5. Update the quantity column with the unit which needs to be returned.
6. Click **Dispatch** and accept the warning message.

RIB Validation: When the dispatch is completed, a message is sent to RMS.

- RTV_Publisher in RIB-SIM
- RTV_Subscriber in RIB-RMS

RMS/SIM Database Validation:

- Stock on Hand is updated in ITEM_LOC_SOH table in RMS. Inventory should be reduced from SIM and RMS by correct quantity.
- Tran code 30 and 32 is posted in TRAN_DATA table.

Return to Vendor

Return to vendor can be created both in SIM and RMS.

Initiating SIM in RTV

Following the the steps to initiate SIM in RTV:

1. Select the correct store from the dropdown in the login page.
2. Navigate and select Main menu > Shipping/Receiving > Returns.
3. Click **Create**.
4. In the Return Type field, select **Supplier** from the dropdown.
5. Enter supplier, authorization number, item details, reason to return and quantity.
6. Click Dispatch.

RIB Validation: When the dispatch is done, message is sent to RMS

- RTV_Publisher in RIB-SIM
- RTV_Subscriber in RIB-RMS

A transfer (RTV) is created in RMS.

RMS/SIM Database Validation

- Stock on hand gets updated in ITEM_LOC_SOH table in RMS (Inventory should be reduced from SIM and RMS by correct quantity).
- Tran code 24 is posted in TRAN_DATA table.

Initiating RMS in RTV

When a Return to vendor along with a mass return transfer is created in RMS, and is returned back to vendor then it is called Return to Vendor.

Steps for creating Returns in SIM:

1. Select the correct store from the login form.
2. Navigate and select Main menu > Shipping/Receiving > Returns.
3. Select the transfer you need to return. (The user ID will be external along with the external ID number which indicates that the transfer originated in RMS).
4. Double-click on the record which needs to be returned.
5. Update the quantity column with the units which need to be returned.
6. Click **Dispatch** and accept the warning message.

RIB Validation: When dispatch is completed, message is sent to RMS

- RTV_PUBLISHER in RIB-SIM
- RTV_SUBSCRIBER in RIB-RMS

RMS/SIM Database Validation:

Stock on hand gets updated in ITEM_LOC_SOH table in RMS (Inventory should be reduced from SIM and RMS by correct quantity). Tran code 24 is posted in TRAN_DATA table.

Store Orders

Store ordering allows to View, Create, Modify, and Approve orders to a supplier or Transfer requests from a warehouse. Use store-level ordering to order items that are not set up for automatic replenishment, when items run short, or demand increases.

Creating a Store Order

Create a store orders to replenish items for which you have the authority to place orders from the store.

1. Log in to SIM with the correct store.
2. Navigate and select **Inv Mgmt > Store Orders**.
3. Click Create Order.
4. Specify a delivery date range.
5. Enter Item number, quantity, and UOM.
6. Click **Done**. Order is created in Pending Status.

Approving a Store Order

1. Log in to the SIM with the correct store.
2. Navigate and select **Inv Mgmt > Store Orders**.
3. Double-click on the store order to be approved.
4. Specify a delivery date range.
5. Click **Approve**. Order status is changed to Approved Status.

Database Validation

Functionality	SIM Table	RMS Table
Store Order	PRINT_STORE_ORDER_ITEM	STORE_ORDER
	PRINT_STORE_ORDER	N/A

Inventory Adjustment

Inventory adjustments that you enter in SIM are supplied to the merchandising system to adjust stock levels and maintain perpetual inventory. Inventory adjustments increment or decrement inventory levels such as Stock on Hand and unavailable inventory.

Each inventory adjustment contains a reason code that determines the disposition of the inventory being adjusted. For example, inventory removed for repair is added to the unavailable inventory and the Stock on Hand is decreased. When the items are returned to inventory, unavailable inventory is decreased and the Stock on Hand is increased.

1. Log in to SIM with the store where you need to do the inventory adjustment.
2. Navigate and select **Inv Mgmt > Inventory Adjustment**.
3. Click **Create**.
4. Enter the item number in the Item field.
5. Update the UOM, Quantity, and Reason.
6. Click **Done**.

Enter UINs for an Inventory Adjustment

If an item requires a serial number type UIN, the Quantity field is disabled on the Inventory Adjustment Detail window.

7. Double-click the **UIN Qty** field.
8. Click **Add**.
9. Enter valid number in the in the Unique Identification Number (UIN) field.
10. Click Save to return to the Inventory Adjustment Detail window.

RIB Validation: When the Inventory adjustment is completed in SIM, a message is sent to RMS to update the inventory stock. Message flows through RIB:

- InvAdjust_Publisher in RIB-SIM
- InvAdjust_Subscriber in RIB-RMS

SIM TABLES Validation:

- INV_ADJUSTMENT: STORE_ID, ITEM_ID, QUANTITY

RMS DATABASE Validation

- INV_ADJ: ITEM, ADJ_QTY, LOCATION
- INV_STATUS_QTY : ITEM, QTY

Stock Count

Stock counts are the principal tools to ensure that the perpetual inventory for a store remains accurate. For maximum flexibility in performing stock counts, SIM allows the following types of stock counts:

- **Unit** - Unit stock counts are scheduled counts that update the RMS and SIM inventory positions, but only for the physical count values. These counts are performed on regular schedules.
- **Unit and Amount** - A Unit and Amount count is an annual count that can be used to adjust the financial systems in a corporate merchandising system, in addition to updating inventory positions in SIM.
- **Problem Line** - Problem Line stock counts are similar to Unit counts. Problem Line product groups use.
 - Pre-defined criteria to the identify problem items. For example, you might use a Problem Line count for all of the items that have negative Stock on Hand values.
- **Ad hoc** - An ad hoc stock count is an unscheduled stock count that is initiated on a handheld device. An ad hoc count is similar to a Unit count, but the items are not pre-assigned and there is no schedule.

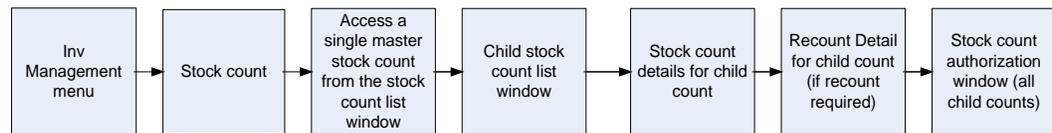
Each stock count must be generated, except for ad hoc stock count type. To generate a stock count, you must first create a product group and schedule. Product groups can include a particular inventory item or entire segments of the inventory hierarchy, including all items in a store.

Stock Counting Process

The general process to conduct a stock count in SIM is as follows:

1. Create the stock count product group.
2. Schedule the stock count.
3. Perform a stock count using both or any of the hand held devices or a personal computer.
4. Recount discrepant items, depending on whether the product group requires a recount.
5. Review the count information and authorize the count quantities.
6. For Unit and Amount counts, export the stock count results to the merchandising system to update Stock on Hand.

The following diagram illustrates the general process flow.



General Process Flow

Executing a Stock Count

The steps to execute a stock count from start to finish are as follows:

1. Take a Snapshot.
2. Enter Stock Count Results.
3. Enter Recount Results.
4. Complete a Child Stock Count.
5. Authorize a Stock Count.

RIB Validations: When the Stock Count is schedule, a message flows to RMS using RIB as interface:

- STKCOUNTSCH_PUBLISHER in RIB-SIM
- STKCOUNTSCH_SUBSCRIBER in RIB-RMS

Price Change

Using the Price Change function, you can request price changes for items at your store. Price changes are set in the pricing system. The pricing system might be in Oracle Retail Price Management (RPM) or another application.

You can request price changes only for items for which you can control prices. This is controlled by an indicator at the store (location) level. Your price change requests are submitted to the pricing system, where they may be approved or rejected. After a price change is approved in the pricing system, a price change event is sent from the pricing system to SIM.

A price change request that is approved in the pricing system creates a pricing event that is sent back to SIM. For an approved price change, you can print labels and tickets for the re-priced items.

Creating Price Change

You can request price changes for those items for which you are allowed to make pricing changes at the store (location) level. For these items, you can request changes to retail, promotion, or clearance pricing.

The request is sent to the pricing system, where it may be approved or rejected. Your price change requests are checked for possible conflicts. You cannot request price changes for items involved in complex promotions (such as buy one, get one free). You cannot request multiple price changes on the same day. Your request may be rejected if it conflicts with any other pricing events.

To create price change and promotions in SIM:

1. Navigate and select **Inv Mgmt > Price Change**.
2. Enter search criteria to limit the price change requests that you want displayed, and click **Search**.
3. Click **Create**.
4. Enter Item, start date, price change description and new price.
5. For clearance or promotion, select end date.
6. Click **Done** and accept the warning message.

