Oracle® Retail Merchandising

Implementation Guide

Release 15.0

E65391-01

December 2015

Oracle® Retail Merchandising Implementation Guide, Release 15.0

E65391-01

Primary Author: Rakhee Prabhudesai

Value-Added Reseller (VAR) Language

Oracle Retail VAR Applications

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

4 Oracle Retail Merchandising System

5 Oracle Retail Trade Management

6 Oracle Retail Sales Audit

7 Oracle Retail Operational Insights

8 Oracle Retail Allocation

11 Oracle Retail Active Retail Intelligence

Send Us Your Comments

Oracle® Retail Merchandising Implementation Guide, Release 15.0

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

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

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

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

> **Note:** Before sending us your comments, you might like to check that you have the latest version of the document and if any concerns are already addressed. To do this, access the Online Documentation available on the Oracle Technology Network Web site. It contains the most current Documentation Library plus all documents revised or released recently.

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

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

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

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

Preface

This Oracle Retail Merchandising Implementation Guide provides a high-level view of how the Merchandising Operations Management applications integrate with each other. The guide includes the following:

- Installation overview

  This chapter is a high-level overview of the process required for a successful installation of the Oracle Retail Merchandising Operations Management applications. In addition, this chapter provides hardware and software requirements and the order in which the applications must be installed. Note that this does not replace existing installation guides, but it does provide a single reference to requirements and what to expect when installing the Merchandising Operations Management applications.

- Chapters on Oracle Retail applications

  These chapters provide an overview of the Merchandising Operations Management applications. They provide a detailed look at how each application integrates with the other Merchandising Operations Management applications, as well as the information passed back and forth between those applications (that is, the information RMS (Retail Merchandising System) provides to other applications and the information other applications provide to RMS). These chapters also describe the users that must be created for each application as well as the security settings that these applications can employ.

Audience

The implementation guide is intended for the Oracle Retail Merchandising Operations Management applications integrators and implementation staff, as well as the retailer's IT personnel.

Documentation Accessibility

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

Access to Oracle Support

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

Related Documents

For more information, see the following documents:

- *Oracle Retail Merchandising Batch Schedule*
- *Oracle Retail Merchandising Security Guide*
- *Oracle Retail Xstore Suite 15.0/Merchandising 15.0 Implementation Guide*
- *Oracle Retail POS Suite 14.1/Merchandising Operations Management 15.0 Implementation Guide*
- Oracle Retail Allocation documentation
- Oracle Retail Invoice Matching documentation
- Oracle Retail Merchandising System documentation
- Oracle Retail Trade Management documentation
- Oracle Retail Sales Audit documentation
- Oracle Retail Price Management documentation
- Oracle Retail Active Retail Intelligence documentation

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, 15.0) or a later patch release (for example, 15.0.1). If you are installing the base release, additional patch, and bundled hot fix releases, read the documentation for all releases that have occurred since the base release before you begin installation. Documentation for patch and bundled hot fix 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/technology/documentation/oracle-retail-100266.html`

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

Conventions

The following text conventions are used in this document:

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

1 Installation Overview

This chapter is a high-level overview of the process required for a successful installation of the Oracle Retail Merchandising Operations Management applications. For complete step-by-step installation process, see the Installation Guide for each application.

Pre-installation

Pre-installation requirements are tasks that must be researched and completed prior to starting the actual installation process.

Hardware requirements vary between applications. See Installation Guide for each of your applications for specific hardware requirements. The amount, capabilities, and types of hardware required for each implementation will vary based on the retailer's needs and objectives.

Software requirements vary among applications. See Installation Guide for each of your applications for specific software requirements.

Database Partitioning Strategy

Establishing a database partitioning strategy is a mandatory step that must be completed prior to installing any of the Oracle Retail Merchandising Operations Management applications. A spreadsheet is included with the installation materials that define requirements for both mandatory and optional partitioning. This enables the resource in charge of implementing the partitioning strategy to determine the correct strategy.

Installation Order

See the Installation Guides for the individual Merchandising products for the required order of installation.

Installation Processes

Some of the Oracle Retail Merchandising Operations Management applications provide installers to facilitate the installation process, while others require a manual installation. The following are the processes used to install each application.

Application	Installation Process
Allocation	Installer

Application	Installation Process
ARI	Manual installation
ReIM	Installer
RMS	Installer
RPM	Installer
ReSA	Installer

Product Compatibility

The *Oracle Retail Merchandising Implementation Guide* provides a high-level view of the interaction between Oracle Retail Merchandising Operations Management applications. However, some Oracle Retail Merchandising Operations Management applications can be integrated with other Oracle products. These additional integrations details are not given in this guide.

Oracle E-Business Suite and PeopleSoft

RMS, Oracle Retail Sales Audit (ReSA), and Oracle Retail Invoice Matching (ReIM) can be integrated with Oracle E-Business Suite or PeopleSoft. This integration allows RMS to obtain the following data from Oracle E-Business Suite:

- General ledger accounting data
- Supplier and supplier site data
- Currency rates
- Freight terms
- Payment terms

For more information, see the Oracle Retail Merchandising System Operations Guide Volume 3 - Back-End Configuration and Operations Guide.

Oracle Retail POS Suite

RMS, ReSA, and Oracle Retail Price Management (RPM) can be integrated with the following Oracle Retail POS Suite applications:

- Oracle Retail Point-of-Service
- Oracle Retail Back Office
- Oracle Retail Central Office

For information on the integration, see the *Oracle Retail POS Suite 14.1/Merchandising Operations Management Implementation 15.0 Guide* and the *Oracle Retail Xstore Suite 15.0/Merchandising 15.0 Implementation Guide.*

Oracle Retail Advanced Inventory Planning

Oracle Retail Merchandising System (RMS) can be integrated with Oracle Retail Advanced Inventory Planning (AIP). For more information, see the AIP and RMS documentation.

Oracle Retail Predictive Application Server

RMS can be integrated with Oracle Retail Predictive Application Server (RPAS). See RMS and RPAS documentation for more information.

2

Merchandising Operations Management Applications

This chapter briefly describes each of the Merchandising Operations Management applications. See the chapters that follow for more detailed descriptions.

Oracle Retail Merchandising System

RMS is the foundation system that records and controls virtually all data in the retail enterprise and ensures data integrity across all integrated systems. RMS includes key functions such as item maintenance, inventory management, and replenishment. This functionality provides easy access to the information that is crucial to the day-to-day merchandising activities within a retail organization, providing the ability to focus on key decisions that help achieve sales and profit targets.

RMS streamlines business practices and unifies business systems across retail channels to better serve customers. Because RMS has been developed as a Web-based, scalable product, it fully supports the large volumes found in retail, leaving more time for retailers to concentrate on the bottom line.

Oracle Retail Trade Management

Oracle Retail Trade Management (RTM) is an import management system designed to integrate and streamline the international trade transaction process. RTM links multiple departments together for all import functions. RTM provides immediate online visibility to the status, location, and disposition of products as they move through the import cycle.

RTM links partners in the supply chain-suppliers, agents, bank, transportation providers, freight consolidators, customs brokers-to share a constant flow of information needed to manage the movement of goods from the source to destination across international borders.

Because RTM is coupled with RMS, the import purchase order process also ties in with regular purchase order features such as open-to-buy, updating stock ledger, and inventory. RTM provides the facility to track and capture expenses incurred in the import process, and to apportion the expenses to the actual landed cost of the inventory. The application also provides letter of credit payment processing, which is typically used in import purchase orders.

Oracle Retail Sales Audit

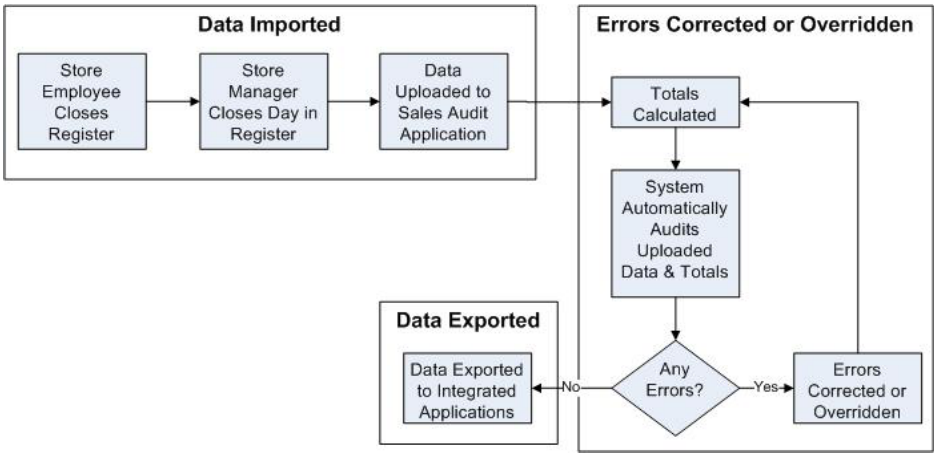
ReSA is an auditing system that provides a simplified sales audit process that accepts raw point of sale data and provides clean data to downstream applications, while ensuring integrity of audited data. The application supports automatic and interactive auditing of the Sales Data. The application is designed to focus on the exception conditions, while allowing clean data to flow through thus increasing productivity.

Flexibility is provided in the creation of user-defined rules and totals to configure exceptional conditions. User-defined audit rules fine-tune the system to focus validation on potential problem areas, and custom totals are created online for validation of calculations such as data entry or over/short.

Interactive audit functionality allows auditors to focus on exceptions and the audit to navigate through resolution to ensure a clean data load to the integrated applications. The application validates and balances the transactions and tender data, and detects and corrects the errors according to the predefined rules. The application allows sales balancing at store/register or cashier levels. The application helps identify, review, and resolve errors and irregularities in a timely manner.

The following diagram illustrates the audit process.

Figure 2–1 Information Maintained by Oracle Retail Sales Audit


Data Imported
Store Employee Closes Register
Store Manager Closes Day in Register
Data Uploaded to Sales Audit Application
Errors Corrected or Overridden
Totals Calculated
System Automatically Audits Uploaded Data & Totals
Any Errors?
Yes
Errors Corrected or Overridden
No
Data Exported
Data Exported to Integrated Applications

Operational Insights Dashboard and ReSA

Operational Insights offers a rich operational business intelligence solution to retail industry users. Operational Insights dashboards and reports are built on top of the latest Oracle technology stack and utilizes Oracle Business Intelligence Enterprise Edition (OBIEE) for reporting and analysis needs.The Merchandising suite of products is a proven, integrated solution with a rich data set. OBIEE reports utilize this data to provide insights into the retailer's operations

Operational Insights have the following salient features:

- Rich underlying data as the base for BI insights
- Proven, integrated solution that supports best practices
- A modern user interface using Oracle BIEE
- Actionable BI which enables the user to act on information gathered from reports without leaving the application or having to record the action in another interface

- Pervasive BI to reduce lag or latency between events and the subsequent action hence driving the workforce efficiency in business

From 14.1 release Operational Insights provides integration of OBIEE reports into the Oracle Retail Sales Audit (ReSA) user interface. ReSA provides the tools to evaluate point-of-sale data, to ensure the accuracy and completeness of information exported to downstream systems used in optimization processes, financial reporting, and analysis. Oracle Retail Sales Audit (ReSA) Operational Insights dashboards and reports provide pervasive business intelligence and are designed to be embedded within the ReSA ADF application.

Oracle Retail Allocation

A retailer's most important asset is its inventory. Oracle Retail Allocation helps retailers determine the inventory requirements at the item, store/warehouse, and week level using real time inventory information. The system calculates individual store or warehouse need based on parameters you set - whether it's the characteristics of the product, the location, or the category. The result is an allocation tailored to each destination location’s unique need.

Oracle Retail Allocation allows you to allocate either in advance of the order's arrival or at the last minute to leverage real-time sales and inventory information. And when you do allocate, the system provides you the flexibility of basing your allocation on many different methods such as: merchandise sales plans, receipt plans, forecast demand and sales history.

Oracle Retail Invoice Matching

Invoice matching is a control procedure designed to ensure that the retailer pays the negotiated cost for actual quantities received.

ReIM supports the invoice verification process with accuracy and efficiency, focusing resources on exception management. ReIM accepts electronic invoice data uploads through Electronic Data Interchange (EDI) and provides rapid online summary entry of invoices. ReIM supports automated and online processes that allow one or more invoices to be matched against one or more receipts. When the cost and quantities of an invoice are matched within tolerance, the invoice is ready for payment and staged to a table to be extracted to the accounts payable application.

If a cost or quantity difference between the invoice and receipts is outside tolerance, a discrepancy is recognized and must be resolved. A flexible resolution process allows discrepancies to be directed to the most appropriate user group for disposition. Reviewers can assign one or more reason codes that they are authorized to use to resolve the discrepancy.

Each reason code is associated with a type of action (for example, create chargeback or receiver cost adjustment). Many reason codes can be associated with a particular action type, allowing more granular reporting. Actions drive document creation and EDI downloads to suppliers, inventory adjustments, and accounting activities. Actions also allow an invoice to be extracted by a retailer and posted for payment.

Oracle Retail Price Management

Oracle Retail Price Management (RPM) is a pricing and promotions execution system. RPM functionality includes the definition, maintenance, and review of price changes, clearances, and promotions. RPM capabilities range from simple item price changes at a single location to complex buy-get promotions across zones.

RPM has three primary pricing execution dialogs to create and maintain regular price changes, clearances, and promotions. Although each of the three pricing activities is unique, the system displays these dialogs using a common look and feel. Each of these dialogs uses the conflict checking engine that leverages the RPM future retail tables. The future retail tables provide a forward-looking view of all pending approved pricing events that affect an item at a given location.

RPM pricing events are defined against the zone structure. The zone structure represents a group of locations organized to support a retailer's pricing strategy. RPM allows the user to break out of the zone structure and create location-level events as needed.

RPM supports the definition and application of price guides to these pricing events. Price guides allow the retailer to smooth retails and provide ends-in rounding logic to derive a final consumer price.

RPM supports pricing strategies you can use to define how item retails are proposed when pricing worksheets are generated. The strategies are defined at department, class, or subclass to represent which items are affected.

RPM also supports creation of vendor-funded promotions, by either associating an existing deal from RMS with the promotion, or by creating a new deal in RMS based on the information provided for the promotion.

Oracle Retail Active Retail Intelligence

Oracle Retail Active Retail Intelligence (ARI) is an exception management and workflow tool driven by business rules. It spans the Oracle Retail Merchandising Operations Management applications and provides a central repository for alert notifications and associated actions across all Oracle-based applications.

ARI can be uniquely configured for each client to fit individual business needs.

ARI provides the capability to build three categories of rules:

- Exception reporting
- Workflow management
- Enterprise process modification

3

Oracle Retail Application Administration Console

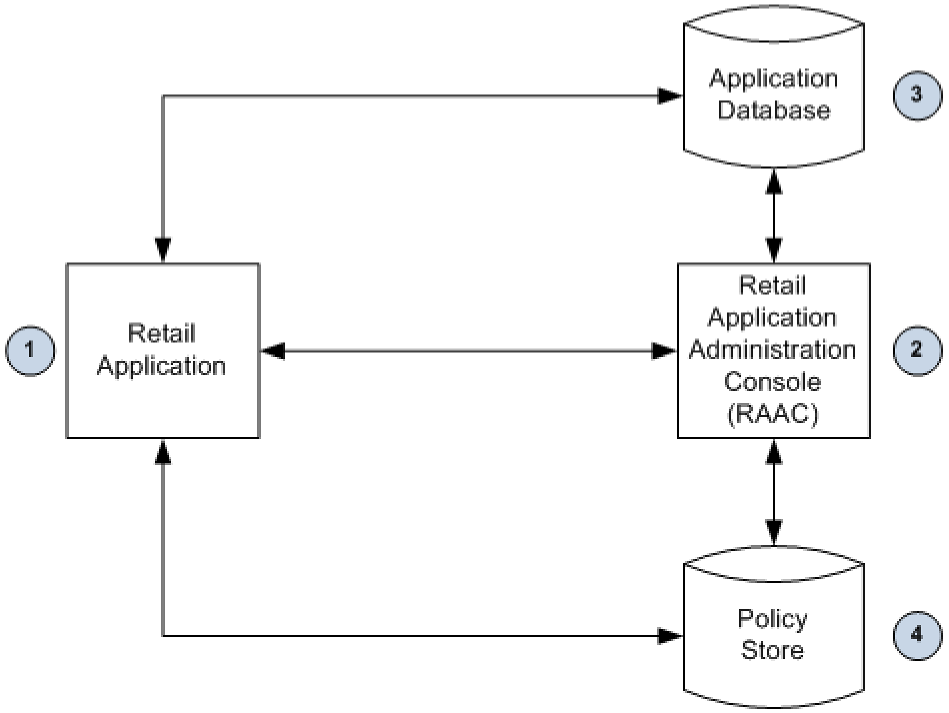
Oracle Retail Application Administration Console Overview

Oracle Retail Application Administration Console (RAAC) is an administrative tool that is used by Oracle Retail applications built using the Retail Application Framework (RAF). It allows an administrator to manage application roles, manage application navigator and manage notifications.

Technical Overview

The following diagram and descriptions present a high-level technical overview of RAAC.

Figure 3–1 RAAC Technical Overview Diagram



1. The RAF based applications have been designed to use RAAC. The Retail applications' installer installs RAAC as an application that can be invoked by a link in the host applications' user menu.
2. Retail Application Administration Console (RAAC) – This is the application that allows administrators to manage application roles, application navigator, and notifications.
3. Application Database – Application Navigator management and Notifications management persist settings in the database.
4. Policy Store – The repository for policies comprising one or more application stripes and code-based grants. Since the OPSS Management API is used to read from the policy store, the fact that the Policy Store may be file-based or database based etc, is irrelevant to RAAC. The API abstracts that deployment choice from RAAC.

Integration

RAAC is accessed from the drop down menu displayed when clicking on the logged-in administrator user ID located on the Retail application's global menu (see Figure 3–2, "RAAC User Menu"). The link launches RAAC. The Roles Mapping page in RAAC displays only the launching application's roles. The administrator can then perform roles mapping for the application.

Navigation

RAAC is enabled within an application. Users can access this application from the user menu option in the Global area of the main application page and is available only for users with the following job roles:

Table 3–1 RAAC Security Privileges

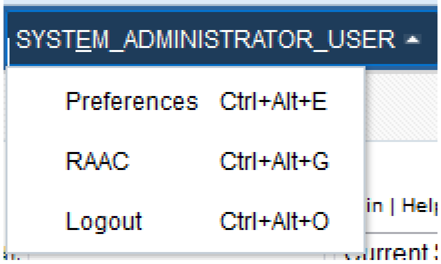
Role	Duty	Privilege
ReIM Application Administrator	Admin Console Duty	Application Navigator Privilege
		Role Manager Privilege
		Notification Manage Privilege
Allocation Application Administrator	Admin Console Duty	Application Navigator Privilege
		Role Manager Privilege
		Notification Manage Privilege
ReSA Application Administrator	Admin Console Duty	Role Manager Privilege
		Application Navigator Privilege
		Notification Manage Privilege

Accessing RAAC

RAAC is accessed through an RAAC supported application. Attempting to access RAAC directly through the URL will result in an error.

1. Click the RAAC application link in the user menu.

Figure 3–2 RAAC User Menu



   A new tab window or browser window is launched (depending on your browser settings).

2. From the Retail applications' login screen (for non-single sign-on), enter your username and password.

Manage Roles

Most retailers have their security systems and groups defined and available for them to use. As part of the Oracle Retail application security set up, default enterprise roles/groups and their mappings to application roles are provided with every application. These enterprise roles/groups may not map one-to-one to retailers' security groups. Role Manager in RAAC provides a way for retailers to modify the default enterprise roles so they map to their security groups. It also provides a way for the retailers to change the mappings of the enterprise roles/groups to the application roles.

Roles Explained

The following is an explanation of Enterprise (or Job) Roles and Application (or Duty and Privilege) roles:

Job Roles

Job roles are called as such because they closely map to the jobs commonly found in most retailer organizations.

Naming Convention: All retail Job role names end with' _JOB'.

Example: ALLOCATOR_JOB.

Duty Roles

Duties are tasks or responsibilities one must do on 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.

Duties that you create should be self-contained and pluggable into any existing or new job role.

Naming Convention: All 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 retail privilege role names end with' _PRIV'.

Example: ALC_ALLOC_SEARCH_PRIV

Security

RAAC uses ADF security to implement functional security. The enforcement of functional security is delegated to the Fusion Middleware.

RAAC allows retailers to tailor Oracle Retail applications’ default security models to their enterprise business model. This tool also helps retailers manage their Retail applications’ security metadata. RAAC provides:

- The ability to create new Application Roles (DUTY) through the Manage Roles Mapping task.
- The ability for a retailer to change the default roles mapping based on its business needs.
- The ability for clients to backup the roles mapping to a file in their local file system.
- Export capability to facilitate the backup and migration of policies from one environment to another.

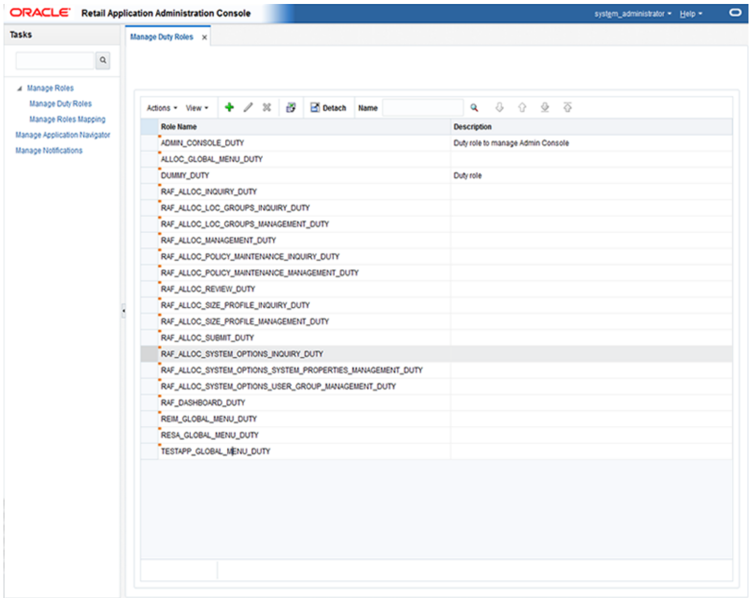
Task Flows

The Manage Roles area of RAAC consists of two main task flows used to fulfill the above mentioned business requirements.

- Manage Duty Roles
- Manage Roles Mapping

Manage Duty Roles

Figure 3–3 Manage Duty Roles Screen



Role Name	Description
ADMIN_CONSOLE_DUTY	Duty role to manage Admin Console
ALLOC_GLOBAL_MENU_DUTY	
DUMMY_DUTY	Duty role
RAF_ALLOC_INQUIRY_DUTY	
RAF_ALLOC_LOC_GROUPS_INQUIRY_DUTY	
RAF_ALLOC_LOC_GROUPS_MANAGEMENT_DUTY	
RAF_ALLOC_MANAGEMENT_DUTY	
RAF_ALLOC_POLICY_MAINTENANCE_INQUIRY_DUTY	
RAF_ALLOC_POLICY_MAINTENANCE_MANAGEMENT_DUTY	
RAF_ALLOC_REVIEW_DUTY	
RAF_ALLOC_SIZE_PROFILE_INQUIRY_DUTY	
RAF_ALLOC_SIZE_PROFILE_MANAGEMENT_DUTY	
RAF_ALLOC_SUBMIT_DUTY	
RAF_ALLOC_SYSTEM_OPTIONS_INQUIRY_DUTY	
RAF_ALLOC_SYSTEM_OPTIONS_SYSTEM_PROPERTIES_MANAGEMENT_DUTY	
RAF_ALLOC_SYSTEM_OPTIONS_USER_GROUP_MANAGEMENT_DUTY	
RAF_DASHBOARD_DUTY	
REIM_GLOBAL_MENU_DUTY	
RESA_GLOBAL_MENU_DUTY	
TESTAPP_GLOBAL_MENU_DUTY	

The Manage Duty Role task flow allows users to create or delete duties. Users can create a new duty role to map according to the retailers enterprise business model.

When a user clicks the manage duty role task flow from the regional area, it opens a new tab in the local area. A table is displayed with all the duties associated with the application. The table toolbar has the following action menu:

- Add
- Edit
- Delete
- Export to Excel

A quick search component is provided on the table toolbar to quickly look up any desired record.

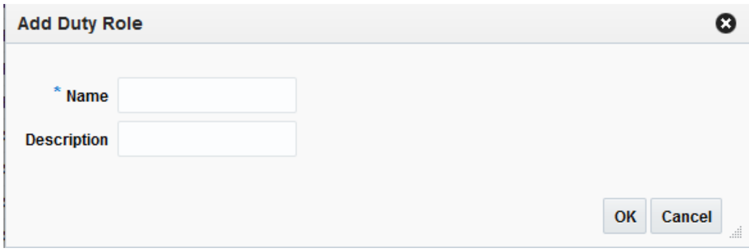
Add

The Add action is enabled at all times and allows you to add a new duty role.

Add a Duty Role

1. From the Actions menu, select Add. The Add Duty Role dialog is displayed.

Figure 3–4 Add Duty Role Dialog



2. Enter a duty name and (optionally) a description for the new duty role and click OK.

   Upon entering a name and moving to the Description field or when you click OK, RAAC performs a validation to prevent name duplication. If the name entered already exists, the following error message is displayed; A Duty with this name already exists. Enter a unique name. If the duty name is unique, the new duty is created.

   All newly created duties are reflected in the Policy Store, allowing them to be mapped to a role.

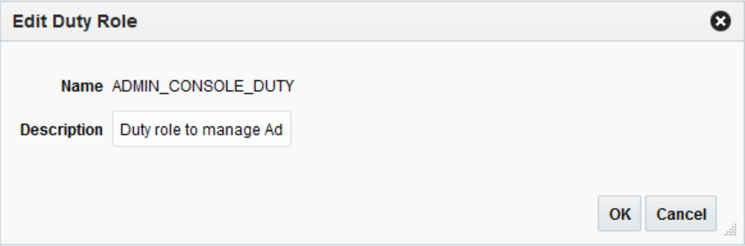
Edit

The Edit action is enabled when a duty role is selected. The Edit action is used to add or modify the description of an existing duty role.

Edit a Duty Role

1. Select the duty role you want to edit.
2. From the Actions menu, select Edit. The Edit Duty Role dialog is displayed.

Figure 3–5 Edit Duty Role Dialog



Edit Duty Role

Name ADMIN_CONSOLE_DUTY

Description Duty role to manage Ad

3. Add or modify the description of the duty role and click OK. Any updates to the description are updated to the mapping that the duty is mapped against.

Delete

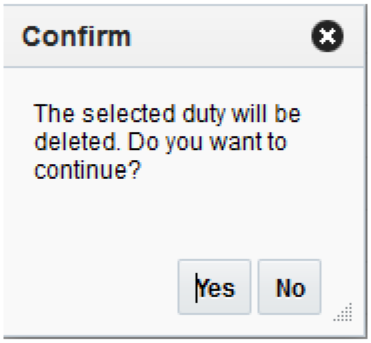
The Delete action is enabled when you select a level that can be deleted from the database.

Only empty duty roles that do not contain any child duty or child privilege roles can be deleted. This is because a child privilege is associated with application code. Therefore, to delete the duty role, you must first remove all mappings between the duty and the child duties and privileges. If you attempt to delete a duty role without first removing the mapping, the following error message is displayed; You cannot delete duty because it has duties or privileges associated with it. Remove these mappings using Manage Roles Mapping task.

Delete a Duty Role

1. Select the duty role you want to delete.
2. From the Actions menu, select Delete. The Confirm dialog is displayed.

Figure 3–6 Delete Duty Role Confirmation Dialog



The selected duty will be deleted. Do you want to continue?

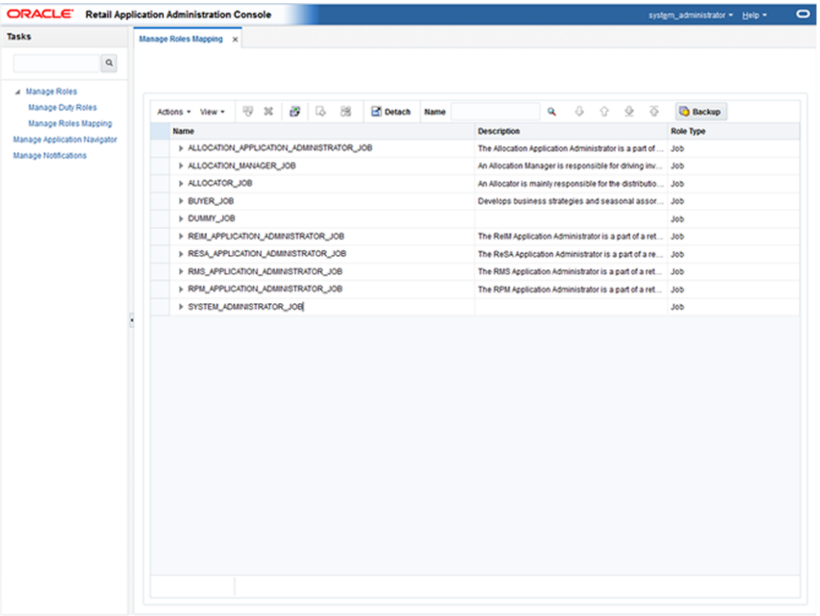
3. Click Yes to delete the selected duty role.

Export to Excel

The Export to Excel action is enabled whenever you have the Manage Duty Roles window open. This action allows you to export the Manage Duty Roles list to a Microsoft Excel spreadsheet.

Manage Roles Mapping

Figure 3–7 Manage Roles Mapping Screen



Name	Description	Role Type
ALLOCATION_APPLICATION_ADMINISTRATOR_JOB	The Allocation Application Administrator is a part of ...	Job
ALLOCATION_MANAGER_JOB	An Allocation Manager is responsible for driving inv...	Job
ALLOCATOR_JOB	An Allocator is mainly responsible for the distributio...	Job
BUYER_JOB	Develops business strategies and seasonal assor...	Job
DUMMY_JOB		Job
REIM_APPLICATION_ADMINISTRATOR_JOB	The ReIM Application Administrator is a part of a ret...	Job
RESA_APPLICATION_ADMINISTRATOR_JOB	The ReSA Application Administrator is a part of a re...	Job
RMS_APPLICATION_ADMINISTRATOR_JOB	The RMS Application Administrator is a part of a ret...	Job
RPM_APPLICATION_ADMINISTRATOR_JOB	The RPM Application Administrator is a part of a ret...	Job
SYSTEM_ADMINISTRATOR_JOB		Job

The Manage Roles Mapping task allows you to add or remove duties or privileges associated with a job role.

When you click the Manage Roles Mapping link, a new tab is opened in the local area that displays a tree table with the already associated roles of the respective application (that is, the source application from which RAAC was launched). You can then choose to perform the following actions:

- Duplicate
- Delete

- Export to Excel
- Select and Add
- Remap

A quick search component is provided on the table toolbar to quickly look up any desired record.

Duplicate

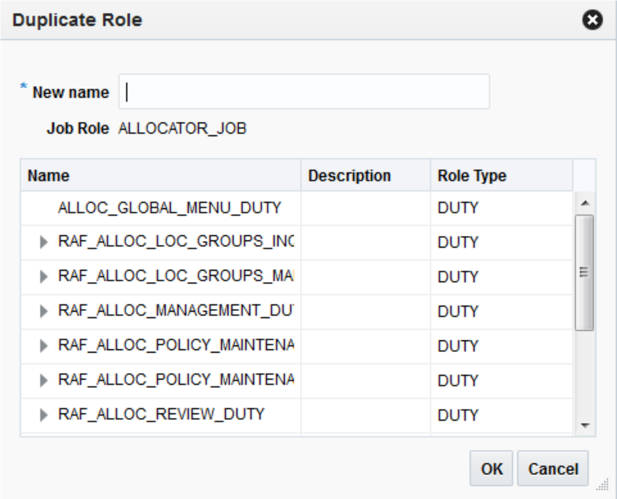
The Duplicate action allows you to copy and paste existing job roles to create a new role. For example:

Within a retailer enterprise security system there are two security groups called Allocator_xyz and Assistant_ Allocator_xyz. By default, only the Allocator role is provided. The Assistant Allocator role is a trimmed down version of the Allocator role with fewer privileges. You can use the Duplicate action to copy the Allocator role and then edit the privileges of the new role.

Duplicate a Job Role

1. Select the job role you want to duplicate.
2. From the Actions menu, select Duplicate. The Duplicate Role dialog is displayed.

Figure 3–8 Duplicate Role Dialog



* New name

Job Role ALLOCATOR_JOB

Name	Description	Role Type
ALLOC_GLOBAL_MENU_DUTY		DUTY
RAF_ALLOC_LOC_GROUPS_INC		DUTY
RAF_ALLOC_LOC_GROUPS_MA		DUTY
RAF_ALLOC_MANAGEMENT_DU		DUTY
RAF_ALLOC_POLICY_MAINTENA		DUTY
RAF_ALLOC_POLICY_MAINTENA		DUTY
RAF_ALLOC_REVIEW_DUTY		DUTY

3. Enter a name for the duplicated role and click OK.

   Upon clicking OK or tabbing out of the field, the New name is validated. The validation checks for a valid name in the enterprise security solution of the retailer. If it finds a match, then it allows the user to proceed. If the name is not valid, an error is displayed.