RIB Validations: When the price is completed, a message is sent to ORPM using RIB as interface.

- PRCCHGREQ_PUBLISHER in RIB-SIM

SIM DATABASE VALIDATION

- ITEM_PRICE : ID_ITM, EFFECTIVE_DATE, UNIT_RETAIL, STATUS.

Integrating RMS with Oracle E-Business Suite Financials Using Oracle Retail Financial Integration

This chapter describes the integration between Oracle Retail systems and Oracle E-Business Suite Financials (including Oracle General Ledger and Oracle Payables), as developed and supported by Oracle Retail Financial Integration (ORFI).

When the option to integrate is chosen, the selected information is shared between the systems. Integration and validation services are in place to ensure the shared data matches.

Note: This chapter addresses the points within Oracle Retail systems that are essential to integration. For more information about the entire integration process, including mapping to Oracle E-Business Suite data and settings, see the ORFI documents, *Oracle Retail Financial Integration for Oracle Retail Merchandising Suite* and *Oracle E-Business Suite Financials - Implementation Guide*. For more information about Web services, see the following chapters in the *Oracle Retail Merchandising System Operations Guide, Volume 2: "Service Provider Implementations API Designs" and "Web Services"*.

Participating Applications

The following Oracle Retail applications are included in the integration covered by this chapter:

- Oracle Retail Merchandising System (RMS)
- Oracle Retail Sales Audit (ReSA)
- Oracle Retail Invoice Matching (ReIM)
- Oracle Retail Integration Bus (RIB)

Assumptions and Dependencies

- The option to integrate must be selected during initial set up of the RMS system.
- ReIM accesses RMS to determine if integration is active. The RMS set up must be done before integrating with REIM.
- The URLs for the RFI Web services that are necessary for this integration are maintained in the RETAIL_SERVICE_REPORT_URL table and in the ReIM IM_SYSTEM_OPTIONS table.
- Real time account validation is done only when the financial integration with Oracle E-Business Suite is ON.

- Partners must be set up as suppliers in Oracle E-Business Suite and manually set up in RMS using the RMS Supplier ID. The RMS supplier ID generated when the Oracle E-Business Suite supplier is interfaced with Oracle Retail. The RMS supplier generated as part of this process is not used.
- Payment terms and freight terms are manually maintained.

Data Setup

Integration of Oracle Retail applications and Oracle E-Business Suite Financials relies on synchronization of essential data, such as currency exchange rates and suppliers. Through careful discussions, the users of both systems determine the common codes and descriptions that will best serve their business needs.

When an agreement is reached, this information is set up and maintained. Depending on the volume, some shared information is set up in Oracle Retail applications or in Oracle E-Business Suite and electronically transferred to other systems. Otherwise, shared information is set up manually within each system, and the users of both systems must ensure that the code and the description match.

RMS Data Set up and Configuration

This section describes setup considerations for RMS data.

RMS System Options

As part of the RMS system options setup, set the following options as indicated:

1. The SYSTEM_OPTION indicates that the Oracle Retail system is integrated with a financial system:
 - FINANCIAL_AP=A
2. A value of A indicates that the financial system to which RMS is interfaced is Oracle E-Business Suite through Oracle Retail Financial Integration (ORFI).
 - GL_ROLL_UP can be D/S/C
 - SUPPLIER_SITES_IND = Y
 - ORG_UNIT_IND = Y

Organization Units

Use the Upload Data screen (**RMS Tasks List > Foundation Data > Data Loading > Upload**) to define organizational units in RMS that match those being setup in Oracle E-Business Suite. When an organizational unit is entered in RMS, the valid organizational units are those associated with the Set Of Books (SOB) used for the general ledger interface.

Currency Exchange Rates

Currency exchange rate is used to translate the monetary value of one currency in terms of another. Depending on business needs, a Currency Exchange Rate Type of Operational or Consolidation is selected for use in all transactions.

This value is set up manually in RMS and mapped to Oracle E-Business Suite through the Currency Exchange Type mapping window.

Currency Exchange Rate data is owned by Oracle General Ledger and updates are sent to Oracle Retail applications.

1. Determine the Exchange Type being sent by Oracle General Ledger (for example, Consolidation or Operational) that you want RMS to use.
2. Update the FIF_CURRENCY_XREF for mapping the external exchange type being sent by Oracle General Ledger with RMS Exchange Type. For example, for Consolidation and Operational exchange types, the FIF_CURRENCY_XREF table holds the following entries:

FIF_EXCHANGE_TYPE	RMS_EXCHANGE_TYPE
C	C
O	O

Supplier Address Types

Within RMS, supplier information (such as Order From and Remit To addresses) is used for generating the purchase orders. Oracle Payables uses supplier information for payment generation. It is important that this information is synchronized.

Suppliers are created in Oracle Payables and exported to RMS. When FINANCIAL_AP is set to A, suppliers cannot be created using the RMS screens. However, if a supplier exists in RMS, all data values for the supplier (except supplier name and status) are updated using the RMS screens. The association of supplier sites to organization units is accessed only in view mode through RMS screens. One supplier site per supplier and organization unit combination can be marked as primary payment site.

Where SYSTEM_OPTIONS.FINANCIAL_AP is A, disable auto generate supplier/partner numbers and associated check boxes.

Note: Supplier information is created, updated and inactivated only in Oracle Payables. This information is transferred from Oracle Payables to the participating Oracle Retail applications, where additional retail-specific attributes are maintained.

Country Codes

When country codes are defined and seeded in RMS, ensure that country codes are mapped to Oracle E-Business Suite country codes through RFI DVM mapping. The following is an example of RFI DVM Mapping (Table RFI_XREF_DVM, available in RFI schema in Retail).

EXT_SYSTEM_ID	COMMON_ID	RETL_ID
USA	700	US
CAN	701	CA

Financial Calendar

The financial calendar within Oracle Retail systems is manually set up and maintained separately from the Oracle General Ledger financial calendar.

Freight Terms

A freight term is an agreement between the retailer and a supplier regarding transportation charges for goods delivered by the supplier. Freight terms are used by RMS as purchase orders are generated.

Within the RMS system, freight terms are set up and maintained manually. They are also maintained in Oracle Payables.

Payment Terms and Currency Exchange Rates

Currency, exchange rates are created and updated in Oracle General Ledger and exported to RMS. Changes to Retail currency exchange rates are not propagated to Oracle General Ledger. Payment terms, however, are manually set up and maintained in each system.

Oracle E-Business Suite Financials Units and Site IDs

The data concepts of Org Units and Site IDs in RMS mirror the data maintained in Oracle E-Business Suite. RMS screens are used to manage and view Oracle Org Units and Site IDs. The RMS windows for Store and Warehouse allow for the association of each store and warehouse with an Org Unit.

Store and Warehouse Maintenance

The organizational unit is found in the Store and Warehouse screens, which allow the Oracle E-Business Suite operating unit to be associated with the Store or Warehouse.

- When RMS is set up for multi-channel operation, the organizational unit is set up at the virtual warehouse level.

Financial sales audit and inventory information is identified through interface routines and posted to the appropriate general ledger accounts. An organizational unit must be designated for each Store and Warehouse location in the RMS.

The following are examples of the Store and Warehouse screens:

Store Screen

Org Entity Type	Virtual Warehouse	Name	Virtual Warehouse Type	Channel	Channel Description	Pricing Location	Pricing Location Description	Transfer Entity	Transfer Entity Description	Finisher	Org Unit	Default Warehouse	Default Warehouse Name	Stock Holding	Customer Order Location
Regular Wareh...	1710	SS PWH - VW...	Non-traditional	1	Brick & Mortar...			111	Transfer Entity...	✓	1111			✓	---
Regular Wareh...	1711	SS PWH - VW...	CSC	2	E-Commerce			112	Transfer Entity...	✓	1112			✓	---
Regular Wareh...	1702	SS PWH 17 - ...	CSC	2	E-Commerce	4241	EIT Company...	112	Transfer Entity...	---	1112			✓	---
Regular Wareh...	1701	West Coast Wa...	CSC	1	Brick & Mortar...	4241	EIT Company...	111	Transfer Entity...	---	1111			✓	---
Regular Wareh...	34348	iam	CSC	1	Brick & Mortar...			112	Transfer Entity...	---	1112			✓	---
Regular Wareh...	12345	stafaf	CSC	4	E-Commerce	2233	EIT Franchise...	111	Transfer Entity...	---	1111			✓	---

Virtual Warehouse Screen

RMS General Ledger Setup

For RMS and ReSA, manual setup is required for validating the chart of accounts. Valid chart of accounts are created and stored in general ledger cross reference tables. Once the validation is completed, transaction data is assigned to specific account codes.