Delete

The Delete action is enabled when a duty or privilege role is selected.

When the user selects a duty or privilege role and clicks Delete, the user is prompted with a warning message. Click Yes and the selected role and all its associated roles mapping are removed.

Note: The Delete action does not delete the role from the database, it only removes the mapping.

Export to Excel

The Export to Excel action is enabled whenever you have the Manage Roles Mapping window open. This action allows you to export the Manage Roles Mapping list to a Microsoft Excel spreadsheet.

Select and Add

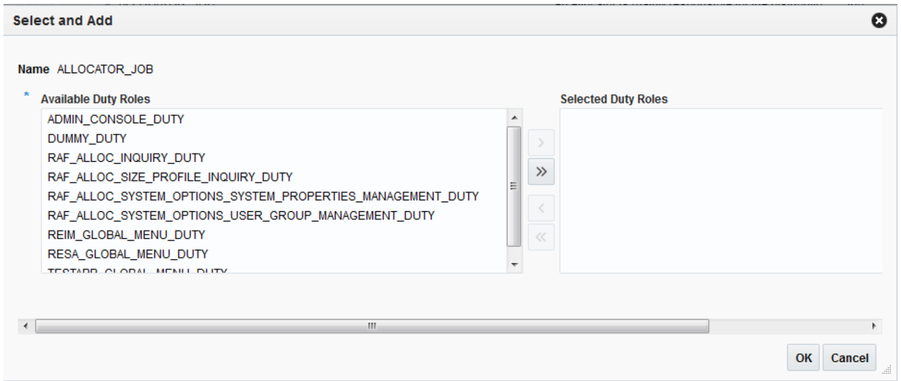
Select and Add is enabled when a job, duty, or privilege is selected. Select and Add allows you to do the following:

- Add duty roles to job roles
- Add duty and privilege roles to duty roles
- Add privilege roles to privilege roles.

Select and Add a Duty

1. Select the job role to which you want to add a duty role.
2. From the Actions menu, select Select and Add. The Select and Add dialog is displayed.

Figure 3–9 Select and Add Dialog



Select and Add

Name ALLOCATOR_JOB

* Available Duty Roles

ADMIN_CONSOLE_DUTY
DUMMY_DUTY
RAF_ALLOC_INQUIRY_DUTY
RAF_ALLOC_SIZE_PROFILE_INQUIRY_DUTY
RAF_ALLOC_SYSTEM_OPTIONS_SYSTEM_PROPERTIES_MANAGEMENT_DUTY
RAF_ALLOC_SYSTEM_OPTIONS_USER_GROUP_MANAGEMENT_DUTY
REIM_GLOBAL_MENU_DUTY
RESA_GLOBAL_MENU_DUTY
TESTAPP_GLOBAL_MENU_DUTY

Selected Duty Roles

The Select and Add dialog has selected job, duty, or privilege fields populated and a shuttle component for the user to select from all the available duties or privileges for the applications.

3. Select one or multiple duties and privileges and click OK to select and add them to the job role.

   If you select a duty level then all the associated privileges with it are inherited to the job role. If a privilege is selected then only that privilege is added under the selected duty role.

Note: To add a new job role, the user should use the Duplicate feature.

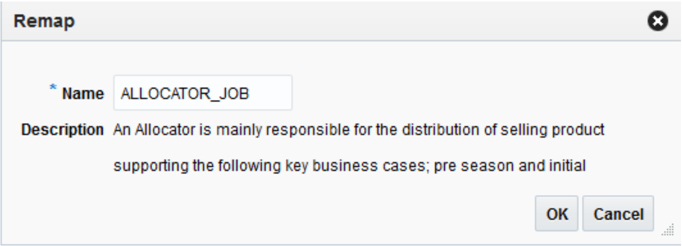
Remap

The Remap action is enabled when a job, duty, or privilege is selected. the Remap action is used to move mapping from one role to another role. During this process, a new role is created and all the associated roles beneath the previous role are moved into the new role, leaving the old role as an orphan or with other roles associated with it.

Remap a Role

1. Select the role you want to remap.
2. From the Actions menu, select Remap. The Remap dialog is displayed.

Figure 3–10 Remap Dialog



Remap

* Name ALLOCATOR_JOB

Description An Allocator is mainly responsible for the distribution of selling product supporting the following key business cases; pre season and initial

3. Enter a Name and (optionally) a Description for the remapped role.

   Upon clicking OK, the New name is validated. The validation checks for a valid name in the enterprise security solution of the retailer. If it finds a match, then it allows the user to proceed. If the name is not valid, an error is displayed.

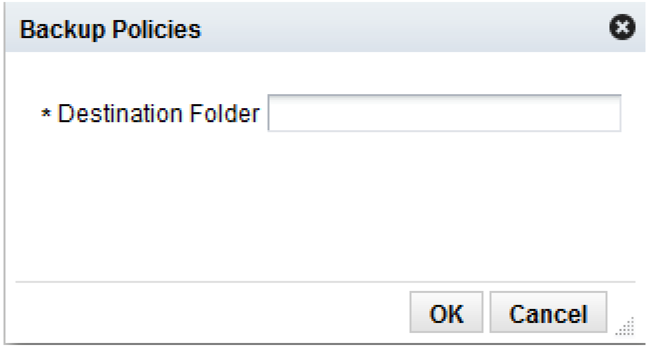
   Job roles are handled in a different manner. The owner of job role creation is the Enterprise manager so when a job role is selected and remapped it will just change the current mapping to the new mapping without creating any new job role to be left as an orphan. This validation should be done once the user enters the new name and clicks OK. If there is no matching record found an error is displayed.

Backup Roles Mapping

The Backup Roles Mapping option allows users to backup roles onto a system so that the same file can be used to load the customized roles mapping onto another server. When the user clicks on this selection, a dialog is displayed where the user can enter in the desired backup location.

1. From the Manage Roles Mapping window, click the Backup button. The Backup Policies dialog is displayed.

Figure 3–11 Backup Policies Dialog



2. Enter the Destination Folder in which to store the backup roles mapping and click OK.

   If the backup is successful, a confirmation message is displayed.

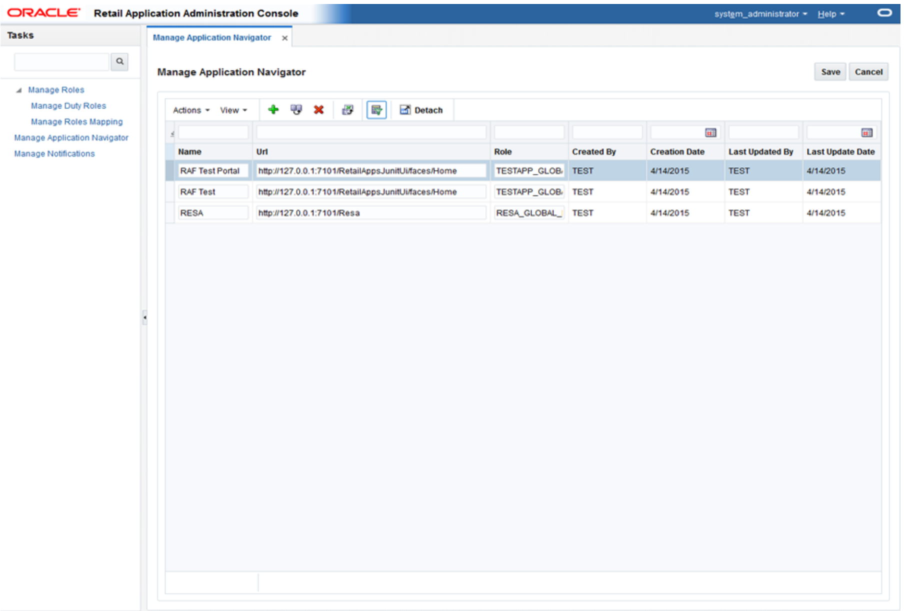
Manage Application Navigator

Application Navigator allows users to launch different applications from their current application. This lets users shuffle between multiple applications based on their privileges and avoid having to open a new tab and enter a new URL to launch an application. Users can instead click on the application, which will launch that application in a new tab or window based on the browser settings. If the application is a Single Sign-On (SSO) deployment then the user will be logged in by default and the default application page is displayed. If the deployment is non-SSO then the user is prompted to log in.

If a user has access to multiple applications (based on their defined role) they are able to see the list of available applications using application Navigator icon. When the Application Navigator icon is clicked, the drop down menu displays a list of available applications. If the user does not have access to any other applications, the icon is not displayed.

Managing Application Links

Figure 3–12 Manage Application Navigator Screen



Name	Url	Role	Created By	Creation Date	Last Updated By	Last Update Date
RAF Test Portal	http://127.0.0.1:7101/RetailAppsJunitUi/faces/Home	TESTAPP_GLOB	TEST	4/14/2015	TEST	4/14/2015
RAF Test	http://127.0.0.1:7101/RetailAppsJunitUi/faces/Home	TESTAPP_GLOB	TEST	4/14/2015	TEST	4/14/2015
RESA	http://127.0.0.1:7101/Resa	RESA_GLOBAL_	TEST	4/14/2015	TEST	4/14/2015

In order for users to see an application in the Application Navigator menu, they need to belong to the role associated with that application on the Manage Application Navigator screen. Only an application administrator is aware of the links to the different installed applications. RAAC provides the ability to add, modify and remove the links so they appear or not appear in the Application Navigator menu. Application administrator can determine which application role has access to which application based on customer's security model. For example, suppose Allocation users belonging to Allocation role X are to see the Invoice Match application link in the Allocation's Application Navigator menu. The administrator would launch RAAC from Allocation, navigate to Manage Application Navigator, add an entry for Invoice Match and enter X for the role.

Add

The Add action is enabled at all times and allows an administrator to add a new Application Navigator URL.

Add an Application Navigator URL

1. From the Actions menu, select Add. A new, empty Application Navigator entry is added to the Manage Application Navigator table.
2. Enter a name, the application URL, and the role to which the URL applies. The remainder of the columns are populated automatically.

Duplicate

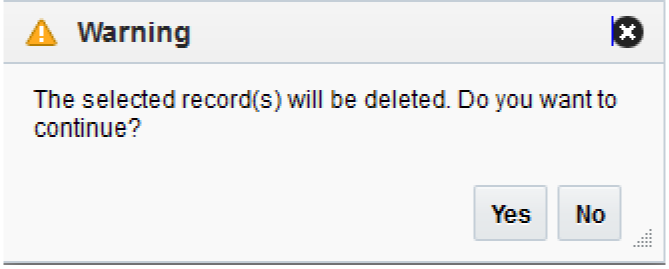
The Duplicate action adds a copy of each selected row below its originating row. The object name or unique identifier of the duplicate row is "Copy of Object Name" and appears in edit mode. Validation is performed so that a user is not presented with multiple applications with the same application name.

Delete

The Delete action is enabled when an entry is selected.

When the user selects an application navigator entry and clicks Delete, the user is prompted with a warning message. Click Yes and the selected entry is removed.

Figure 3–13 Delete Application Navigator Confirmation Dialog



Warning

The selected record(s) will be deleted. Do you want to continue?

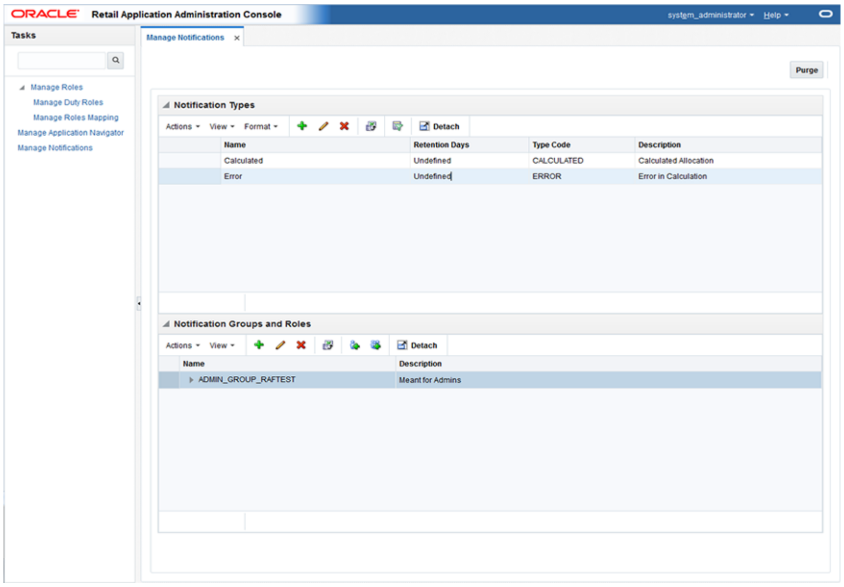
Export to Excel

The Export to Excel action is enabled whenever you have the Manage Application Navigator window open. This action allows you to export the Manage Application Navigator list to a Microsoft Excel spreadsheet.

Manage Notifications

From the RAAC left hand navigation menu, the user can access the Manage Notifications link. The Manage Notifications tab displays the Manage Notifications table. This table is laid out using a master table and a detail table. The master table consists of Notification Types and the detail table consists of Notification Groups and Roles associated with the notification types.

Figure 3–14 Manage Notifications Screen



Notification Types

Name	Retention Days	Type Code	Description
Calculated	Undefined	CALCULATED	Calculated Allocation
Error	Undefined	ERROR	Error in Calculation

Notification Groups and Roles

Name	Description
ADMIN_GROUP_RAFTEST	Meant for Admins

Notification Types

Notification Types drive how users are notified about the result of their action. This result can be communicated to one user or a group of users based on the job roles.

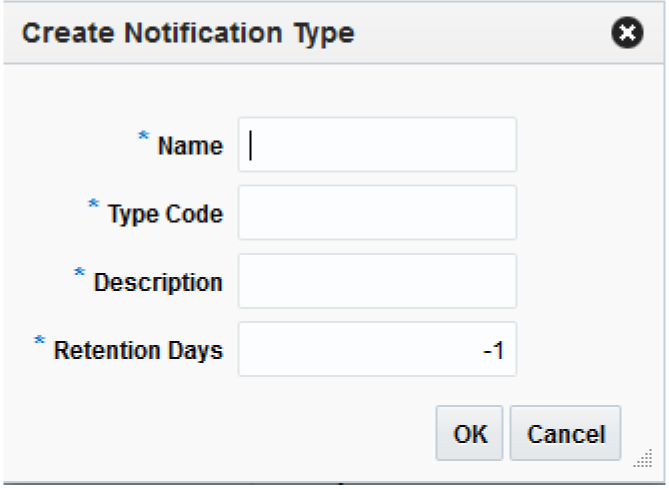
Add

The Add action is enabled at all times and allows you to add a new notification type.

Add a Notification Type

1. From the Actions menu, select Add. The Create Notification Type dialog is displayed.

Figure 3–15 Create Notification Type Dialog



* Retention Days -1

2. Each notification type requires the following information:

   - Name – The name of the notification type
   - Type Code – A meaningful name to identify the notification type

     The type code provides applications a way to uniquely identify a notification type.
   - Description – The description of the notification type
   - Retention Days – The number of days a notification of this type should be retained

     The reason for capturing this information at event type is that it allows users to have flexibility to delete certain types of notification quickly without cluttering notification views. To retain records infinitely, this value should be set to -1.

     Note: Administrators are discouraged from using '-1' for Retention Days as associated notifications will never be purged automatically. Only a system administrator can delete those manually through a sql script or direct row deletes through a sql tool.

Note: Retention Days cannot be set to zero.

3. Click OK to create the new notification type.

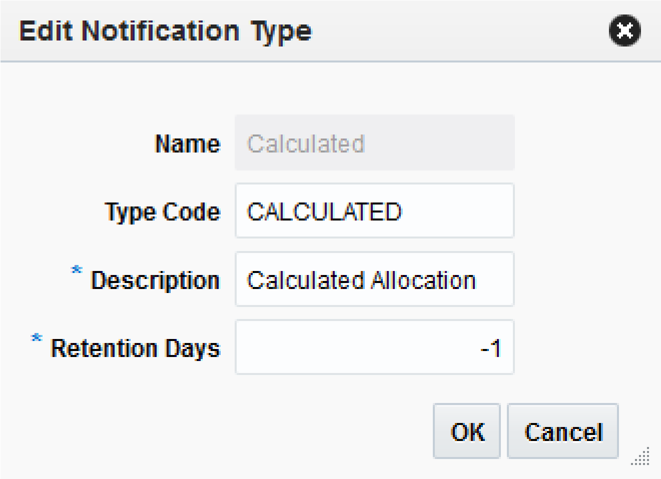
Edit

The Edit action is enabled when a notification type is selected. The Edit action is used to add or modify the description and retention days of an existing notification type.

Edit a Notification Type

1. Select the notification type you want to edit.
2. From the Actions menu, select Edit. The Edit Duty Role dialog is displayed.