Ongoing maintenance of the chart of accounts information (such as adding, changing, or deleting chart of accounts) requires re-validation. In this regard, Oracle General Ledger is the system of record, as it is used to verify the chart of accounts used by Oracle Retail applications. When these applications send a chart of accounts for validation, Oracle General Ledgers issues a message with:

- Valid or invalid status
- Response date
- Chart of accounts

The RMS table `FIF_GL_SETUP`, stores the Oracle E-Business Suite Set of Books IDs to post financial information. This table must be setup manually after Set of Books IDs are determined. Where a system indicator Multiple Set of Books ID is set to N, `FIF_GL_SETUP` must hold a single Set of Books (SOB) record.

The Set of Books IDs is associated with the chart of accounts when setting up general ledger cross-reference.

RMS General Ledger Cross Reference

Select **RMS Tasks List > Foundation Data > Data Loading > Upload**. The Upload Data screen opens. Map Chart of Accounts to department, Class, Subclass, Set Of Books, location, and transaction codes by selecting the GL Cross Reference template to maintain the data using spreadsheet upload.

ReSA General Ledger Cross Reference

Select **ReSA task list>Foundation Data>Data Loading> Upload Data**. The Upload Data screen opens. Through this screen, the user can choose the GL Cross Reference template to maintain the data using spreadsheet upload.

When `SYSTEM_OPTIONS.FINANCIAL_AP` is A, the upload validates entries of valid segment combinations.

Upload Data

ReIM Data Setup and Configuration

This section describes setup considerations for ReIM data.

System Options

As part of the RMS system options setup script, set the following options as indicated:

- FINANCIAL_AP = A

Chart of Accounts Setup

The chart of accounts is set up manually in Oracle Retail applications and in Oracle General Ledger. All account combinations are set up in each Set Of Books. Account validation is done while executing Financial Posting batch.

Segment Mapping

The retailer determines how many segments are populated. Up to 20 account segments can be specified. The following is an example of how segments are mapped between the ReIM transaction table and Oracle General Ledger:

ReIM Segments	Oracle General Ledger Chart of Accounts
Segment 1	PRODUCT
Segment 2	ACCOUNT
Segment 3	ALTACCT
Segment 4	OPERATING_UNIT
Segment 5	FUND_CODE
Segment 6	DEPTID
Segment 7	PROGRAM_CODE
Segment 8	CLASS_FLD
Segment 9	BUDGET_REF

ReIM Segments	Oracle General Ledger Chart of Accounts
Segment 10	BUSINESS_UNIT_PC
Segment 11	PROJECT_ID
Segment 12	ACTIVITY_ID
Segment 13	RESOURCE_TYPE
Segment 14	RESOURCE_CATEGORY
Segment 15	RESOURCE_SUB_CAT
Segment 16	CHARTFIELD1
Segment 17	CHARTFIELD2
Segment 18	CHARTFIELD3
Segment 19	AFFILIATE
Segment 20	AFFILIATE_INTRA1

If any one of the values in the 20 segments does not match the Oracle General Ledger, the account combination is considered as invalid. The following error message is added to IM_POSTING_DOC_ERRORS table “One or more accounts subjected to posting is invalid”.

Segments 1 and 2 may be set up as dynamic at the Location level, or Segments 4 and 5 can be dynamic at the Department and Class level respectively. Segments defined as dynamic are allowed to be null for certain types of Basic Transaction or Reason Code cross-reference types. When a segment is null, the segment is assigned dynamically when transactions are posted. (Non-dynamic segments cannot be blank). Validation applies to the segment combination, not to individual segments.

Note: For Tran code TAP, each segment must have a value regardless of whether the segment is dynamic.

The initial load for ReIM is run by Oracle E-Business Suite and includes the following information:

- Suppliers
- Currency Rates

Note: The view, mv_currency_conversion_rates should be refreshed once the initial loads of currencies from Oracle General Ledger are loaded to ReIM.

IM_SYSTEM_OPTIONS table setup

To accommodate integration, the IM_SYSTEM_OPTIONS table should be configured with the following properties.

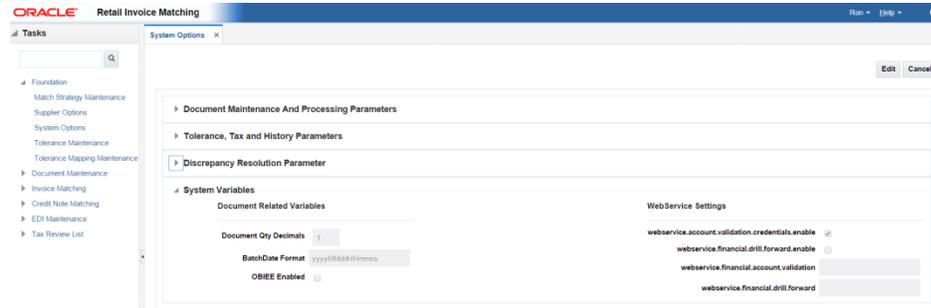
WS_FIN_ACC_VALID_URL - This attribute holds the URL for the financial account validation web service.

WS_FIN_ACC_VALID_CRED - This attribute indicates if the account validation web service call is authenticated or not. Y means Authentication enabled. N means Authentication is not enabled (See configuring web service credentials in weblogic server).

WS_FIN_DRILL_FWD_URL - This attribute holds the URL for the financial drill forward web service.

WS_FIN_DRILL_FWD_CRED - This attribute indicates if the drill forward web service call is authenticated or not. Y means Authentication enabled. N means Authentication is not enabled (See configuring web service credentials in weblogic server).

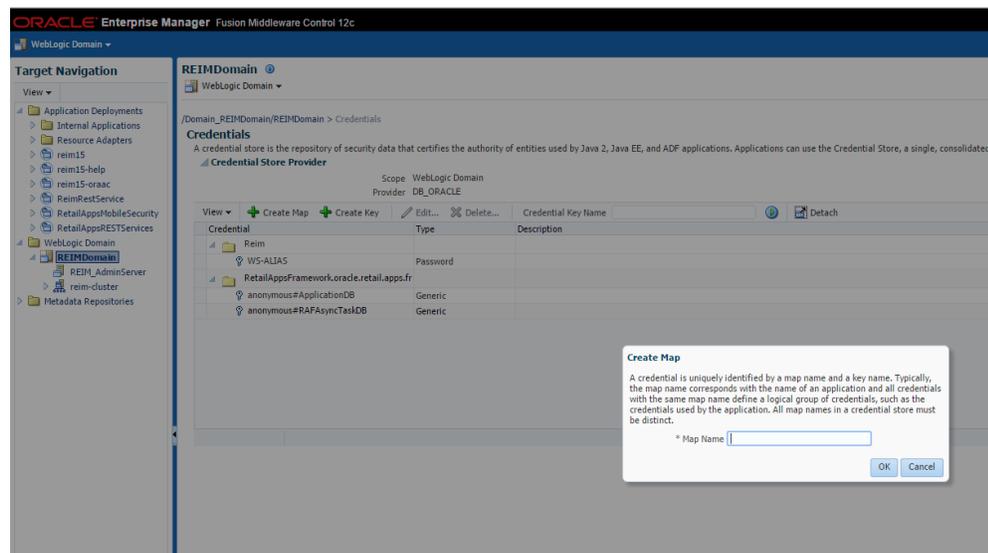
The above attributes are configured in the System Options screen.



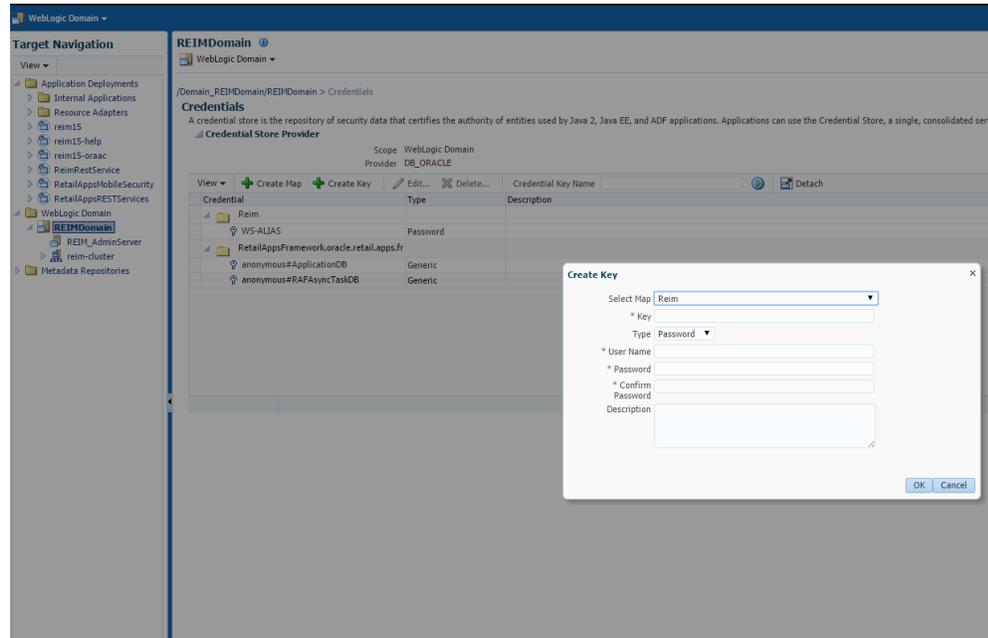
Configuring WebService Credentials in Weblogic Server Enterprise Manager:

The credentials for the web service call are configured in the default domain credential store of the weblogic server through the admin console.

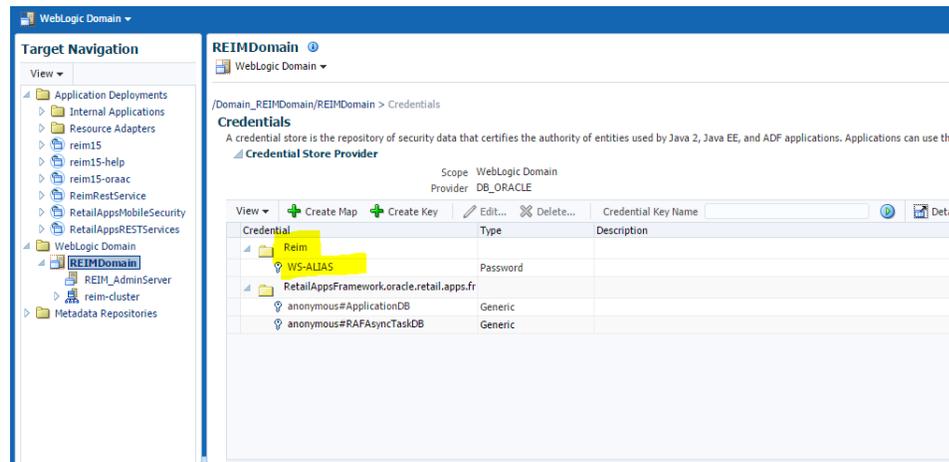
Create A Map:



Creating Credentials:



Note: Select the map as ReIM. The Key name should always be WS-ALIAS. Any user name / password can be assigned to the key



Reports are created by Business Intelligence Publisher for the following:

The URL for each report must be updated in the table, RETAIL_SERVICE_REPORT_URL. The following table provides sample URLs.

ReIM Transactional Maintenance

Integration to Oracle General Ledger includes a number of transactions, as described below.

Calculation of TRANS_AMOUNT

The TRANS_AMOUNT field in the IM_FINANCIAL_STAGE table stores the value of the journal entry to be posted to Oracle General Ledger.

(The currency for the calculated amount is the currency assigned to the transaction). The TRANS_AMOUNT value is calculated as follows:

Row Description	DEBIT_CREDIT_IND	TRANS_AMOUNT Value
Normal	Debit	Transaction Amount
Normal	Credit	(-1) * Transaction Amount
VAT	Debit	Transaction Amount * VAT Rate
VAT	Credit	(-1) * Transaction Amount * VAT Rate

Note: Transaction Amount is taken from the database column, IM_FINANCIALS_STAGE.AMOUNT.

Generation of Outgoing Data

A staging table accommodates the outgoing transfer of data. The reference key assigned to each document or receipt is used to find data on this table.

From	To	Transactions
ReIM	Oracle Payables	Invoices Debit Memos Credit Memos Credit Notes
ReIM	Oracle General Ledger	General Ledger accounting entries resulting from the Invoice Matching process, including: Pre-paid invoices Receipt Write-offs
RMS	Oracle General Ledger	Accounting entry data (potentially very high volume)
ReSA	Oracle General Ledger	Accounting entry data (potentially very high volume)

Validation of Accounts When Posting Financial Entries

Valid chart of accounts are stored in the ReIM table, IM_VALID_ACCOUNTS, which includes the Set of Books ID (sob_id) and 20 segments. An ORFI Web service validates accounts against the Oracle General Ledger. Valid accounts are posted to IM_VALID_ACCOUNTS; invalid accounts are posted to IM_POSTING_DOC_ERROR. The following steps describe the validation process:

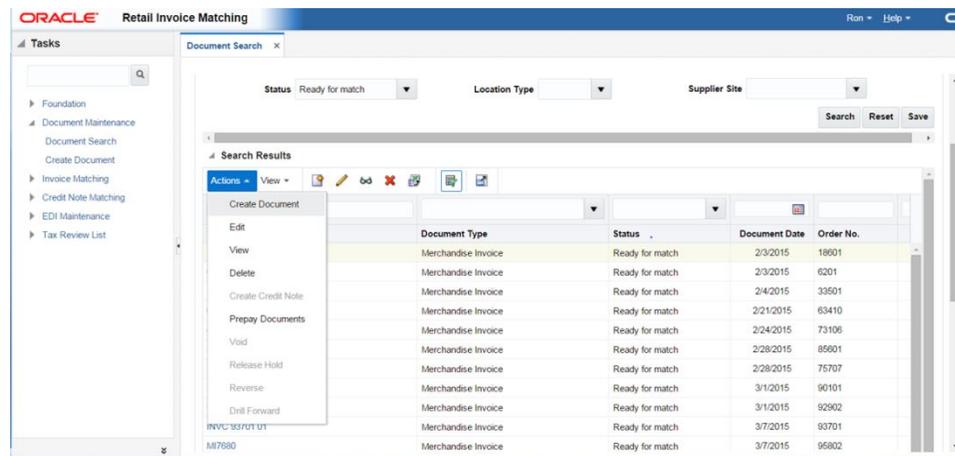
1. The ReIM system invokes the validation Web service to validate the chart of accounts. (A URL for the ORFI Web service is configured in the SYSTEM_OPTIONS table.)
2. The posting batch job checks the accounts to be posted against the IM_VALID_ACCOUNTS table.
3. If the chart of accounts is in the table, the transaction is posted to staging tables.
4. If the chart of account does not exist in the table, a collection of accounts is built. These collected accounts are validated against the Oracle General Ledger, and a status is returned.

- If the status of the collected accounts is valid, the accounts are inserted in the IM_VALID_ACCOUNTS table, and the transactions are posted to the staging tables.
- If the status of the accounts is NOT valid, the entire collection is flagged as errors, and transactions are posted to IM_POSTING_DOC_ERROR.

Note: ReIM completes the first level of account validation and posts the transaction to staging tables. It is assumed the second level of account validation is done at the end of the extraction process (where transactions are moved from ReIM staging tables to Oracle General Ledger). If account validation fails at this point, Oracle General Ledger must change the account information before transactions are loaded to Oracle General Ledger, and the chart of accounts must be re-validated in ReIM.

Validation of Accounts When Prepaying a Merchandise Invoice

A Merchandise document with a ready for match status can be pre paid before matching. The Document Search screen provides this option.



Document Type	Status	Document Date	Order No.
Merchandise Invoice	Ready for match	2/3/2015	18601
Merchandise Invoice	Ready for match	2/3/2015	6201
Merchandise Invoice	Ready for match	2/4/2015	33501
Merchandise Invoice	Ready for match	2/21/2015	63410
Merchandise Invoice	Ready for match	2/24/2015	73106
Merchandise Invoice	Ready for match	2/28/2015	85601
Merchandise Invoice	Ready for match	2/28/2015	75707
Merchandise Invoice	Ready for match	3/1/2015	90101
Merchandise Invoice	Ready for match	3/1/2015	92902
Merchandise Invoice	Ready for match	3/7/2015	93701
Merchandise Invoice	Ready for match	3/7/2015	95802

Valid chart of accounts are stored in the ReIM table, IM_VALID_ACCOUNTS, which includes the Set of Books ID (sob_id) and 20 segments. An ORFI Web service validates accounts against the Oracle General Ledger. Valid accounts are posted to IM_VALID_ACCOUNTS; invalid accounts are posted to IM_POSTING_DOC_ERROR. The following steps describe the validation process:

1. The ReIM system invokes the validation Web service to validate the chart of accounts. (A URL for the ORFI Web service is configured in the SYSTEM_OPTIONS table.)
 2. The Prepay operation checks the accounts to be posted against the IM_VALID_ACCOUNTS table.
 3. If the chart of accounts is in the table, the transaction is posted to staging tables.
- If the chart of account does not exist in the table, a collection of accounts is built. These collected accounts are validated against the Oracle General Ledger, and a status is returned.