Figure 3–16 Edit Notification Type Dialog



Edit Notification Type

Name: Calculated

Type Code: CALCULATED

* Description: Calculated Allocation

* Retention Days: -1

3. Modify the type code, description, or retention days of the notification type and click OK.

Delete

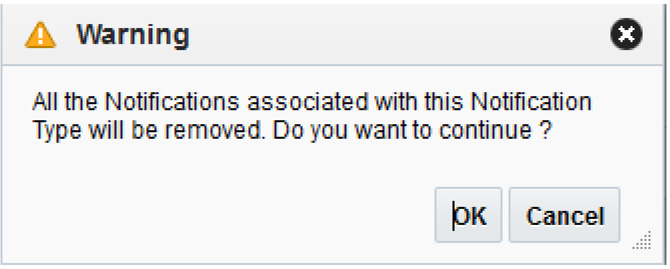
The Delete action is enabled when you select a notification type that can be deleted from the database.

Deleting a notification type removes the notification type and all of the notification type’s associated roles and groups. Any past notifications are also removed in order to maintain data integrity.

Delete a Notification Type

1. Select the notification type you want to delete.
2. From the Actions menu, select Delete. The Confirm dialog is displayed.

Figure 3–17 Delete Notification Type Confirmation Dialog



Warning

All the Notifications associated with this Notification Type will be removed. Do you want to continue ?

3. Click OK to delete the selected notification type.

Export to Excel

The Export to Excel action is enabled whenever you have the Manage Notifications window open. This action allows you to export the Notification Types list to a Microsoft Excel spreadsheet.

Notification Groups and Roles

The Notification Groups and Roles table consists of all the roles associated with a notification type master record. If there are no records associated with the notification type then a "no data to display" message appears in the Notification Groups and Roles table when the notification type is selected in the Notification Types table.

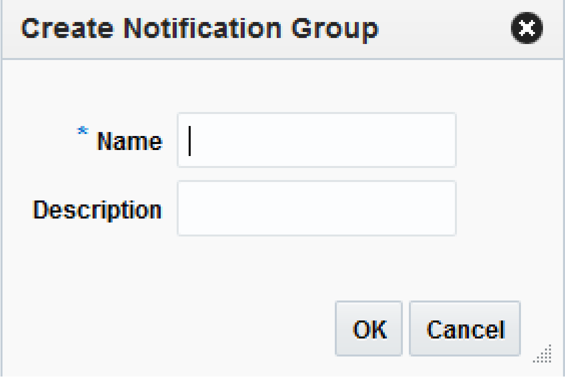
Create/Add Notification Group

The Create action is enabled at all times when a notification type is selected and allows you to create a new notification group associated with the selected notification type.

Create a Notification Group

1. From the Actions menu, select Create (or Add Notification Groups). The Create Notification Group dialog is displayed.

Figure 3–18 Create Notification Group Dialog



Create Notification Group

2. Enter a name and (optionally) a description of the group.
3. Click OK to create the new notification group.

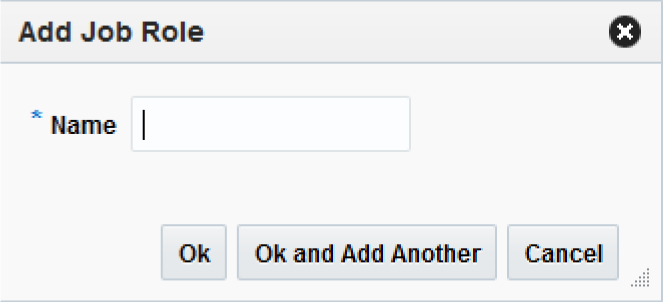
Add Job Role

Once a notification group is created, you can add job roles that apply to the selected group.

Add a Job Role

1. From the Actions menu, select Add job Role. The Add Job Role dialog is displayed.

Figure 3–19 Create Job Role Dialog



Add Job Role

* Name

2. Enter a name for the job role.
3. Click OK to create the new job role or OK and Add Another to create additional job roles.

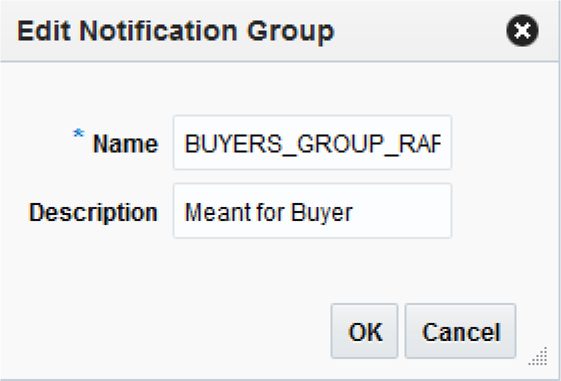
Edit

The Edit action is enabled when a notification group or job role is selected. The Edit action is used to modify the name and description of an existing notification group or job role.

Edit a Notification Group

1. Select the notification group or job role you want to edit.
2. From the Actions menu, select Edit. The Edit Notification Group (or Edit Job Role) dialog is displayed.

Figure 3–20 Edit Notification Group Dialog



Edit Notification Group

* Name: BUYERS_GROUP_RAF

Description: Meant for Buyer

3. Modify the notification group or job role attributes and click OK.

   Upon clicking OK, the New name is validated. The validation checks for a valid name in the enterprise security solution of the retailer. If it finds a match, then it allows the user to proceed. If the name is not valid, an error is displayed.

Delete

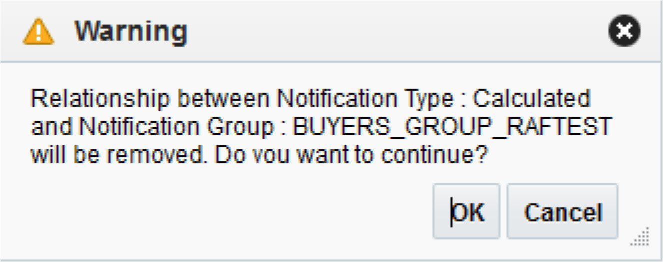
The Delete action is enabled when you select a notification group or job role that can be deleted from the database.

Deleting a notification type removes the notification type and all of the notification type's associated roles and groups. Any past notifications are also removed in order to maintain data integrity.

Delete a Notification Type

1. Select the notification type you want to delete.
2. From the Actions menu, select Delete. The Confirm dialog is displayed.

Figure 3–21 Delete Notification Group Confirmation Dialog



Relationship between Notification Type : Calculated and Notification Group : BUYERS_GROUP_RAFTEST will be removed. Do you want to continue?

3. Click OK to delete the selected notification group or job role.

Export to Excel

The Export to Excel action is enabled whenever you have the Manage Notifications window open. This action allows you to export the Notification Groups and Roles list to a Microsoft Excel spreadsheet.

4

Oracle Retail Merchandising System

This chapter gives an overview of the information maintained by RMS and RMS integration with other applications.

Information Maintained by RMS

RMS is the System of Record (SOR) for the following:

Seed Data

RMS contains data that must be populated at the time of installation. Either the data is required by the application, or the data is static and can be loaded for any client. The code tables CODE_HEAD and CODE_DETAIL are examples of tables with system-required values that must be loaded at the time of installation. Additional codes can be added as needed to these tables after installation, by either using the online form or additional client scripts. Customization of the seed data is based on the client's requirements.

Additionally, some configuration tables must be populated for the application to open and function correctly, even though the configuration values can be modified later. These configurations are maintained in System Options set of tables, and are required for initial setup that can be updated prior to the implementation, to reflect final configuration.

Foundation Data

Foundation data is the base for all future information on which RMS builds. This information needs to be present before you begin using the system. The majority of foundation data can be set up online; more commonly, a client performs a data conversion process to import this information from legacy systems.

Foundation data consists of three types of information:

- Organizational hierarchy
- Merchandise hierarchy
- Supplier and partner management

Organizational Hierarchy

The organizational hierarchy allows you to create maximum of six levels of hierarchy relationships to support the operational structure of your company.

In a Retail store, the customer walks through the store and buys, retails procure goods from suppliers and the retailer acts as a customer. In a Franchise store, the retailer supplies goods to the franchise customers, the retailer acts as a supplier and tracks the sales.

You can create a preferred organizational structure to support consolidated reporting at various levels of the company. You can also assign responsibility for any level of the hierarchy to one or more people to satisfy internal reporting requirements.

The organizational hierarchy also supports two types of stores to satisfy the franchise requirements:

- Franchise
- Company

A company store act as retail store in RMS.

The Franchise stores are used to support franchise business models. Other applications, such as RPM and RWMS, view franchise stores as they would view any other store in RMS; however, RMS performs special processing based on the store types.

Merchandise Hierarchy

The merchandise hierarchy allows you to create maximum of six levels of hierarchy relationships to support the product management structure of your company. You can assign a buyer and merchandiser at the division, group, and department levels of the merchandise hierarchy. You can also link a lower level to the next higher level. For example, you must indicate to which group a department belongs, or to which division a group belongs.

Supplier and Partner Management

A supplier supplies the merchandise and a partner is involved in other financial taxation that results in an invoice. Supplier and partner management provides functionality to create and store valid merchandise suppliers and partners. You can maintain a variety of information about suppliers such as financial arrangements, inventory management parameters, types of EDI transactions, and invoice matching attributes. Suppliers are typically created in a financial system and interfaced into RMS; supplier enrichment can be maintained in RMS.

The supplier structure can be extended to supplier-parent and supplier-site relationships, to accommodate financial systems that support this configuration. A supplier site is a location from which the supplier ships merchandise.

Item Maintenance

RMS is responsible for the creation and maintenance of all items. RMS uses a flexible data hierarchy for an item, with levels that allow you to model items in a desired way. The hierarchy consists up to three levels, highest (level 1) to lowest (level 3). Within the defined levels for an item family, one level is denoted as the transaction level. This is the level at which all inventory and sales transaction takes place. This model gives retailers a flexibility to create families of items that share common characteristics.

RMS creates several types of items, such as regular items, deposit items, packs, concession items, consignment items, and transformable items.

Through item maintenance, RMS also maintains the relationships of items with other entities such as suppliers, locations, and attributes.

Purchasing

The Purchase Order module allows you to create and maintain purchase orders in a variety of ways. It provides commitments to vendors for products in specific amounts for specified locations. Purchase orders are created manually or automatically through replenishment or from an external system. They can be created against entered contracts and deals, or directly through direct store delivery or Vendor Managed Inventory (VMI). RMS also provides the ability to maintain the items, locations, and quantities ordered for Purchase Orders.

Contracts

The contract dialog gives you the ability to create, maintain, submit, and approve contracts. A contract is a legally binding agreement with a supplier to supply items at a negotiated cost.

In RMS, the contracting functions fit closely with the replenishment and ordering functions. The main functions of the Contracts window are to book manufacturing time, track supplier availability and commitments, and match them with business requirements. The main business benefit of contracting is to achieve supplier involvement during the planning phase of a retailer's business.

Deals

Deals management allows you to create and maintain deals with partners or suppliers. Deal partners can be suppliers, distributors, and manufacturers. Within a deal, clients create deal components, specify the items for each deal component, and define thresholds.

Components are deals or parts of deals that a retailer receives from a supplier. There can be multiple components in a single deal. You must define thresholds to define the quantity or amount that must be purchased or sold to receive the deal. RMS components include off-invoice deals, rebates, vendor-funded promotions, vendor-funded markdowns, and fixed deals.

You also define the items and locations for which the deal can be applied. You can choose to include or exclude locations as necessary.

You also define the Proof Of Performance (POP) terms for a deal. POP terms are defined by the deal vendor that offers the deal. For deals, POP terms are defined at the deal, deal/component, or deal/component/item-location combination. For fixed deals, POP terms are defined at the deal level.

The deal pass-through functionality allows a percentage of a deals discount to be passed from a warehouse to a franchise store. This functionality applies to franchise stores.

For clients that choose to use supplier sites with RMS, deals are managed at the supplier parent level.

Cost Management

For cost changes, the Cost Management dialog gives you the ability to:

- Accept cost changes received through EDI (flat files)
- Create a cost change
- Edit a cost change
- View a cost change

A cost change is an adjustment to the supplier cost of an item, either up or down. Before you create a cost change, you must create a list of user-defined cost change reasons and then apply a reason to each cost change. This is useful in reporting.

The initial cost of an item is established at item setup. The cost of the item is adjusted in the item record until the status of the item is Approved. After the item is approved, any cost changes needs to be handled through the cost change dialog.

When a cost change is submitted through EDI, the EDI cost change is reviewed and released to create a cost change document. The cost change document is then viewed and submitted for approval.

When a cost change document is created online, you enter the cost change, an event description, an effective date, and a reason code, and then submit the cost change for approval.

After a cost change is approved, the item/supplier cost record is updated. Any outstanding purchase order line items with no received units are recalculated, if recalculation is indicated on the cost change.

Additionally, you use the Cost Management dialog to create cost zone groups for zone-type expenses for item estimated landed cost. Zone-type expenses are incurred when imported goods are moved from the discharge port to the purchase order receiving location. Because the expenses can vary depending on the distance between the discharge port and the receiving location, cost zones can be configured to appropriately reflect the expenses. The locations (stores and physical warehouses) must be grouped to reflect the expense variances for moving the goods. Normally a zone strategy is used for these cost zone groups, but it is possible that every location within the company has different expenses to move the goods from the discharge port. If that is the case, a store strategy would be used. If every location within the company has the same transportation costs from the discharge port, a corporate strategy is adequate (but not when multiple currencies are being used). After these cost zone groups are defined, they are added to new items as they are created, in anticipation of the expense profiles that are needed for the items.

Multiple Sets of Books

Support for multiple sets of books provides better integration with financials systems that supports Multiple Sets of Books within a single installation. Multiple Sets of Books option is enabled by default in RMS and the client will need to set up additional location-specific foundation data, including:

- Organizational units
- Transfer entities
- Set of books IDs

Inventory Control

Inventory functionality in RMS is the core of the application. Inventory is tracked perpetually and financially in RMS. The following describes perpetual inventory tracking. For information on financial inventory tracking, see Stock Ledger.

RMS achieves inventory control through functions that include transfers, Return to Vendor (RTV), Inventory Adjustments, Sales Upload, Purchase Order Receipts (shipments), Stock Counts, Allocations, Franchise Orders and Returns, and Customer Orders.

Transfers

Transfers in RMS provide an organized framework for monitoring the movement of stock. RMS creates and maintains transfers; however, you can also interface transfer information into RMS from other systems.

RMS supports a number of different types of transfers such as intercompany transfers, book transfers, Purchase Order-linked transfers, externally generated transfers, customer orders and franchise order. Transfer functions also support the movement of one or more items between two internal RMS locations, and multi-leg transfers in which the intermediate location is considered a finisher location. Finishers are locations where work is performed on merchandise, such as dying fabric and attaching labels.

Mass return transfers are used to reallocate merchandise to locations or to return merchandise to the supplier.

Returns to Vendor

Return to Vendor (RTV) transactions are used to send merchandise back to a vendor. The RTV transaction in RMS allows one or more items to be returned to a single vendor. For each transaction, the items, quantities, and costs are specified. Upon shipment out of a location, inventory is removed from the stock on hand.

RTVs are created manually in RMS or imported from an external system. RMS also provides the ability to maintain RTVs. Shipped RTVs create a debit memo or credit note request (based on supplier configuration) in the invoice matching staging table in RMS, for export to Oracle Retail Invoice Matching.

Inventory Adjustments

Inventory adjustments are used to increase or decrease inventory to account for events that occur outside the normal course of business (for example, receipts, sales, stock counts). Inventory adjustments are created in RMS or imported from an external system (store or warehouse application). RMS supports two types of inventory adjustments; stock on hand or unavailable inventory. Inventory adjustments can also be created by locations for multiple items, by item for multiple locations, or through a product transformation for a specific location.

Note: The following Inventory Adjustment Reason Codes are required by RMS and cannot be deleted unless the noted functionality is not utilized:

- Reason Code 1 - Used for wastage adjustments
- Reason Code 2 - Used for adjustments against processed stock counts
- Reason Code 13 - Used for inventory adjustment for unavailable receipts
- Reason Code 190 - Used for inventory adjustments related to 'destroy on receipt' situations for Wholesale and Franchise
- Reason Code 191- Used for Customer Returns without inventory

Additionally, reason codes must be synchronized between SIM and WMS, or any other system communicating inventory adjustments to RMS.

Purchase Order Receipts (Shipments)

Purchase order receipts (Shipments) record the increment to on-hand when goods are received from a supplier. Weighted average cost (WAC) is recalculated at time of receipt using the PO landed cost. Transaction audit records are created for financial audit, and the receiver is made available for invoice matching.

Stock Counts

Stock counts are the processes by which inventory is counted in the store and compared against the system inventory level for discrepancies. RMS supports two types of stock counts:

- Unit stock counts

  These adjust the on hand quantities for the item-locations affected and create an inventory adjustment transaction for the stock ledger.

- Unit and value stock counts

  These adjust the on hand quantities for the item-locations affected and adjust the stock ledger to the results of the stock count.

Replenishment

Automated replenishment constantly monitors inventory conditions. Based on inventory conditions, purchase orders or transfers are created to fulfill consumer demand.

Automated replenishment parameters are set up at the supplier, supplier/department, and supplier/location or supplier/department/location level. These parameters include:

- Review cycle and order control
- Due order processing
- Investment buy attributes
- Scaling constraints
- Rounding attributes
- Supplier minimums
- Truck splitting constraints

Items can be set up for automated replenishment through the Item Maintenance dialog, either individually or through item lists.

Automated replenishment also supports different methods to determining whether purchase orders are created and quantities ordered. These replenishment methods are applied at the item/location.

- **Constant** is a stock-oriented method in which the item is replenished when the inventory level falls below a specified level.
- **Min/Max** is a stock-oriented method in which the item is replenished up to the maximum when the inventory level falls below a specified minimum stock level.
- **Floating Point** is a stock-oriented method in which the item is replenished when the inventory level falls below a dynamic system-calculated maximum stock level.

- **Time Supply** is a stock-oriented method in which replenishment is based on the number of days of supply for the item a retailer wants in inventory. The Time Supply method requires a forecasting system.
- **Time Supply Seasonal** is the same as Time Supply, but it takes seasonality and terminal stock into account. The Time Supply Seasonal method requires a forecasting system.
- **Time Supply Issues** is used only by warehouses, this is the same as Time Supply, but it uses warehouse issues forecast rather than store sales forecast. The Time Supply Issues method requires a forecasting system.
- **Dynamic** is a method that controls inventory using dynamic calculations of order point and order quantities based on a number of factors, including forecast sales over order lead time, review lead time, inventory selling days, lost sales factor, and safety stock. The Dynamic method requires a forecasting system.
- **Dynamic Seasonal** is the same as Dynamic, but it takes seasonality and terminal stock into account. The Dynamic Seasonal method requires a forecasting system.
- **Dynamic Issues** is used by warehouses only, this is the same as Dynamic, but it uses warehouse issues forecast rather than store sales forecast. The Dynamic Issues method requires a forecasting system.
- **Store Orders** is a method that allows replenishment to look at the store order need quantity when determining the recommended order quantity.

Franchise Management

The Franchise Management allows the retailer to manage their franchise business in the following scenarios:

- Retailer owns and manages the inventory for a franchise location.

  In this case, the franchise customer (location) needs to be set up as stockholding stores in RMS, with a store type as Franchise. A stockholding franchise store functions similar to a company store with locations of inventory transactions such as Replenishment, Allocation, Stock Counts and Inventory Adjustments being allowed for such stores in RMS and the pricing being carried out in RPM. The main differences in these stores are, the way in which the orders are captured and accounted financially.
- Retailer does not own or manage inventory for a franchise location.

  Here the retailer does not manage the inventory for a franchise location or wherein the wholesale operations constitute a small fraction of the retailers business and thus does not warrant a separate Order Management System.

In both these scenarios, non-stockholding stores must be setup in RMS to represent these franchise (or wholesale) customers.

Franchise Pricing

Franchise pricing determines the price that is charged on a franchise partners for an item. Pricing for these stores is maintained in the future cost table. The pricing for franchise locations are determined by setting up cost templates in RMS and associating these templates with an item or a franchise location. Franchise pricing includes any landed costs and applicable deals through deal pass-through to the final pricing.

The user has an option to override the future cost franchise price and instead define a fixed price to be charged for an item for manually or through the batch upload orders.

Franchise Ordering

Franchise store can source the merchandise from company locations (Warehouse or Store) or from a supplier. The franchise order can be initiated manually using the franchise order form or by an batch upload process using flat file received from an external application. The franchise order created using the flat file also creates a purchase order for supplier sourced and transfers for company location sourced orders.

In addition, franchise order gets initiated in response to any inventory transaction process where the receiving location is a franchise store and the sending location is a company location or supplier. Some of these inventory transactions are Replenishment Requests, Allocation, Store Orders, Items requests; AIP generated Purchase Orders or Transfers, and externally generated transfers.

Franchise Returns

Franchise stores returns the items back to the company location (warehouse or store). Item return from a franchise store directly to the supplier is not allowed. The franchise returns can either be a physical return to the company location or can be a book return. Book Return is possible when the item is destroyed at the site, in such scenario the inventory is not physically returned but can be financially accounted.

The franchise return can be initiated manually using the franchise return form or by batch upload process using flat file received from an external application. Franchise returns results in creating a transfer to track the inventory movement.

In addition, franchise returns gets created in response to any inventory transaction where the sending location is a franchise store and receiving location is a company location. Some of these processes are AIP or externally generated transfers.

Stock Ledger

The stock ledger in RMS records the financial results of the merchandising processes such as buying, selling, price changes, and transfers. All of these transactions are recorded in the RMS stock ledger and rolled up to the subclass/location level for days, weeks, and months, depending on calendar settings. The aggregate levels in the stock ledger are used to measure inventory amounts and merchandise profitability. The stock ledger is mainly used for reporting purposes; however, there is some online visibility as well.

The stock ledger supports multiple currencies. All transaction-level information is stored in the local currency of the store or warehouse where the transaction occurred. As transaction-level information is rolled up to the aggregated levels in the stock ledger, records are kept in local currency and converted to primary currency. This allows corporate reporting to be performed in the primary currency of the company, while still providing visibility by location to the profitability in the local currency.

The stock ledger supports both the retail and cost methods of accounting. The cost method can use standard cost or average cost, depending on how the system is configured. The stock ledger supports both the retail (4-5-4) and the normal (Gregorian) calendar. If the retail calendar is used, data is maintained by the 4-5-4 month and the week. If the normal calendar is used, data is maintained only by the Gregorian month. Data can also be maintained daily using the retail (4-5-4) or normal (Gregorian) calendar.

RMS supports multiple sets of books. Clients that use multiple sets of books assign RMS locations to a particular set of books defined in an external financial system.

Changes to the stock ledger affect the set of books with which a particular transaction is associated.

Investment Buy

Investment buy facilitates the process of purchasing inventory in excess of the replenishment recommendation in order to take advantage of a supplier deal or to leverage inventory against a cost increase. The inventory is stored at the warehouse or in outside storage to be used for future issues to the stores. The recommended quantity to investment buy, that is, to order, is calculated based on the following:

- Amount of the deal or cost increase
- Upcoming deals for the product
- Cost of money
- Cost of storage
- Forecasted demand for the product, using warehouse issue values calculated by Oracle Retail Demand Forecasting
- Target return on investment (ROI)

The rationale is to purchase as much product as profitable at the lower cost and to retain this profit rather than passing the discount on to customers and stores. The determination of how much product is profitable to purchase is based on the cost savings of the product versus the costs to purchase, store and handle the additional inventory.

Investment buy eligibility and order control are set at one of these four levels:

- Supplier
- Supplier-department
- Supplier-location (warehouse locations only)
- Supplier-department-location

Warehouses must be enabled for both replenishment and investment buy on RMS WH (warehouse) table.

The investment buy opportunity calculation takes place nightly during the batch run, after the replenishment need determination, but before the replenishment order build. The investment buy module IBCALC.PC attempts to purchase additional inventory beyond the replenishment recommendation in order to achieve future cost savings. Two distinct events provide the incentive to purchase investment buy quantities:

- A current supplier deal ends within the look-ahead period.
- A future cost increase becomes active within the look-ahead period.

The calculation determines the future cost for a given item-supplier-country-location for physical warehouse locations only.

If the order control for a particular line item is buyer worksheet, it might be modified in the buyer worksheet dialog, and can be added to either new or existing purchase orders.

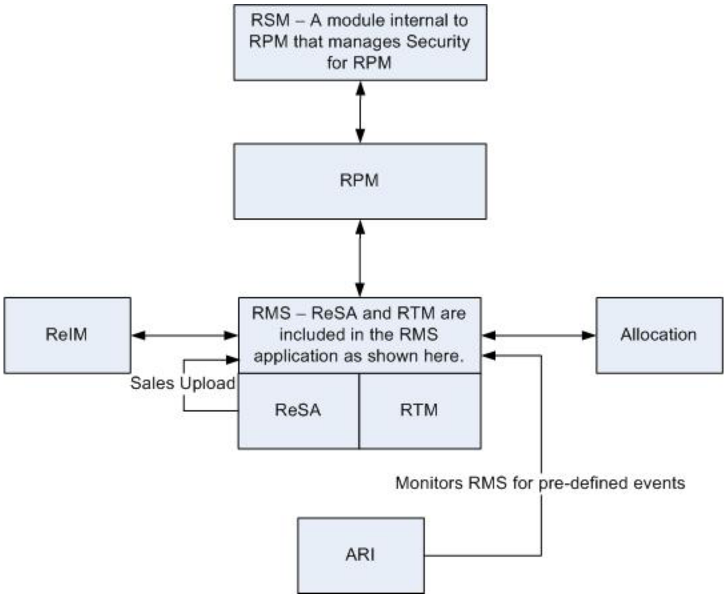
RMS Integration with Other Applications

RMS provides essential information to all of the Oracle Retail Merchandising Operations Management applications (ReSA, RTM, RPM, ReIM, Allocation), and interacts with all of them. RMS exists on the same database schema as all of the other applications, which provides flexibility in how information is shared between RMS and the other Oracle Retail Merchandising Operations applications.

Information is shared with other Oracle Retail Merchandising Operations Management applications through direct reads from Oracle Retail Merchandising Operations Management application tables, calls to Oracle Retail Merchandising Operations Management application packages, batch processes, and the Oracle Retail Integration Bus (RIB) if the client is using this option.

The following diagram illustrates the RMS location in the merchandising footprint.

Figure 4–1 RMS location in the Merchandising Footprint


RSM – A module internal to RPM that manages Security for RPM
RPM
ReIM
RMS – ReSA and RTM are included in the RMS application as shown here.
Allocation
Sales Upload
ReSA
RTM
Monitors RMS for pre-defined events
ARI

RMS and RTM

Oracle Retail Trade Management (RTM) and RMS share the same database instance. When RTM is enabled in an RMS instance, certain import-specific data maintenance is required for country, supplier, partners and items. These are directly updated into the RMS database and subsequently used in RTM.

RMS and ReSA

Oracle Retail Sales Audit (ReSA) and RMS share the same database. ReSA shares some of its master data with RMS. Foundation data such as stores a company/location close dates, location traits, bank setup and tender types are maintained in RMS and used in ReSA.

Sales Upload Process

Current reference data is retrieved from RMS into ReSA by the batch program SAGETREF. The data is extracted into multiple data files. The data in the files is used by the batch program SAIMPTLOG as reference data for doing validation checks on the POS/OMS transaction data during the data upload to ReSA. Having the reference in data file formats increases the performance of the SAIMPTLOG process. SAGETREF generates the following reference files:

- Items
- Wastage
- Sub-transaction level items
- Primary variant relationships
- Variable weight PLU
- Store business day
- Code types
- Error codes
- Store POS
- Tender type
- Merchant code types
- Partner vendors
- Supplier vendors
- Employee IDs
- Banner IDs

Along with the reference files, the following files are generated:

- Promotions File - This file contains RPM promotions.
- Currency File - This file contains valid currency codes in RMS.
- Warehouse File - This file contains valid physical warehouses from RMS.
- Inventory Status File - This file contains valid inventory status values from RMS.

All clean and audited sales and returns data is extracted from ReSA into a POSU file by the batch program SAEXPRMS. All sale and return transactions that do not have RMS errors are extracted into the file. The sales audit system options parameter work unit controls the export of data into files in case of the presence of RMS errors in the POS/OMS transaction data. The shell scripts UPLOADSALES.KSH and SALESPROCESS.KSH will load the data from the POSU file into the RMS tables.

RMS and RPM

RPM exists on the same database schema as RMS which allows information to be shared between applications through direct database reads, package calls, and batch processes. RPM uses APIs to facilitate the exchange of information with RMS.

RPM provides the following to RMS:

- **Regular Price Event Approval/Modification/Deletion**-Regular price event creation, modification, or deletion triggers a call to an RMS API to generate (or remove if deleting) the ticket request information.
- **Price Event Execution**- For regular, promotional, or clearance price events that end or are set to go into effect, the PriceEventExecutionBatch owns the process. When the pricing event has been processed by the batch program it updates item/location pricing in RMS by interfacing with the RMSSUB_PRICECHANGE API in RMS.
- **Initial Pricing**-Initial pricing for items in RMS is dependent upon the primary zone group for the item defined in RPM and characteristics of that zone group.

These characteristics include markup percent, markup percent type, and pricing guides. RPM provides this information to RMS through an API (MERCH_RETAIL_API_SQL).

- **Deal Creation**- For Price Changes and Clearances, RPM creates new details under existing deals. Promotions can create new deals in addition to creating details under existing deals. When this occurs RPM uses an RMS API (PM_DEALS_API_SQL) to create the deal in RMS.

RMS provides the following to RPM:

- **Foundation Data** is essential to RPM functionality. To successfully set up price changes RPM requires RMS merchandise hierarchy, organizational hierarchy, and suppliers. RPM is able to access this information through the RMS database.
- **Item-Price** changes created in RPM ultimately relate to an item/location within RMS. RPM must know all eligible items currently in the merchandising system, the locations at which they are eligible (item/location relationships) in any status and the suppliers associated with the items. RPM can access this information through the RMS database.
- **Competitive Pricing Information**-RPM has the ability to create price changes based off competitive activity in the marketplace. RPM is able to access this information through the RMS database.
- **Deals** can be associated with price changes in RPM (including vendor funded promotions). In order to associate a price change to an existing deal RPM needs visibility to the deals currently available in the RMS system. RPM is able to access this information through the RMS database.
- **Event Notification** -To ensure appropriate processing, RPM must be notified of certain events:
  - **Store/Warehouse Creation**-A zone structure must be added to RPM when new stores and warehouses are created in RMS. To do this RMS provides RPM with the store and/or virtual warehouse being added, its pricing location, and its currency (to ensure it is the same as the zone it is being added to). A store/virtual warehouse creation event in RMS triggers an API call to RPM to perform the necessary processing.
  - **Item/Location Creation**-When new item/location relationships are established, RPM must verify that no future retail records currently exist, create an initial future retail record (for sellable items), and determine if there are existing price changes that would affect the item resulting in a future retail record for the price change as well. An item/location creation event in RMS triggers data to be staged in RPM so that it is picked up for the batch processing.
  - **Item Modification** is used to notify RPM when an item is reclassified. The details of the reclassification are written to an item modification table in RPM for the next batch processing run. An item modification creation event in RMS triggers an API call to RPM to perform the necessary processing.
  - **Department Creation** is used to notify RPM when new departments are created in RMS. RPM creates aggregation-level information for the new department using predefined system defaults. A department creation event in RMS triggers an API call to RPM to perform the necessary processing.

RMS and Allocation

RMS provides the following to Allocation:

- **Foundation Data** is essential to all areas of Allocation including valid locations to allocate to and from, location groupings, and valid merchandise hierarchies to allocate within.
- **Item**-Allocations are generated at the item location level so it is necessary that the Allocation application understand what items and item/locations are eligible in the system.
- **Purchase Order**-One of the sources from which a user can allocate. Allocation relies on RMS to provide Purchase Order information.
- **Transfer**-One of the sources from which a user can allocate. Allocation relies on RMS to provide Transfer information.
- **Bill Of Lading (BOL)**-One of the sources from which a user can allocate. Allocation relies on RMS to provide BOL information.
- **Advance Shipment Notices (ASN)**-One of the sources from which a user can allocate. Allocation relies on RMS to provide ASN information.
- **Inventory**-In order to determine the correct need at an item location level before performing an allocation the application needs visibility to the current on hand inventory at each location being allocated to. Allocation relies on RMS to provide inventory information at the item/location level.
- **Sales Information**-Allocation can use historical sales, forecast sales, and plan sales in order to determine the need at an item/location level for an allocation. Allocation interfaces this information in from external planning system to an Allocation table.

Allocation provides the following to RMS/RTM/ReSA:

- **Allocations**- When the allocations is moved to Approved or Reserved status, the Allocation is written to RMS tables to give visibility to the allocation results.
- **Purchase Orders** created by What-If process in Allocation-If this option is enabled in Allocation the client can create a simulated allocation based on current need and then automatically create a purchase order from that Allocation in RMS to fulfill that need. Allocation uses an RMS API to build the purchase order in RMS.