- If the status of the collected accounts is valid, the accounts are inserted in the IM_VALID_ACCOUNTS table, and the transactions are posted to the staging tables.
- If the status of the accounts is NOT valid, the entire collection is flagged as errors, and transactions are posted to IM_POSTING_DOC_ERROR and the user will be shown an error message "One or more accounts subjected to posting is invalid".

Maintenance of Valid Accounts

As account information is changed in the Oracle General Ledger, Retail must re-validate the locally stored chart of accounts. Oracle General Ledger will not propagate chart of account changes to Retail. The AccountWorkspacePurge Batch can clear all valid accounts in the IM_VALID_ACCOUNTS table or only those that are considered updates in Oracle E-Business Suite.

Usage

```
AccountWorkspacePurge userid/password PURGE [ALL | <Accounts>]
```

Where:

- The argument is a combination of user ID and password.
- The argument is the word PURGE.
- The argument is either ALL or specific accounts to be deleted from the local table.

Integrating RMS with PeopleSoft Financials Using Oracle Retail Financial Integration

This chapter describes the integration between Oracle Retail systems and PeopleSoft Financials (including PeopleSoft General Ledger and PeopleSoft Payables), as developed and supported by Oracle Retail Financial Integration (ORFI).

When the option to integrate is chosen, the selected information is shared between the systems. Integration and validation services are in place to ensure the shared data matches.

Note: This chapter addresses the points within Oracle Retail systems that are essential to integration. For more information about the entire integration process, including mapping to data and settings, see the ORFI document, *Oracle Retail Financial Integration for Oracle Retail Merchandise Operations Management and Oracle Financials Implementation Guide*. For more information about Web services, see the following chapters in the *Oracle Retail Merchandising System Operations Guide, Volume 2: "Service Provider Implementations API Designs" and "Web Services"*.

Participating Applications

The following Oracle Retail applications are included in the integration covered by this chapter:

- Oracle Retail Merchandising System (RMS)
- Oracle Retail Sales Audit (ReSA)
- Oracle Retail Invoice Matching (ReIM)
- Oracle Retail Integration Bus (RIB)

Assumptions and Dependencies

- The option to integrate must be selected during initial set up of the RMS system.
- ReIM accesses RMS to determine if integration is active. The RMS set up must be done before integrating with REIM.
- The URLs for the RFI Web services that are necessary for this integration are maintained in the RETAIL_SERVICE_REPORT_URL table and in the ReIM SYSTEM_OPTIONS table.
- Real time account validation is done only when the financial integration with Oracle E-Business Suite is ON.
- Partners must be set up as suppliers in Oracle E-Business Suite and manually set up in RMS using the RMS Supplier ID. The RMS supplier ID generated when the Oracle E-Business Suite supplier is interfaced with Oracle Retail. The RMS supplier generated as part of this process is not used.
- Payment terms and freight terms are manually maintained.

Data Constraints

The Location ID field is restricted to eight characters, to accommodate PeopleSoft Operating Unit, which has a maximum of eight characters.

The Ext_Doc_ID field is restricted to 30 characters, because the corresponding PeopleSoft field has only 30 characters. Characters beyond 30 are truncated.

RMS allows for four decimals, and PeopleSoft allows only three. Truncation may occur when data is passed to PeopleSoft Enterprise Financials.

ReIM values in the IM_CURRENCY_LOCALE are restricted to three decimals, because the corresponding PeopleSoft Enterprise Financials field can accept no more than three decimal positions.

Data Setup

Integration of Oracle Retail applications and PeopleSoft Financials relies on synchronization of essential data, such as currency exchange rates, suppliers, and payment terms. Through careful discussions, the users of both systems determine the common codes and descriptions that will best serve their business needs.

Once agreement is reached, this information is set up and maintained. Depending on the volume, some shared information is set up in Oracle Retail applications or in PeopleSoft and electronically transferred to the other systems. Otherwise, shared information is set up manually within each system, and the user of both systems must ensure that the code and the description match.

RMS Data Setup and Configuration

This section describes setup considerations for the RMS data.

This SYSTEM_OPTION indicates that the Oracle Retail system is integrated with a financial system:

- FINANCIAL_AP=A

A value of A indicates that the financial system to which RMS is interfaced is PeopleSoft through Oracle Retail Financial Integration (ORFI).

- GL_ROLL_UP can be D/S/C
- SUPPLIER_SITE_IND = Y
- ORG_UNIT_IND = Y

Organization Units

Use the Upload Data screen (**RMS Tasks List > Foundation Data > Data Loading > Upload**) to define organizational units in RMS by uploading a spreadsheet of organizational unit information that matches those being setup in PeopleSoft. When an organizational unit is entered in RMS, the valid organizational units are those associated with the Set Of Books (SOB) that is being used for the general ledger interface.

Currency Exchange Rates

Currency exchange rate is used to translate the monetary value of one currency in terms of another. Depending on business needs, a Currency Exchange Rate Type of Operational or Consolidation is selected for use in all transactions.

This value is set up manually in RMS and mapped to PeopleSoft through the Currency Exchange Type mapping window. Currency Exchange Rate data is owned by PeopleSoft General Ledger and updates are sent to Oracle Retail applications.

Determine the Exchange Type being sent by PeopleSoft General Ledger (for example, Consolidation or Operational) that you want RMS to use. Update the FIF_CURRENCY_XREF for mapping the external exchange type being sent by PeopleSoft General Ledger with RMS Exchange Type.

For example, for Consolidation and Operational exchange types, the FIF_CURRENCY_XREF table holds the following entries:

FIF_EXCHANGE_TYPE	RMS_EXCHANGE_TY E
C	C
O	O

Supplier Address Types

Within RMS, supplier information (such as Order From and Remit To addresses) is used for generating the purchase orders. PeopleSoft Payables uses supplier information for payment generation. It is important that this information is synchronized.

Suppliers are created in PeopleSoft Payables and exported to RMS. When FINANCIAL_AP is set to A, suppliers cannot be created using the RMS screens. However, after the supplier exists in RMS, all data values for the supplier (except supplier name and status) continue to be updated using the RMS screens. The association of supplier sites to organization units is accessed only in view mode through RMS screens. One supplier site per supplier organization unit combination can be marked as primary payment site.

Where SYSTEM_OPTIONS.FINANCIAL_AP is A, disable auto generate supplier/partner numbers and associated check boxes.

Note: Supplier information is created, updated and inactivated only in PeopleSoft Payables. This information is transferred from PeopleSoft Payables to the participating Oracle Retail applications, where additional retail-specific attributes may be maintained.

Country Codes

When country codes are defined and seeded in RMS, ensure that country codes are mapped to PeopleSoft country codes through RFI DVM mapping. The following is an example of RFI DVM Mapping (Table RFI_XREF_DVM, available in RFI schema in Retail).

EXT_SYSTEM_ID	COMMON_ID	RETL_ID
USA	700	US
CAN	701	CA

Financial Calendar

The financial calendar within Oracle Retail systems is manually set up and maintained separately from the PeopleSoft General Ledger financial calendar.

Freight Terms

A freight term is an agreement between the retailer and a supplier regarding transportation charges for goods delivered by the supplier. Freight terms are used by RMS as purchase orders are generated.

Within the RMS system, freight terms are set up and maintained manually. They also are maintained in PeopleSoft Payables.

Payment Terms and Currency Exchange Rates

Currency exchange rates are created and updated in PeopleSoft General Ledger and exported to RMS. Changes to Retail currency exchange rates are not propagated to PeopleSoft General Ledger. Payment terms set up in PeopleSoft are propagated to RMS but changes to payment terms in RMS are not propagated back to PeopleSoft.

PeopleSoft Financials Units and Site IDs

The data concepts of Org Units and Site IDs in RMS mirror the data maintained in PeopleSoft. RMS screens are used to manage and view Oracle Org Units and Site IDs. The RMS windows for Store and Warehouse maintenance allows for the association of each store and warehouse with an Org Unit. The following is an example of the Organizational Unit form:

Store and Warehouse Maintenance

The organizational unit is found in the Store and Warehouse screens, which allow the PeopleSoft operating unit to be associated with the Store or Warehouse. When RMS is set up for single-channel operation, the organizational unit is set at the physical warehouse level. When RMS is set up for multi-channel operation, the organizational unit is set up at the virtual warehouse level. Financial sales audit and inventory information can then be identified through interface routines and posted to the appropriate general ledger accounts. An organizational unit must be designated for each Store and Warehouse location in the RMS.