RMS and Invoice Matching

RMS provides the following to Invoice Matching:

- **Foundation Data** is essential to all parts of invoice matching including valid locations for Invoices to be implemented at, valid suppliers to receive invoices from, and supplier addresses to send credits and debits based on invoice matching results.
- **Item** is essential to the invoice matching process as item information ensures that invoices being received are valid for the business. For example an item received on an invoice is carried by the client, is supplied by the supplier who sent the invoice, and is carried in the locations for which the item was received.
- **Purchase Orders** are used by Invoice Matching to facilitate the invoice matching process which is performed at the purchase order location level.
- **Shipments**-Shipment information is used by Invoice Matching to determine if a PO has been received yet which affects the matching algorithm used by the AutoMatch batch program in Invoice Matching.
- **Deals and Rebate**-Invoice Matching creates credit memos, debit memos, and credit requests based on deal and rebate information in RMS for processing by the

financial (AP) system. This is performed by the ComplexDealUpload and FixedDealUpload batch processes that read from RMS staging tables.

Invoice Matching provides the following to RMS:

- **Invoice Matching** results for shipments-Shipment records are updated with the invoice matching results from the invoice match process (this involves updating the match status and quantity matched of the shipments in question). The matching process is handled by the AutoMatch batch process in Invoice Match which attempts to match all invoices in ready-to-match, unresolved, or multi-unresolved status.
- **Receiver Cost Adjustments**-An API is executed when invoice matching discrepancies are resolved through a receiver cost adjustment. The API updates the purchase order, shipment, and potentially the item cost in RMS, depending on the reason code action used.
- **Receiver Unit Adjustments**-An API is executed when invoice matching discrepancies are resolved through a Receiver Unit Adjustment. The API updates the purchase order and shipment in RMS to complete the transaction.
- **Closing unmatched shipments**-Invoice matching closes the invoice matching status for shipments in RMS after a set period of time (defined by the client in system options). This updates the invoice matching status of the shipment on the shipment table in RMS. This process is managed by the ReceiptWriteOff batch program.

RMS and ARI

ARI is a monitoring system that interacts with any applications database (including RMS). It stores the RMS database structure as metadata information, monitors the event defined by a client and notifies the client when the event occurs.

RMS and Xcenter/Xstore

RMS provides the following to Xcenter/Xstore:

- Foundation Data - This data is essential to the Xcenter/Xstore suite functionality. This includes the following:
  - Full organizational hierarchy to support functionality such as rolling out new keyboard configurations by region, etc.
  - Stores including their addresses
  - Full merchandise hierarchy to support Xcenter/Xstore reporting functionalities
  - Differentiators and differentiator groups to support functionality such as looking up a sku by style/color.
- Item - Item information is generated at both the corporate and location level specific files and are sent to the Xcenter/Xstore application. Item information being sent includes the Item header, Item/Location, VAT Item, and Related Item information.

Note: The Oracle Retail Merchandising System interfaces can integrate with other third party Point of Sale applications.

Implementation Rules for Custom APIs

Because different retailers have different rules that they want to apply for their business in order to meet their own specific business processes, RMS supports the ability for retailers to configure custom rules for certain areas of RMS by providing an API that can be used to call a PL/SQL function. This process allows for configuration without modification to base code, which might impact the ability to apply patches or upgrade. Currently, this is supported for Item and Purchase Order approval, as well as for VAT calculations when RMS is configured to run in a simple VAT configuration. Some examples of how this could be used include:

- Ensuring all transaction level items are not approved before reference item has been created for the item.
- Preventing purchase orders from being approved if a factory has not yet been associated with the order.
- Using alternative rates for calculating tax for certain items based on custom flex attributes (CFAS), when running RMS in a simple VAT configuration.

For items and purchase orders, a configurable API function is used. The package function is supplied by the retailer and contains the additional validations that should be performed for items or POs. The name of the function is entered into a configuration table, which allows it to be called by RMS base code. For VAT this works a bit differently, but conceptually supports the same concepts - allowing retailers to configure a process in RMS without modifying base code. The approaches for both are described in the sections below, along with the general guidelines that retailers and integrators should follow.

Item and Purchase Order Approval

In order to configure RMS to use this functionality, records must be added to the CUSTOM_PKG_CONFIG table. This table holds the schema and package/function names defined by a retailer to be called for a particular entity. Each custom function that should be called from base RMS should have an entry in this table. The key fields in this table are:

Table 4–1 CUSTOM_PKG_CONFIG table columns

Column Name	Notes
SCHEMA_NAME	The schema where the custom package function is compiled.
PACKAGE_NAME	This should be the name of the retailer's custom package, if applicable. This should be left blank if the custom code is a function without a package.
FUNCTION_NAME	The name of the retailer's custom function or package function within the package noted above. Only PL/SQL functions are allowed as custom code in this table, either standalone PL/SQL functions or as part of a package. Procedures are not allowed because RMS expects a return value to determine if the execution was successful or not.

Table 4–1 (Cont.) CUSTOM_PKG_CONFIG table columns

Column Name	Notes
FUNCTION_KEY	This value determines which part of RMS the function will be called from. Valid values are: ▪ ITEM_APPROVE_UI - this will be executed for item approval whether items are created in RMS screens or any method of Item Induction (Xitem API, Bulk Upload, manual) ▪ ORDER_APPROVE_RPL - this will be executed when by replenishment processes when approving orders ▪ ORDER_APPROVE_UI - this will be executed when approving orders in the RMS screen ▪ ORDER_APPROVE_POI - this will be executed when orders are approved as part of the Store Orders API or in any method of PO Induction (Xorder, Bulk Upload, manual) Note: In custom order approval, if the retailer wishes to use the same package function for replenishment, UI, Store Orders or PO Induction, three records would need to be inserted into CUSTOM_PKG_CONFIG. The schema name, package name, and function name should be same for each of the three records each of which will bear a different function key value.
CALL_SEQ_NO	This indicates the sequence of the custom code to be executed. This must be set to 1. The calling function would need to be customized when using more than one package function for a particular function key. If retailers have more than one function that needs to be called it is recommended that these be called in the correct sequence from the package function indicated in this table.

Custom Function Definition

The custom function (CALL_CUSTOM_SQL.EXEC_FUNCTION) is called in the RMS code as described above. This package determines which custom code is to be executed by querying the CUSTOM_PKG_CONFIG table based on the function key and the call sequence number passed in. After the custom code schema, package, and function names are retrieved, it calls the custom code via Dynamic SQL and passes the input payload. The inputs required for the custom functions should be as follows:

Table 4–2 Custom Function Inputs

Input	Input/ Output	Type	Comments
O_ERROR_ MESSAGE	In/Out	RTK_ ERRORS.RTK _TEXT	Errors returned from the function must exist on the RTK_ERRORS table in RMS to ensure that messages are properly displayed in standard RMS error handling.
O_ORD_ APPRERR_ EXIST	In/Out	BOOLEAN	This is only used for PO approval package functions. It works with the error message parameter and in the case of any custom validation failures, the parameter should be set to TRUE and the error message parameter should be populated accordingly.

Table 4–2 (Cont.) Custom Function Inputs

Input	Input/ Output	Type	Comments
IO_CUSTOM_ OBJ_REC	In/Out	CUSTOM_ OBJ_REC	This is PL/SQL TYPE object which has placeholders for base RMS to pass the input parameters to the custom code. The calling function will pass the following to this object: ■ Function key ■ Call sequence number ■ Entity ID (e.g. item or order number)

Notes

Consider the following notes:

- The custom function return type must be BOOLEAN. No other data type is allowed.
- In case of fatal errors, the custom function should return FALSE along with an appropriate error message in the output variable.
- Errors for item approval should be coded to write to the item approval errors (ITEM_APPROVAL_ERROR) table, similar to base approval rule violations. This will ensure that the approval rules are applied seamlessly for the end user. The custom package should return FALSE to the calling function only if exceptions occur. If there are no exceptions, but only item approval errors, the package should return TRUE.
- When using the Order Subscription API (Xorder) or the Store Order Subscription API, the errors will be propagated to the calling function. Approval errors encountered during other methods of PO induction (e.g. spreadsheet upload or bulk upload) are logged in an error table, together with the other order creation validation issues.
- For PO approval, the custom package should return TRUE to the calling function when only validation errors are encountered and the O_ORD_APPRERR_EXIST parameter should be set to TRUE.

An example package specification for items is as follows:

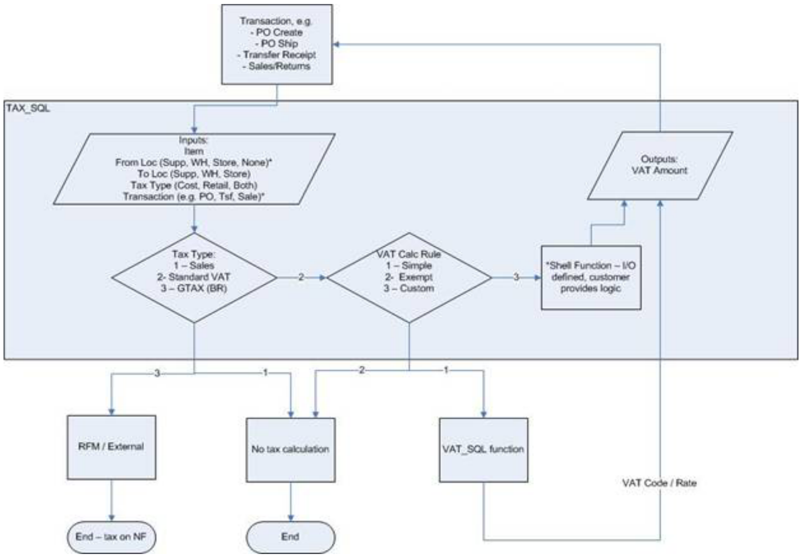
```
CREATE OR REPLACE PACKAGE MY_OWN_ITEM_APPROVAL AS
-------------------------------------------------------------------------------
-- This function checks whether items that are to be approved have comments
FUNCTION CHECK_COMMENTS (O_error_message     IN OUT   RTK_ERRORS.RTK_TEXT%TYPE,
                         IO_custom_obj_rec   IN OUT   CUSTOM_OBJ_REC)
RETURN BOOLEAN;
-------------------------------------------------------------------------------
END MY_OWN_ITEM_APPROVAL;
```

An example package specification for POs is as follows:

```
CREATE OR REPLACE PACKAGE MY_OWN_APPROVAL_RULES AS
-------------------------------------------------------------------------------
-- This function checks whether orders exist.
FUNCTION CHECK_PRESENCE (O_error_message   IN OUT RTK_ERRORS.RTK_TEXT%TYPE,
                         O_apprerr_exist   IN OUT BOOLEAN,
                         IO_custom_obj_rec IN OUT CUSTOM_OBJ_REC)
RETURN BOOLEAN;
-------------------------------------------------------------------------------
END MY_OWN_APPROVAL_RULES;
```

Custom Tax Configuration in RMS

The configurations set at both the system level and the VAT region level determines how RMS calculates tax for various transactions. The following diagram describes this process:



At a system level, the tax calculations in RMS are governed by the system option called Default Tax Type. The tax type options are:

- SALES - which is used for US only implementations and results in no tax calculations within RMS.
- GTAX - which is used for Brazil implementations and assumes the taxes are calculated by an external tax engine via Oracle Retail Fiscal Management (RFM).
- SVAT - which is used for all other implementations of RMS in which one or more locations for the retailer are subject to value added tax (VAT).

For SVAT implementations, retailers can additionally configure the tax processing in RMS by VAT region as follows:

- Simple - VAT Regions defined as having a tax calc type of 'Simple' will have VAT computation based on the existing logic in TAX_SQL that uses rates defined for items to compute VAT on the transaction.
- Exempt - VAT Regions defined having a tax calc type of Exempt will not support VAT computation for transactions occurring in this region. For example, this is what would be used for US locations in an implementation that also contains locations for which VAT is applicable.
- Custom Tax - VAT Regions having a tax calc type of Custom will use Custom Taxation Rules that need to be defined by the retailer. The configuration required to implement this functionality is described in this section.

Note: If only one entity or location is passed into the tax function as described in the diagram above, then the tax calculation function behavior will be based on the tax calculation type of that entity or location. If the transaction involves two entities or locations (e.g. transfers), the tax calculation support will be based on the below matrix.

Table 4–3 Custom Calculation Matrix

Location A*	Location B*	Tax Calculation Support
Simple	Simple	If Entities A and B lie in the same VAT Region, base simple VAT rules used. If different regions, VAT calculated as zero.
Simple	Exempt	VAT calculated as zero
Simple	Custom	Custom function executed
Exempt	Simple	VAT calculated as zero
Exempt	Exempt	No VAT calculation performed
Exempt	Custom	Custom function executed
Custom	Simple	Custom function executed
Custom	Exempt	Custom function executed
Custom	Custom	Custom function executed

* Location could refer to a store or warehouse, or to a supplier, partner or other entity.

> **Note:** For VAT regions flagged as custom, VAT codes and rates are expected to be set up, even though the VAT code may not solely determine the tax behavior in all cases.

Defining Custom Taxation Rules

The CUSTOM_TAX_SQL package has been created for defining custom rules. This package contains the below functions:

- CALC_CTAX_COST_TAX - which is used for cost-based VAT calculations
- CALC_CTAX_RETAIL_TAX - which is used for retail-based VAT calculations
- ADD_CTAX_RETAIL_TAX - which is used for mark-up calculations

The input and output parameters of these functions are same as those passed to the base tax function.

Inputs:

- Item
- From Location
- To Location
- Tax Type – Cost, Retail, or Both
- Transaction ID (e.g. PO number)

Outputs:

- Tax Code
- Tax Rate
- Tax Amount
- Tax Exempt flag (Y or N)
- Error Message

However, the function body for each of these functions will not contain any logic for identification of the applicable tax rates or computation of the tax amount. The logic to achieve this will need to be added by the retailer as part of implementation. The logic written within these functions is not limited to using just the parameters passed as part of the call and can fetch additional data points required to calculate taxes, as needed. Any failures in VAT calculation logic in any of these functions should be configured to return an output error message. This error message must be added to the RTK_ERRORS table or an existing error message from that table could also be used.

Internationalization

For details on the language supported information see, Oracle Retail Merchandising System documentation for the current release.

5

Oracle Retail Trade Management

This chapter is an overview of Oracle Retail Trade Management (RTM).

Master Data

Oracle Retail Trade Management shares the same database with RMS. RTM is enabled in an RMS instance by the following system options parameter settings:

- Simplified RTM Indicator is N
- Import Indicator is Y

RTM requires certain master data maintenance such as outside locations, Freight Type, Freight Size and Standard Carrier Alpha Codes (SCAC). Import-specific data maintenance is required for country, supplier, partners and items. For calculation of duties, fees, and taxes applicable on import merchandise, Harmonized Tariff Schedule (HTS), files must be uploaded or entered into the system.

Landed Cost

Landed cost is the total cost of an item received from a vendor inclusive of the supplier cost and all costs associated with moving the item from the supplier's warehouse or factory to the purchase order receiving location. RTM facilitates the setting up of various cost components, associating them to the purchase orders, calculating the estimated landed costs at the time of purchase order creation. It also facilitates the tracking and booking of the actual costs after the receipt process.

Estimated Landed Cost (ELC) is composed of cost components from the Supplier, Trading Partners, Item and Origin Country, which are brought together during Purchase Order (PO) creation to develop an estimate of costs associated with purchasing a particular item on the current PO.

The components of landed cost are defined using Expenses, Assessments, and Computation Value Bases (CVBs).

Expenses

Expenses are direct and indirect costs incurred in moving a purchased item from the supplier's warehouse/factory to the purchase order receiving location. Expenses should not be confused with up charges, which allow add-on costs from an initial receiving location to a final retail location and are not part of the landed cost.

An example of a direct expense is the packing cost or insurance cost. Charges incurred for clearing and loading goods at the lading port are an example of indirect costs. Expenses are either added to the base inventory value or booked as a separate expense. Expenses apportioned to inventory affect the weighted average cost (WAC) of the item. Expenses can be assigned to a particular country, supplier, or partner. Expenses are tracked at country or zone levels, as the following defines.

Country Level Expenses

Country level expenses track the costs of bringing merchandise from the origin country, through the lading port, to the import country's discharge port. For example, track expenses for a silk blouse from China, through the lading port, Hong Kong, to the discharge port, Los Angeles.

Zone Level Expenses

Zone level expenses track the costs of bringing merchandise from the import country's discharge port to the purchase order receiving location. For example, track expenses for a silk blouse from discharge port, Los Angeles, through to the retailer's New York City warehouse and to the retailer's Chicago warehouse. Costs are different based on the final destination (for example, longer truck route, railroad).

Zones are defined using the Cost Zone dialog (see Cost Management information in RMS documentation). When the zones are created they are used to define expenses at the supplier level (by zone) for default to items.

Assessments

Assessments are the cost components that represent the total tax, fee, and duty charges for an item. Computation formulas and specific fee and tax types contained in the harmonized tariff schedule (HTS) determine most assessments. The harmonized tariff schedule is defined for an import country.

The HTS comprises a hierarchical structure for describing all goods in trade for duty, quota and statistical purposes. The HTS structure is based on the international Harmonized Commodity Description and Coding System (HS), administered by the World Customs Organization in Brussels, Belgium.

There are two components used to track HTS within the system:

- HTS chapter tables
- HTS tariff item tables

The HTS chapter tables are organized by the first one to four digits of the tariff schedule classification. The HTS tariff item tables are organized around the full tariff item number. The length of the HTS heading or chapter is defined by import country along with the display format of the tariff number on the HTS import country setup table. The item level tariff codes do vary among GATT (General Agreement on Trades and Tariffs) countries. Each tariff item has various duty rates assigned to it. Classification of goods in this system must be done in accordance with the country-specific rules. For example, in the U.S. it should be done as per General and Additional U.S. Rules of Interpretation, starting with the a four digit heading level to find the most specific provision and then moving to the subordinate categories. Other assessments, such as taxes, fees, countervailing charges, and anti-dumping charges are also assigned at the tariff item level.

RTM also allows the retailer to set up the quota restrictions imposed by the government on an item. The quota is linked to the HTS classification.

HTS calculations can be performed based on the country of manufacture or the country of sourcing for their products. This determination is made at implementation time and is defined on the System Options table.

Computation value bases (CVB) are mechanisms used to create a compound expense or assessment. CVBs allow expenses to be a certain percentage of a group of other expenses and assessments rather than just of one value.

Purchasing

If RTM is enabled in an RMS instance, then import orders can be created in RMS. The purchase order dialog provides additional import functionalities along with standard PO information, if the import purchase order indicator is checked. Throughout each step of the import purchase order creation, there are several options available to capture additional information-specific to ordering imported merchandise.

The menu options within the PO Header Maintenance window are used to attach shipping, letter of credit, order dates, attributes, required documents, and timeline information. The menu options within the PO Item Maintenance window are used to attach HTS, required documents, and timeline information for line items on the PO. The PO Item/Location window provides the option to assign expenses at the Item/Location level. All of the expense components are brought together in one view in the Order Item Expense Maintenance window.

Letter of Credit

A letter of credit (LC) is a bank instrument used by most retailers to finance purchase of imported goods. LCs are a preferred method of international settlement because the conditions of the purchase, such as required documents and special instructions are detailed in the text of the LC and reviewed by the bank for compliance before the release of payment. They provide importers with a secure method to pay for merchandise and vendors with a secure method to receive payment for merchandise. Letters of credit can be created and applied to purchase orders. Activity against the letter of credit can also be tracked. When the goods are shipped and shipping documents are available, the seller presents documents at the desk of the advising bank, and if there are no discrepancies, collects the specified payment amount. The advising bank debits the issuing bank for the amount of the negotiation as well as any charges, and the issuing bank makes payment and notifies the retailer of the amount of the principal draw down and associated charges.

RTM also supports the exchange of letter of credit (LC) information with a bank in the internationally recognized Society for Worldwide Interbank Financial Telecommunications (SWIFT) format.

Transportation

The transportation functionality in RTM provides a facility to track information from trading partners as merchandise is transported from the manufacturer through customs clearance in the importing country. This information is recorded at various levels: vessel/voyage, bill of lading (BOL), container, PO/item, and commercial invoice. Transportation information is most often received through EDI. When entered, the information is tracked and edited, as it changes.

The Transportation module is used to track the following information: Shipments, deliveries, licenses and visas, claims, missing documents, packing, dates, timelines, commercial invoices, inland freight, and totals.

When a transportation record is complete, the record is finalized. Finalized transportation records are used to automatically create customs entries. The goods are then tracked as they move through customs.

Customs Entry

The Customs Entry module manages entries of merchandise for clearance through customs while providing the information required for government documentation and reporting. Government duties, taxes and fees are calculated in accordance with the country of import. Entry information is prepared for transmission to the customs broker for entry submission. The Customs Entry module provides the ability to track the arrival of a shipment at customs, the customs clearance or associated delays, customs exams, and entry and liquidation payment amounts.

When the charges and assessments are complete, the retailer can choose to allocate the costs to the actual landed cost module. When the customs entry is complete, the custom entry record needs to be confirmed. Non-merchandise invoices are created automatically from confirmed customs entries in the Oracle Retail Invoice Matching application.

Obligations

As an item progresses through the import process, bills are received from various service providers. As non-merchandise invoices are received from trade partners and suppliers, they can be recorded in the obligations module. They are allocated over the shipment, orders and items that they cover, and the system proportionally allocates the charges to the line item level. Approving an obligation allocates the costs to the actual landed cost module. Approving an obligation creates a non-merchandise invoice in approved status in the Oracle Retail Invoice Matching application.

Actual Landed Costs

RTM helps to track the actual landed cost incurred when buying an import item. The module reports variances between estimated and actual landed costs by cost component and shipment. The Actual Landed Cost (ALC) dialogs provide the flexibility to view information about actual and estimated landed costs for any item on a purchase order or invoice obligation. The information can be organized by obligation, obligation cost component, shipment, or location. Duties, fees and taxes are posted directly to the Actual Landed Cost module bypassing the obligations maintenance process. While finalizing the ALC at a purchase order level, the user can also choose whether or not to update weighted average cost (WAC) for each purchase order. For those circumstances where goods might be partially or completely processed through a distribution center to another location prior to assembly and calculation of ALC, it might be more accurate to have WAC be determined by ELC to stand rather than updating it with an ELC/ALC variance.

RTM Integration with Other Applications

This section describes RTM integration with other application.

Integration with RMS

Oracle Retail Trade Management (RTM) and RMS share the same database schema. When RTM is enabled in an RMS instance, certain import-specific data maintenance is required for country, supplier, partners and items. These are directly updated into the RMS database and subsequently used in RTM.

Integration with Oracle Retail Invoice Matching

RTM, RMS, and ReIM share the same database schema.

Customs Entry Record - Confirmation of the customs entry record inserts an approved non-merchandise invoice record into the Invoice Matching staging tables for extract and uploads into ReIM.

Obligations Entry - Approving an obligation inserts an approved non-merchandise invoice record into the Invoice Matching staging tables for extract and upload into ReIM.

Integration with External Partners

RTM automates the international import transaction data. Four components of RTM: Customs entry, harmonized tariff schedule, letter of credit, and transportation have batch-processing modules that facilitate the flow of data between RTM and external applications and files.

Sharing the Letter of Credit Data with the Trading Partners

Letter of credit batch modules process letter of credit applications and amendments to banks, and upload confirmations, drawdown notifications, and related information from banks. Letter of credit batch module downloads and uploads data in an internationally recognized standard format called SWIFT (Society for Worldwide Interbank Financial Telecommunications).

The batch program LCADNLD extracts approved letter of credit applications to banks. The LCMT700 Perl script converts the LC applications from an RTM file format to the SWIFT (MT 700) format.

The issuing bank sends the retailer a confirmation when it agrees to stand as guarantee for the LC. The LCMT730 Perl script converts letter of credit confirmations from a SWIFT format (MT730) to an RTM flat file format. The batch program LCUPLD uploads the converted data from the table to the RTM database tables.

The issuing bank informs the retailer when credit draw downs are made against the LC or when bank charges the retailer the bank fees. The LCMT798 Perl script converts draw downs and bank fees data from a SWIFT file format to an RTM format. The batch program LCUP798 uploads the converted data from the table to the RTM database tables.

At times after a LC has been issued and confirmed, the retailer might want to make amendments to the LC. An amendment to a LC is valid when all parties involved agree to the new conditions and the agreement has been registered. Retailer initiated amendments are extracted by the batch program LCMDNLD. The LCMT707 Perl script converts the LC Amendments from an RTM file format to the SWIFT (MT707) format.

Integration with Customs Broker

The batch program CEDNLD (customs entry download) extracts custom entry information from the RTM database to custom brokers. Custom Entry (CE) transactions that are in a Sent status are written into a flat file. One flat file is written per broker. Information contained in the files include: order items, bill of lading/airway bill information, shipment information, container information, license, and visa information, broker charges and missing documents information.

The output file could be FTP to the broker. This process has to be handled during implementation and is outside the scope of RTM.

Upload of HTS Data

Customs agencies provide HTS data, this information must be converted to the RTM HTS upload file format for uploading into the system. The RTM batch program TRANUPLD uploads the data from the file into the RTM/RMS database. The program handles both the initial HTS information load as well as mid-year HTS updates that are supplied by the import country's government. The initial upload is handled by inserting information from the file into the tables. HTS information already available in the tables is handled by adjusting the effective dates of the existing HTS records and inserting a new set of HTS records into the tables. Files provided by the import country's government includes HTS Chapters, HTS classification, HTS Tax, HTS Fee, HTS OGA, Tariff Treatments, Tariff Treatment Exclusions.

The files provided by the import country government must be placed in the proper input directory after it is converted, so that the batch program TRANUPLD can pick up the file and process the data.

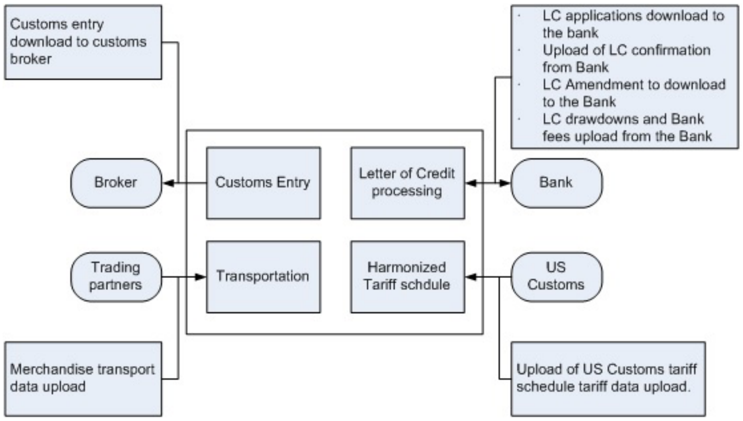
Integration with Supply Chain Partners

As the merchandise moves from manufacturers' warehouse/factory to the discharge port at the import country, the supply chain partners share information with the retailer in a specified file format. The batch program TRANUPLD uploads the information provided in the files into the transportation tables in RTM/RMS database, thus providing online visibility of the merchandise in the supply chain.

The files provided by the supply chain partners should be placed in the proper input directory, so that the batch program TRANUPLD can pick up the file and process the same. In case the files provided by the supply chain partners differ from the expected file format, custom program must be written to convert the data from the original file format to the required file format.

The following diagram illustrates the interaction between RTM and external agencies.

Figure 5–1 Interaction between RTM and External Agencies


Customs entry download to customs broker
· LC applications download to the bank
· Upload of LC confirmation from Bank
· LC Amendment to download to the Bank
· LC drawdowns and Bank fees upload from the Bank
Broker
Customs Entry
Letter of Credit processing
Bank
Trading partners
Transportation
Harmonized Tariff schdule
US Customs
Merchandise transport data upload
Upload of US Customs tariff schedule tariff data upload.

User Setup and Security

As RTM and RMS share the same instance, the application shares the security framework of RMS. The user setup, role assignment and access permission are done in the RMS framework.

For details on the user and security setup in RMS, refer to the *RMS Users and Security section* of this guide.

Simplified RTM Configuration

RTM is a simplified version of Oracle Retail product suite targeted at mid-tier retailers. The simplified Oracle Retail Merchandising Operations Management applications support basic retail processes needed by a mid-tier retailer. Advanced features are turned-off through system parameters, with the goal to reduce implementation complexity and enabling faster implementation and lower total cost of ownership.

If the system option parameter Simplified RTM indicator is enabled, then the following RTM functionality is not available in the application:

- Setting up RTM-specific master data such as freight type, freight size, and standard carrier alpha codes (SCAC)
- Letter of credit functionality
- Transportation functionality
- Customs entry functionality
- Obligation maintenance
- Actual landed costs

If both the Simplified RTM indicator and the Import indicator are enabled, then some import-related functionality is available in RMS. With this set up, the retailer has the option to set up the HTS data and use it in the purchase order process. The retailer can also choose letter of credit as a payment option in the Purchase Order header level, but all other related LC functionality is not available. It is assumed that the retailer is using some other external system for LC processing.

If the import indicator is not enabled, then no RTM functionality is available in the application. For additional information on setting the value in a System Configurations table, see *Oracle Retail Merchandising System Installation Guide*.

Other Features

As RTM is related to import purchases and might use foreign currency, it is important that the current exchange rate between the primary currency and the supplier's currency is maintained in the application. RMS does maintain currency exchange rates and can accept currency updates.

6

Oracle Retail Sales Audit

This chapter is an overview of Oracle Retail Sales Audit (ReSA).

Information Maintained by ReSA

The following information is maintained in the ReSA application.

System Options

Oracle Retail Sales Audit (ReSA) contains a set of system options that is different from the RMS system options set. These system options control functionality in ReSA. One of the important system option parameters is the balancing level. The balancing parameter could be Cashier or Register. If the balancing parameter is Cashier, the totals are balanced for each cashier. If the balancing parameter is Register, the totals are balanced for each register. Another important option is the unit of work. This determines whether transactions are exported prior to the elimination of all errors for the store/day. Other parameters include defining the escheatment party and the details of the clearinghouse. Certain business rules like duplicate and missing transaction number checks are also controlled through the system options setting.

Foundation Data

ReSA requires a certain amount of foundation data to start working. ReSA shares some of its foundation data with RMS, while the rest of the foundation data needs to be maintained in ReSA.

Foundation data in ReSA includes company/location close dates, location traits, bank setup, tender types, reference maintenance, error codes and store-specific foundation data.

Totals

ReSA allows the retailer to define the Totals needed for their business. Totals in ReSA can be a ReSA-calculated value from raw transaction data or a total that comes from the POS/OMS through the RTLog. ReSA calculates a total based on raw transaction data or on existing totals. Totals are used for performing store balancing over/short analysis within ReSA. Totals are also used to create data for external systems such as General Ledger. By assigning an audit rule to the total, business validation is built in ReSA. For example, the retailer defines a ReSA-calculated total of total cash tendered at a register for a store/day. He defines another total, a POS/OMS-declared total, for the total cash declared at the register for a store/day. He defines an audit rule to compare both the totals for over/short analysis. Totals are used to export consolidated data to an external system. So, while defining a total, the retailer defines the systems to which the total is exported.

Audit Rules

Rules are used in ReSA to perform custom data validation against transactions and totals. Audit rules run on POS/OMS transaction data and totals during the automated and interactive audit process. When the rule is broken, an exception error is thrown that must be edited or overwritten by the auditor. ReSA comes with some standard pre-configured rules and also provides flexibility to the retailers to define rules needed for their business.

Error Codes

Error codes provide information to the auditor about the type of error and the remedial action needed to solve the error, thus increasing the effectiveness of the audit process. When setting up audit rules for automated audit process, an error code is associated to the rule. The error code contains an explanation of why the rule failed, as well as a recommended solution. The error code also contains security settings to determine if the error can be overridden at the store or headquarters. When a rule fails, the error code is available to the auditor in the error list. Additionally the specific location where the error has occurred is available in the transaction details, thus helping the auditor to quickly understand and correct the error.

Automatic Audit Process

Automatic auditing is done in ReSA using batch programs. The goal of the automatic audit process is to accept transaction data from point-of-sale (POS/OMS) applications and move the data through a series of processes that culminate in clean data. ReSA uses several batch-processing modules to perform the following activities:

- Import POS/OMS transaction data from the RTLog.
- Perform initial validation of data during upload of data from the RTLog to ReSA database.
- Produce totals using user-defined totaling calculation rules that are user reviewable during the interactive audit.
- Validate transaction and total data with user-defined audit rules and generate errors whenever data does not meet the criteria. The user reviews these errors during the interactive audit.
- Create and export files of clean data in formats suitable for transfer to other applications.
- Update the ReSA database with adjustments received from external systems on previously exported data.

Interactive Audit Process

Auditors use the interactive audit process to view and correct errors. This process is done after the automated audit is completed. This process allows the auditors to view errors at summary or detail level, fix or override errors, update the totals and close the store/day. In addition, it is also possible to review and edit data of missing transactions or transactions that have passed automated audit, add transactions and delete invalid or missing transactions.

Summary Views

ReSA provides summary views for the auditor to identify and fix the problem areas quickly. A Store/Day Summary View screen contains the following information about a store/day: The audit status, the data status, the number of transactions, the number of errors and the over/short amount. The auditor has access to other options in the Store/Day Summary screen such as: The error list, the balancing level summary, over/short, miscellaneous totals, missing transactions and the import/export log.

ReSA provides two other summary views: Tender Summary and Item Summary.

Automated Clearing House (ACH) Processing

Automated Clearing House (ACH) is a U.S.-based banking network used to electronically transfer funds. Retailers use ACH to enable them to have access to funds before the funds have been physically deposited in the bank. This is done by estimating the following day's bank deposit and sending this amount to the consolidating bank through the ACH network. In this way, the cash to be received from the stores is hedged.