The following are examples of the Store and Warehouse screens:

Store Screen

Org Entity Type	Virtual Warehouse	Name	Virtual Warehouse Type	Channel	Channel Description	Pricing Location	Pricing Location Description	Transfer Entity	Transfer Entity Description	Finisher	Org Unit	Default Warehouse	Default Warehouse Name	Stock Holding	Customer Order Location
Regular Wareh...	1710	SS PWH - VW...	Non-traditional	1	Brick & Mortar			111	Transfer Entity...	✓	1111			✓	---
Regular Wareh...	1711	SS PWH - VW...	CSC	2	E-Commerce			112	Transfer Entity...	✓	1112			✓	---
Regular Wareh...	1702	SS PWH 17 - ...	CSC	2	E-Commerce	4241	EIT Company...	112	Transfer Entity...	✓	1112			✓	---
Regular Wareh...	1701	West Coast Wa...	CSC	1	Brick & Mortar	4241	EIT Company...	111	Transfer Entity...	✓	1111			✓	---
Regular Wareh...	34348	ram	CSC	1	Brick & Mortar			112	Transfer Entity...	✓	1112			✓	---
Regular Wareh...	12345	stafdf	CSC	4	E-Commerce	2233	EIT Franchise...	111	Transfer Entity...	✓	1111			✓	---

Virtual Warehouse Screen

RMS General Ledger Setup

For RMS and ReSA, manual setup is required for validating the chart of accounts. Valid chart of accounts are created and stored in general ledger cross reference tables. Once the validation is completed, transaction data can be assigned to specific account codes.

Ongoing maintenance of the chart of accounts information (such as adding, changing, or deleting chart of accounts) requires re-validation. In this regard, PeopleSoft General Ledger is the system of record, as it is used to verify the chart of accounts used by Oracle Retail applications. When these applications send a chart of accounts for validation, PeopleSoft General Ledgers issues a message with:

- Valid or invalid status
- Response date
- Chart of accounts

The RMS table, FIF_GL_SETUP, stores the PeopleSoft Set of Books IDs to post financial information. This table must be setup manually after Set of Books IDs are determined. Where system indicator Multiple Set of Books ID is set to N, FIF_GL_SETUP must hold a single Set of Books (SOB) record.

The Set of Books IDs is associated with the chart of accounts when setting up general ledger cross reference.

RMS General Ledger Cross Reference

Select **RMS Tasks List > Foundation Data > Data Loading > Upload**. The Upload Data screen opens. Map Chart of Accounts to department, Class, Subclass, Set Of Books, location, and transaction codes by selecting the GL Cross Reference template to maintain the data using spreadsheet upload.

ReSA General Ledger Cross Reference

Select **ReSA task list>Foundation Data>Data Loading>Upload Data**. The manage data screen opens. Through this screen, the user can choose the General Ledger template to maintain the data using spreadsheet upload.

When SYSTEM_OPTIONS.FINANCIAL_AP is A, the upload validates entries of valid segment combinations.

Upload Data

Configuring Drill Back and Forward Web Services

Retail web services table, RETAIL_SERVICE_REPORT_URL, must be updated with appropriate URLs to integrate with PeopleSoft Enterprise Financials.

The records in the table for Services (indicated by RS_TYPE=S) for Account Validation (RAV) and Drill Forward (RDF), must be updated with the URL information from AIA where the services are hosted.

Note: If Web services are secure, then the SYS_ACCOUNT column must be populated with authentication information in the form of *user name/password*.

The records in the table for Reports (indicated by RS_TYPE=R) for both RMS and ReIM reports, must be updated with the URL information from the BIP Server where the reports are hosted.

ReIM Data Setup and Configuration

This section describes setup considerations for ReIM data.

System Options

As part of the RMS system options setup script, set the following options as indicated:

- FINANCIAL_AP =A

CURRENCY_PRECISION

Because PeopleSoft Enterprise Financials uses only three decimals, the transactions generated by the Oracle Retail ReIM application must not include more than three decimals.

Make sure that the RMS table CURRENCIES has maximum of 3 decimal places for the currency_cost_dec attribute for difference currency codes. The currency_rates table in RMS should be loaded initially by PeopleSoft Enterprise Financials.

System Options window

Chart of Accounts Setup

The chart of accounts is set up manually in Oracle Retail applications and in PeopleSoft General Ledger. All account combinations are set up in each Set Of Books. Account validation is done while executing Financial Posting batch.

Segment Mapping

The retailer determines how many segments are populated. Up to 20 account segments can be specified. The following is an example of how segments are mapped between the ReIM transaction table and PeopleSoft General Ledger:

ReIM Segments	PeopleSoft General Ledger Chart of Accounts
Segment 1	ACCOUNT
Segment 2	ALTACCT
Segment 3	DEPTID
Segment 4	OPERATING_UNIT
Segment 5	PRODUCT
Segment 6	FUND_CODE
Segment 7	CLASS_FLD
Segment 8	PROGRAM_CODE
Segment 9	BUDGET_REF
Segment 10	AFFILIATE
Segment 11	AFFILIATE_INTRA1
Segment 12	AFFILIATE_INTRA2
Segment 13	CHARTFIELD1
Segment 14	CHARTFIELD2
Segment 15	CHARTFIELD3
Segment 16	RESOURCE_TYPE
Segment 17	RESOURCE_CATEGORY
Segment 18	RESOURCE_SUB_CAT
Segment 19	BUSINESS_UNIT_PC
Segment 20	PROJECT_ID

If any one of the values in the 20 segments does not match the PeopleSoft General Ledger, the account combination is considered as invalid. The following error message is added to IM_POSTING_DOC_ERRORS table: "One or more accounts subjected to posting is invalid"

Segments 1 and 2 may be set up as dynamic at the Location level, or Segments 4 and 5 can be dynamic at the Department and Class level respectively. Segments defined as dynamic are allowed to be null for certain types of Basic Transaction or Reason Code cross-reference types. When a segment is null, the segment is assigned dynamically when transactions are posted. (Non-dynamic segments cannot be blank). Validation applies to the segment combination, not to individual segments.

Note: For Tran code TAP, each segment must have a value regardless of whether the segment is dynamic.

Running the Initial Load from PeopleSoft Financials

The initial load for ReIM is run by PeopleSoft and includes the following information:

- Suppliers
- Currency Rates
- Payment Terms

Note: The view, mv_currency_conversion_rates should be refreshed once the initial loads of currencies from PeopleSoft General Ledger are loaded to ReIM

IM_SYSTEM_OPTIONS table Setup

To accommodate integration, IM_SYSTEM_OPTIONS table should be configured to with the following properties.

Note: Drill forward functionality is applicable for PeopleSoft - RMS and ReIM integration. Drill backward functionality is only applicable for PeopleSoft - RMS integration.

Drill forward and drill backward functionality is applicable for Peoplesoft - RMS integration only.

WS_FIN_ACC_VALID_URL - This attribute holds the URL for the financial account validation web service.

WS_FIN_ACC_VALID_CRED - This attribute indicates if the account validation web service call is authenticated or not. Y means Authentication enabled. N means Authentication is not enabled (See configuring web service credentials in weblogic server).

WS_FIN_DRILL_FWD_URL - This attribute holds the URL for the financial drill forward web service.

WS_FIN_DRILL_FWD_CRED - This attribute indicates if the drill forward web service call is authenticated or not. Y means Authentication enabled. N means Authentication is not enabled (See configuring web service credentials in weblogic server).

The above attributes are configured in the System Options screen.

Reports are created by Business Intelligence Publisher for the following:

The URL for each report must be updated in the table, retail_service_report_url. The following table provides sample URLs.

Reporting

Reports are created by Business Intelligence Publisher for the following:

- Merchandise Invoice
- Non-Merchandise Invoice
- Credit Note
- Credit Memo
- Debit Memo
- Receipt Write-Off

The URL for each report must be updated in the table, `retail_service_report_url`. The following table provides sample URLs:

Document Type	Report Name	Sample Report URL
MRCCHI	Merchandise invoice document Report	http://hostname:portno/xmlpserver_nonsso/Guest/REIM13/Finance/invreport/invreport.xdo
NMRCCHI	Non-Merchandise invoice document Report	http://hostname:portno/xmlpserver_nonsso/Guest/REIM13/Finance/invreport/invreport.xdo
CRDNT	Credit Note document Report	http://hostname:portno/xmlpserver_nonsso/Guest/REIM13/Finance/invreport/crnreport.xdo
CRDMEC	Credit Memo cost document Report	http://hostname:portno/xmlpserver_nonsso/Guest/REIM13/Finance/invreport/memoreport.xdo
CRDMEQ	Credit Memo quantity document Report	http://hostname:portno/xmlpserver_nonsso/Guest/REIM13/Finance/invreport/memoreport.xdo
DEBMEC	Debit Memo cost document Report	http://hostname:portno/xmlpserver_nonsso/Guest/REIM13/Finance/invreport/imemoreport.xdo
DEBMEQ	Debit Memo quantity document Report	http://hostname:portno/xmlpserver_nonsso/Guest/REIM13/Finance/invreport/memoreport.xdo
DEBMEV	Debit Memo Tax document Report	http://hostname:portno/xmlpserver_nonsso/Guest/REIM13/Finance/invreport/memoreport.xdo
RWO	Receipt Write Off document Report	http://hostname:portno/xmlpserver_nonsso/Guest/REIM13/Finance/invreport/rworeport.xdo

ReIM Transactional Maintenance

Integration to PeopleSoft General Ledger includes a number of transactions, as described below.

Calculation of TRANS_AMOUNT

The TRANS_AMOUNT field in the im_financial_stage table stores the value of the journal entry to be posted to PeopleSoft General Ledger. (The currency for the calculated amount is the currency assigned to the transaction). The TRANS_AMOUNT value is calculated as follows:

Row Description	DEBIT_CREDIT_IN D	TRANS_AMOUNT Value
Normal	Debit	Transaction Amount
Normal	Credit	(-1) * Transaction Amount
VAT	Debit	Transaction Amount * Tax Rate
VAT	Credit	(-1) * Transaction Amount * Tax Rate

Note: Transaction Amount is taken from the database column, IM_FINANCIALS_STAGE.AMOUNT.

Generation of Outgoing Data

A staging table accommodates the outgoing transfer of data. The reference key assigned to each document or receipt is used to find data on this table.

Outgoing Data

From	To	Transactions
ReIM	PeopleSoft Accounts Payable	Invoices Debit Memos Credit Memos Credit Notes
ReIM	PeopleSoft General Ledger	General Ledger accounting entries resulting from the Invoice Matching process, including: Pre-paid invoices Receipt Write-offs
RMS	PeopleSoft General Ledger	Accounting entry data (potentially very high volume)
ReSA	PeopleSoft General Ledger	Accounting entry data (potentially very high volume)

Validation of Accounts When Posting Financial Entries

Valid accounts are stored in the ReIM table, `IM_VALID_ACCOUNTS`, which includes the Set of Books ID (`sob_id`) and 20 segments. An ORFI Web service validates accounts against the PeopleSoft Enterprise Financials system. Valid accounts are posted to `IM_VALID_ACCOUNTS`; invalid accounts are posted to `IM_POSTING_DOC_ERROR`. The following steps describe the validation process:

1. The ReIM system invokes the account validation Web service to validate the account. (A URL for the ORFI Web service is configured in the ReIM `SYSTEM_OPTIONS` table.
2. The posting batch job checks the accounts to be posted against the `IM_VALID_ACCOUNTS` table.
3. If the account entries are in the table, the transaction is posted to the G/L or AP tables.
4. If the account does not exist in the table, a collection of accounts is built. These collected accounts are validated against the PeopleSoft Enterprise Financials system, and a status is returned.
 - If the status of the collected accounts is valid, the accounts are inserted in the `IM_VALID_ACCOUNTS` table, and the transactions are posted to the staging tables.
 - If the status of the accounts is NOT valid, the entire collection is flagged as errors, and transactions are posted to `IM_POSTING_DOC_ERROR`.

Note: ReIM completes the first level of account validation and posts the transaction to staging tables. It is assumed the second level of account validation is done at the end of the extraction process (where transactions are moved from ReIM staging tables to PeopleSoft). If account validation fails at this point, Oracle Data Integrator (ODI) or PeopleSoft must change the account information before transactions are loaded to PeopleSoft, and the account change must be communicated to ReIM.

Maintenance of Valid Accounts

As account information is changed in the PeopleSoft system, the same changes are communicated to, and manually completed in, the ReIM system. After ReIM is updated accordingly, the `AccountWorkspacePurge` Batch is run to clear the valid accounts maintained locally in ReIM.

The `AccountWorkspacePurge` Batch can clear all valid accounts in the `IM_VALID_ACCOUNTS` table or only those that are considered updates in PeopleSoft.

Usage:

```
AccountPurge batch-alias-name PURGE [ALL | <Accounts>]
```

Where:

The first argument is batch alias name.

The second argument is the word `PURGE`.

The third argument is either `ALL` or specific accounts to be deleted from the local table.

Building and Posting Reference IDs

Drill back and drill forward functionality uses Reference ID to locate documents and receipts. A Reference ID is a combination of document type and document (or receipt) ID, as illustrated in the table below:

Type	Doc ID	Receipt ID	Reference ID
Merchandise Invoice	101	Null	MRCHI#101
Non-Merchandise Invoice	102	Null	NMRCHI#102
Receipt	Null	103	RECEIPT#103

For documents, the Resolution Posting Batch program builds the Reference ID using the standard, Document Type + DeDelimiter + Doc_id. For receipts, the program builds the Reference ID using the standard, Document Type + DeDelimiter + Receipt_id.

To enable drill down functionality, Reference IDs are loaded to staging tables. FinancialsAPStageDao and FinancialsGLStageDao are populated, as are IM_RWO_SHIPMENT_HIST and IM_RWO_SHIPSKU_HIST.

Drilling Back

Drilling back allows users to view the source of posted PeopleSoft transactions that originated in Oracle Retail systems (from a voucher to an invoice, for example).

When drilling back from PeopleSoft Enterprise Financials, users are not directed to an actual screen within RMS, ReIM or ReSA. Rather, a retail Web service generates and launches a URL to a BI Publisher report. The report contains the information that typically appears on the appropriate retail screen.

Depending on the information requested by the user, PeopleSoft invokes web service in the ORFI layer which in turn calls the report locator service available in RMS. This service returns back BIP report reference URL which is passed back to PeopleSoft and report gets launched in browser window.

Information from the reference key determines what kind of report URL to issue. For example, if the retail key has a prefix of ReIM, the ReIM_REPORT_URL function is called, else the RMS_REPORT_URL function is called to retrieve the appropriate report URL. If the key does not match any key in the retail systems, an error message is launched.

Drilling Back to RMS and ReSA from PeopleSoft Enterprise Financials

The following function determines which RMS report to return to the user:

```
RMS_REPORT_URL() -
  O_error_message    IN OUT    RTK_ERRORS.RTK_TEXT%TYPE
  O_rpt_url          IN OUT    RETAIL_SERVICE_REPORT_URL.URL%TYPE
  I_ref_key          IN        KEY_MAP_GL.REFERENCE_TRACE_ID%TYPE
```

The appropriate report URL is found and issued as follows:

1. The ref_trace_type is found on KEY_MAP_GL by matching I_ref_key with the KEY_MAP_GL.REFERENCE_TRACE_ID column.
2. When ref type is determined, the re_trace_type is used to find the appropriate report URL on the RETAIL_SERVICE_REPORT_URL table.
3. The value of I_ref_key is appended to the end of the URL retrieved from the table.

4. The URL is sent back to the calling function.
5. If I_ref_key does not exist on KEY_MAP_GL, an error message is sent back to the calling function.

Drilling Back to ReIM from PeopleSoft Enterprise Financials

The following drill back options are available for viewing information within the ReIM system:

Using Document ID, users can drill back to ReIM to view information related to a voucher or payment. The report includes information from im_doc_head and im_invoice_detail, the same data shown on the Document Maintenance Header screen within ReIM.

Using the Receipt ID, users can drill back to view information from the Receipt Write-off History screen. Receipt write-offs occur either when an open receipt is closed in ReIM or if a receipt is purged in RMS before it is fully matched. Details come from the IM_RWO_SHIPMENT_HIST and IM_RWO_SHIPSKU_HIST tables.

The function below determines which of the two ReIM reports to return to the user:

```

REIM_REPORT_URL() -
  O_error_message    IN OUT    RTK_ERRORS.RTK_TEXT%TYPE
  O_rpt_url          IN OUT    RETAIL_SERVICE_REPORT_URL.URL%TYPE
  I_ref_key          IN        KEY_MAP_GL.REFERENCE_TRACE_ID%TYPE
    
```

The I_ref_key contains the reference ID, which ultimately determines the type of report required. The appropriate BI Publisher report URL is found on the RETAIL_SERVICE_REPORT_URL table.

In general, if the reference ID has a prefix of RECEIPT, the report type (RS_CODE) is RCPT. Otherwise, the report type is DOC. For example:

Reference ID	Report Type (RS_CODE)
MRCHI#101	DOC
NMRCHI#102	DOC
RECEIPT#103	RCPT

The following is an example of a BI Publisher URL that is generated upon drilling back to PeopleSoft Enterprise Financials for information on an invoice in ReIM, using Document ID as the search parameter:

```

http://mspdev6970vip:7777/BIPublisher/Guest/ReIM/13.0.3/doc/tsf_det.xdo
?doc_id=101
    
```

Where

http://mspdev6970vip:7777/BIPublisher = the BI Publisher application server address and port

Guest/ReIM/13.0.3 = the directory/folder location

doc/tsf_det.xdo ? = report name (Document Report)

doc_id=101 = the parameter name and value (Document ID 101)

The following is an example of an Oracle Business Intelligence Publisher URL that is generated upon drilling back to PeopleSoft Enterprise Financials for information on an invoice in ReIM, using Receipt ID as the search parameter:

```
http://mspdev6970vip:7777/BIPublisher/Guest/ReIM/13.0.3/doc/tsf_det.xdo ?receipt_id=101
```

Where

http://mspdev6970vip:7777/BIPublisher = the BI Publisher application server address and port

Guest/ReIM/13.0.3 = the directory/folder location

doc/tsf_det.xdo ? = report name (Receipt Report)

receipt_id=101 = the parameter name and value (Receipt ID 101)

Drilling Forward

Drilling forward allows users to see detailed information about retail transactions that have been posted to PeopleSoft Enterprise Financials. When drilling forward, users are directed to selected "view-only" screens.

Drilling Forward From RMS/ReSA to PeopleSoft Enterprise Financials

The following forms may be used to drill forward from RMS/ReSA:

- RMS StartMenu->Finance->Transaction Data View (trandata.fmb)
- RMS StartMenu->Ordering->Fixed Deals->Fixed Deal Transaction Data View (fdltrandata.fmb)
- Resa Task List->Operations->General Level Transaction View

Drilling Forward From ReIM to PeopleSoft Enterprise Financials

For drilling forward, the ORFI Web service uses the Invoice ID and Accounting Entry parameters. The ReIM system uses these parameters together as the Reference ID.

From the Document search screen, users can drill forward to PeopleSoft Enterprise Financials accounts payable to view voucher and payment status. The information is displayed on a read-only Payment Doc Status Inquiry screen. Drill forward access to the accounts payable system is available only for pre-paid invoices (but not for manually pre-paid invoices).

To drill down to the payables/ledger screens, the user invokes the Web service as follows:

Invoice ID/Accounting Entry parameter = Reference ID

Drill forward access to the G/L system is available only for pre-paid and manually pre-paid invoices.

Note: For more information on drilling forward, see the *Oracle Merchandising System or PeopleSoft Financials*.

Understanding Data Access Schema

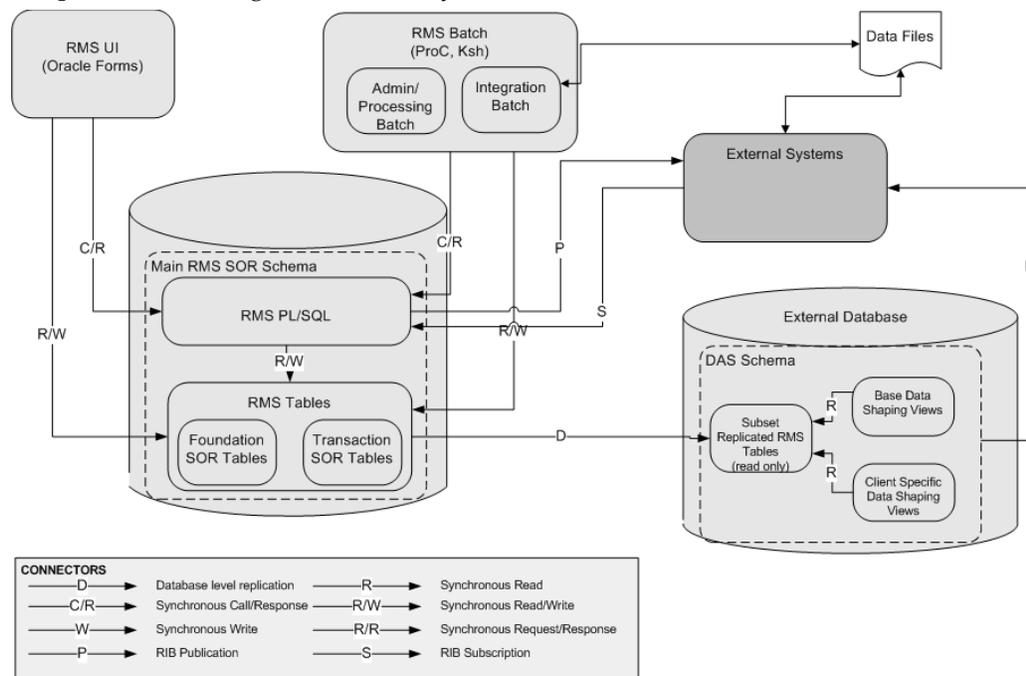
The Data Access Schema (DAS) is a new schema that allows third party applications to access RMS owned data. A subset of the data mastered by RMS is replicated to a new database schema. This replicated data is in a READ ONLY format. The DAS is an optional function; where the base RMS does not require DAS. However, you may want to install DAS for the following reasons:

- Accelerate integration with some of the third party systems.
- Insulate core RMS processes from the resource demands for the third party systems.
- Provide downstream applications with data 24x7, instead of a predetermined time for the batch cycle that extracts RMS data.
- Perform custom operational reporting without impacting the core RMS operations.
- Use out of box integration with Merchandise Financial Planning's (MFP) PoViewer micro application.

Views are layered on top of the replicated data to shape it in a less RMS specific terms. Oracle recommends that DAS be deployed in a separate database instance on a separate hardware. This insulates the operations and performance of RMS from all demands made of DAS by other systems.

For more information on DAS, see the *Oracle Retail Merchandising System Data Access Schema Developer's Guide* and the *Oracle Retail Merchandising System Data Access Schema Data Model*.

The process flow diagram of a DAS system:

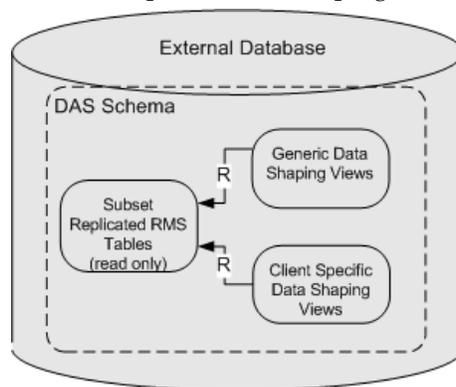


DAS Process Flow

DAS Views

DAS contains replicated tables and parallel layers of views. The two parallel layers of views are:

- Generic Data Shaping View
- Client Specific Data Shaping View



DAS Views

The Generic Data Shaping View makes RMS owned data more understandable to external systems by both flattening and exploding some of the concepts in RMS.

The client specific data shaping views presents data for specific downstream client applications. You can add additional client specific data shaping views to execute queries that are important to manage the DAS integrations with the third party systems.

You are encouraged to use the DAS for downstream, custom integration batch, and query service integrations. The following are the advantages of using DAS instead of the core RMS system of record tables:

- DAS insulates core RMS processing from the resource demands of third party systems.
- DAS allows constant access to data that of a RMS batch cycle which is processed once in a day, restricting the access of data or a real time query that is slower when processing demands of high point of RMS batch cycles.
- Custom client specific data shaping views can accelerate the development of custom integrations.

Using Oracle Wallet

RMS Batch programs runs using wallet alias as the first parameter to the batch command line arguments. This is enabled to prevent the security concerns around exposing database user ID and password while running the batch programs.

The wallet creation steps are described in the RMS Installation Guide. The wallet and wallet alias creation are a required in order to use batch programs in secured mode.

If we assume wallet alias is "dvols29_rms01batch";

Usage:

```
./dtesys $UP
```

(where UP will be set during installation to the wallet alias, that is:

```
$UP=@dvols29_rms01batch)
```

or

```
./dtesys @dvols29_rms01batch (wallet alias)
```

RMS/RPAS and RMS/AIP RETL scripts uses Oracle Wallet. Defined Oracle Wallet is referenced in `rmse_config.env`, `rmse_aip_config.env` and `rmse_rpas_config.env` files.