Escheatment Processing

Escheatment is the process of forwarding monies of outstanding, non-expiring vouchers to the proper government authorities (state or country) after a defined period of time from the date of issuance. Some government authorities require that unredeemed vouchers be escheated after a specific period of time has passed. When a voucher is escheated, an invoice is generated that initiates payment of the escheated voucher amount to the government authority. The government authorities then attempt to locate the consumers owed the monies.

To accommodate Escheatment, a new total should be added to Sales Audit to create escheatment totals. ReSA automatically totals sales transactions based on calculation definitions set up for the total.

Audit Trail

The audit trail functionality provides the store and headquarters employees with the capability of tracking all changes to transactions and totals. ReSA maintains versions of all modified transactions thus enabling easy tracking of changes.

Totals for General Ledger that are impacted by a revised transaction are reversed and both the reversal and the new total are extracted for the General Ledger.

Reporting

ReSA provides some basic online reporting that allows users to view sales audit data that is in the system. The following reports are available in ReSA:

- Credit Card Summary report
- Flash Sales report
- Flash Totals report
- Voucher Activity report
- Operational Insights OBIEE report

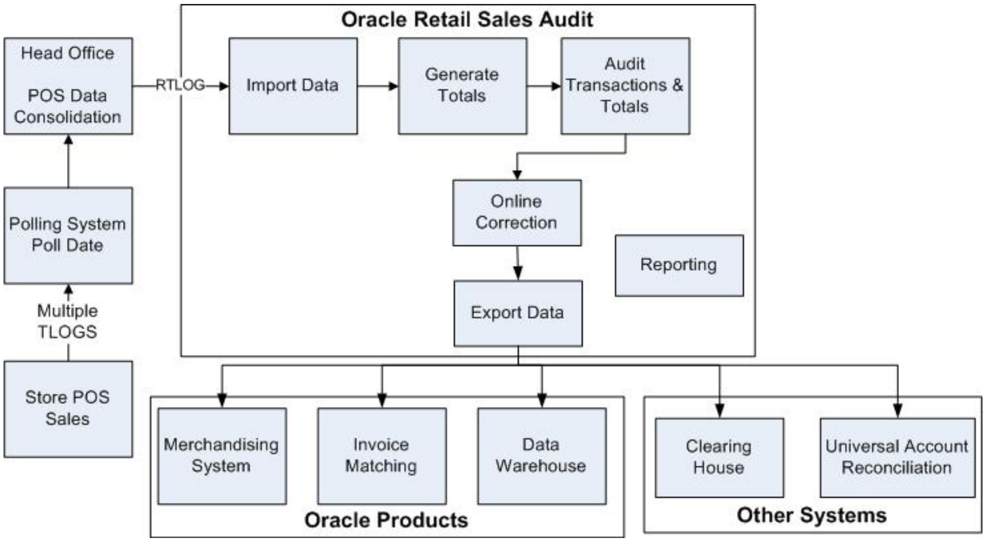
Integration with Other Applications

Oracle Retail Sales Audit interfaces with the following Oracle products:

- Oracle Retail Merchandise System
- Store Inventory Management
- Oracle Retail Invoice Matching System
- Retail Analytics

This figure displays the ReSA interfacing products:

Figure 6–1 Oracle Retail Sales Audit Interface



Oracle Retail Sales Audit also interfaces with the following third-party applications:

- Automated Clearing House.
- Universal Account Reconciliation System.

Integration with Oracle Retail Merchandising System

ReSA and RMS share the same database. ReSA shares some of its master data with RMS. Foundation data such as items, stores, company/location close dates, location traits, bank setup, and tender types are maintained in RMS and used in ReSA.

Current reference data is retrieved from RMS into ReSA by the batch program SAGETREF. The data is extracted into multiple data files. The data in the files are used by the batch program SAIMPTLOG as reference data for doing validation checks on the POS/OMS transaction data during the data upload from POS/OMS to ReSA. Having the reference in data file formats increases the performance of the SAIMPTLOG process. SAGETREF generates the following reference files: Items, Wastage, Sub-transaction level items, Primary variant relationships, Promotions File, Currency File, Warehouse File, Inventory Status File, Variable weight PLU, Store business day, Code types, Error codes, Store POS, Tender type, Merchant code types, Partner vendors, Supplier vendors, Employee IDs, Banner IDs.

All clean and audited sales and returns data is extracted from ReSA into a POSU file by the batch program SAEXPRMS. All corrected sale and return transactions that do not have RMS errors are extracted into the file. The sales audit system options parameter work unit controls the export of data into file in case of presence of RMS errors in the POS/OMS transaction data. The shell scripts UPLOADSALES.KSH and SALESPROCESS.KSH load data from the POSU file into the RMS tables.

Integration with Xstore Point-of-Service and Order Management System

Sales, returns, and other transaction data are loaded into ReSA from a Point-of-Service application, such as Oracle Retail Xstore, for in-store transactions and an Order Management System, for ecommerce orders fulfilled from non-store locations. These applications should provide a standard RTLog file. The RTLog data is loaded into ReSA either in trickle mode or once a day. If the data is uploaded in trickle mode, then corporate inventory reflects a more accurate intra-day stock position. The data from the RTLog is loaded into ReSA using the batch program SAIMPTLOG for end-of-day. If trickle processing is used the final RTLog for the day must include a count of all RTLog files for the store/day.

Integration with Oracle Retail Invoice Matching

In the normal course of business, payments are made to vendors at the store level. Payments for merchandise purchases done at store level are booked against a corresponding merchandise invoice. Payments of non-merchandise purchases or miscellaneous services availed at the store are booked against a corresponding non-merchandise invoice. These transactions are passed from the POS/OMS to ReSA as specially designated PAID OUT transactions (sub-transaction type of EV - Expense Vendor or MV - Merchandise Vendor). All these invoices are assumed paid. The batch program SAEXPIM transfers the PAID OUT type of transactions to the Invoice Matching staging tables.

The batch program SAEXPIM is also used for escheatment processing. Unclaimed monies of outstanding, non-expiring vouchers are totaled after a defined period of time from the date of issuance of the voucher and posted to the Invoice Matching staging tables as a non-merchandise invoice by SAEXPIM. The unclaimed amount is paid out as income to the issuing Retailer. In some U.S. states, it is paid out to the state (based on configuration). ReSA determines who receives this income and accordingly posts a non-merchandise invoice for the partner. These invoices are assumed not paid.

The batch job EDIDLINV is used to extract the invoices from the Oracle Retail Invoice Matching (ReIM) staging tables and load as EDI invoices to ReIM.

Integration with Oracle General Ledger

The batch program SAEXPGL transfers the sales data from ReSA into the financial staging tables in RMS. This batch program executes only if the external financial system is set to O (for other), E-Business Suite, or some other external financial application based on system options configuration.

SAEXPGL directly inserts the data into financial stage tables of RMS.

Integration with Automated Clearing House

ReSA determines the estimated bank deposit for each store/day. The batch program SAEXPACH posts the store/day deposits into a database table and creates the standard ACH format file. The output file is sent to a Clearing House. The output file conforms to the requirements imposed by the National Automated Clearing House Association (NACHA). The nature of the ACH process is such that as much money as possible must be sent as soon as possible to the consolidating bank. Any adjustments to the amount sent can be made front-end. This batch assumes that there is only one total to be exported for ACH per store/day.

Integration with Universal Account Reconciliation Solution

The batch program SAEXPUAR extracts specified TOTALS to a flat file that is interfaced to an account reconciliation application. For each store day, all specified totals are posted to their appropriate output files. All the stores and totals with usage type starting with UAR are exported.

Integration with Store Inventory Management

ReSA sends audited data to SIM which helps SIM to be synchronized with RMS with respect to inventory levels. The batch program SAEXPSIM extracts information from transactions with changed item quantities, item deletions, item addition, change in item status to VOID, and change in transaction status to PVOID. This information in the form of a file (SIMT) is uploaded and processed by SIM.

User Setup and Security

Access to ReSA and access to stores an auditor can take action against, is controlled by privileges available to the user's application role and RMS data level security.

To access ReSA, the user must be mapped to one of the available ReSA roles. There are four default application roles:

- Sales Auditor
- Sales Auditor Manager
- Finance Manager
- Administrator

Each of these roles have multiple privileges attached to them. The privileges are defined at granular level and restrict access to a particular screen, or access to take specific audit action, or access to view specific fields. The privileges get directly assigned to the user once a user is mapped to the ReSA roles. A user can be assigned multiple roles.

ReSA shares the database instance with RMS. RMS location level data security is also applied to ReSA users. The location level security is assigned to every ReSA user. This controls the Organization Hierarchy locations the user has access to.

> **Note:** RMS also supports Item and Merchandise Hierarchy level data security to restrict user access to specific products. Item level security is not applicable for ReSA.

ReSA Accessibility Mode

Accessibility involves making your application usable for persons with disabilities such as low vision or blindness, deafness, or other physical limitations. This means creating applications that can be used without a mouse (keyboard only), used with a screen reader for blind or low-vision users, and used without reliance on sound, color, or animation and timing. Application Development Framework (ADF) Faces user interface components have built-in accessibility support for visually and physically impaired users. User agents such as a web browser rendering to nonvisual media such as a screen reader can read component text descriptions to provide useful information to impaired users.

ADF Faces provides two levels of application accessibility support:

- **Default**: By default, ADF Faces generates components that have rich user interface interaction, and are also accessible through the keyboard. Note that in the default mode, screen readers cannot access all ADF Faces components. If a visually impaired user is using a screen reader, it is recommended to use the Screen Reader mode.
- **Screen Reader**: ADF Faces generates components that are optimized for use with screen readers. The Screen Reader mode facilitates the display for visually impaired users, but will degrade the display for sighted users (without visual impairment).

Additional fine grained accessibility levels as described below are also supported:

- **High-contrast**: ADF Faces can generate high-contrast–friendly visual content. High-contrast mode is intended to make ADF Faces applications compatible with operating systems or browsers that have high-contrast features enabled. For example, ADF Faces changes its use of background images and background colors in high-contrast mode to prevent the loss of visual information. Note that the ADF Faces high-contrast mode is more beneficial if used in conjunction with your browser's or operating system's high-contrast mode. Also, some users might find it beneficial to use large-font mode along with high-contrast mode.
- **Large-fonts**: ADF Faces can generate browser-zoom–friendly content. In default mode, most text and many containers have a fixed font size to provide a consistent and defined look. In large-font mode, text and containers have a scalable font size. This allows ADF Faces both to be compatible with browsers that are set to larger font sizes and to work with browser-zoom capabilities. Note that if you are not using large-font mode or browser-zoom capabilities, you should disable large-font mode. Also, some users might find it beneficial to use high-contrast mode along with the large-font mode.

ReSA provide the ability to switch between the above accessibility support levels in the application, so that users can choose their desired type of accessibility support, if required. ReSA exposes a user preferences screen in which the user can specify the accessibility preferences/mode which will allow the user to operate in that mode.

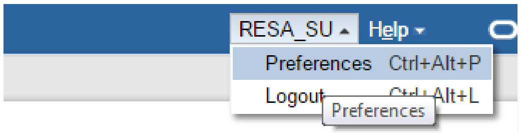
Configuring Application for Screen Reader Mode

Users can configure their session to the accessibility mode by setting user preferences on the home page of the ReSA application as shown below.

Perform the following procedure to configure a user preference for screen reader mode.

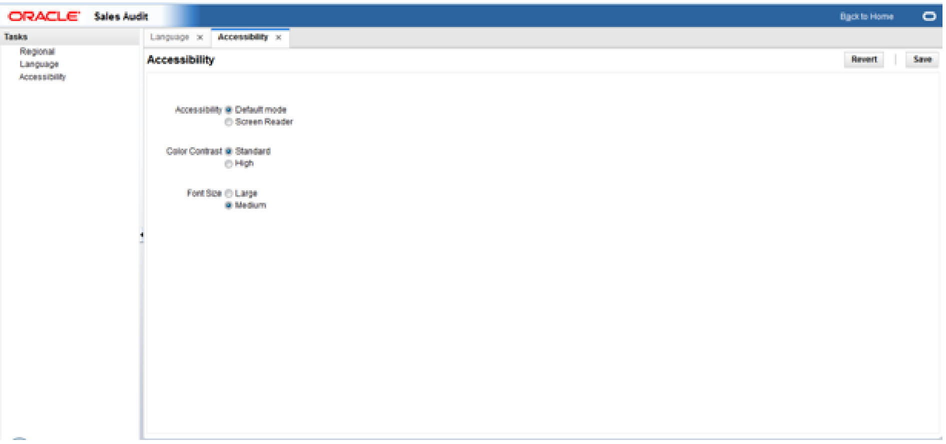
1. Log into the ReSA Portal.
2. From the ReSA application home page, select **Preferences** from the logged in user menu.

Figure 6–2 Logged in User Menu



3. From the Tasks pane, click **Accessibility** to open the Accessibility tab.

Figure 6–3 Accessibility Tab



Accessibility

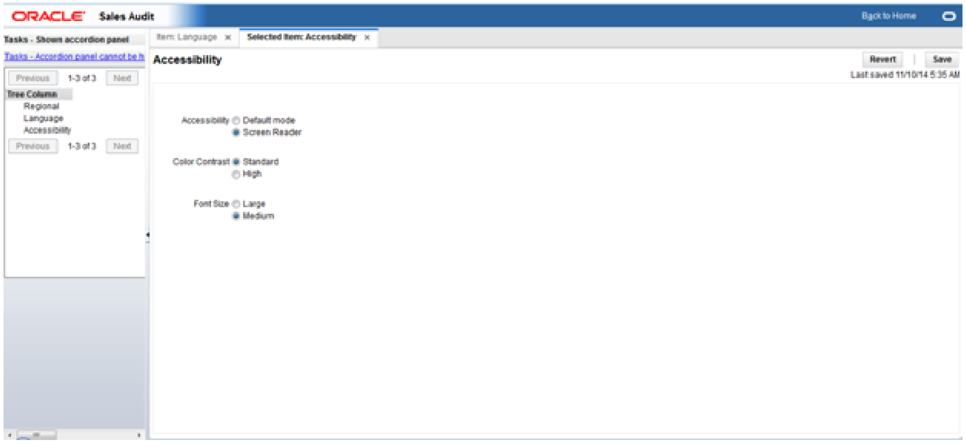
Accessibility ● Default mode
○ Screen Reader

Color Contrast ● Standard
○ High

Font Size ○ Large
● Medium

4. Select **Screen Reader** to enable accessibility mode and click **Save**.

Figure 6–4 Enabling the Screen Reader



Accessibility

Last saved 11/10/14 5:35 AM

Accessibility ○ Default mode
● Screen Reader

Color Contrast ● Standard
○ High

Font Size ○ Large
● Medium

5. Click on **Back to Home** to return to the ReSA home page.

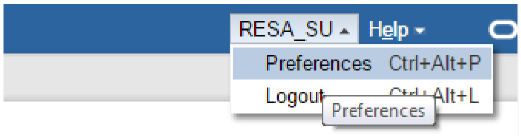
Now the application is enabled in the screen reader mode to assist a visibly challenged user. Some of the graphical content of Operational Insights is also displayed in a tabular mode.

Setting Accessibility to Default

Perform the following procedure to set Accessibility mode to Default mode.

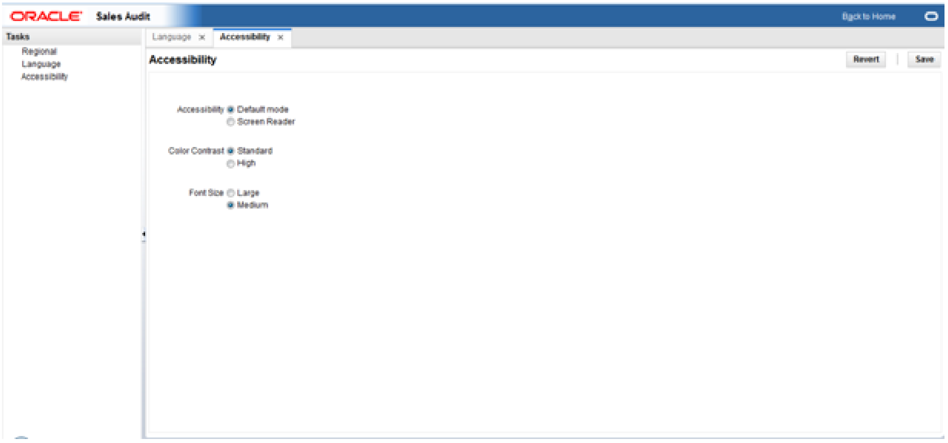
1. From the ReSA application home page, select Preferences from the logged in user menu.

Figure 6–5 Logged in User Menu



2. From the Tasks pane, click Accessibility to open the Accessibility tab.

Figure 6–6 Accessibility Tab



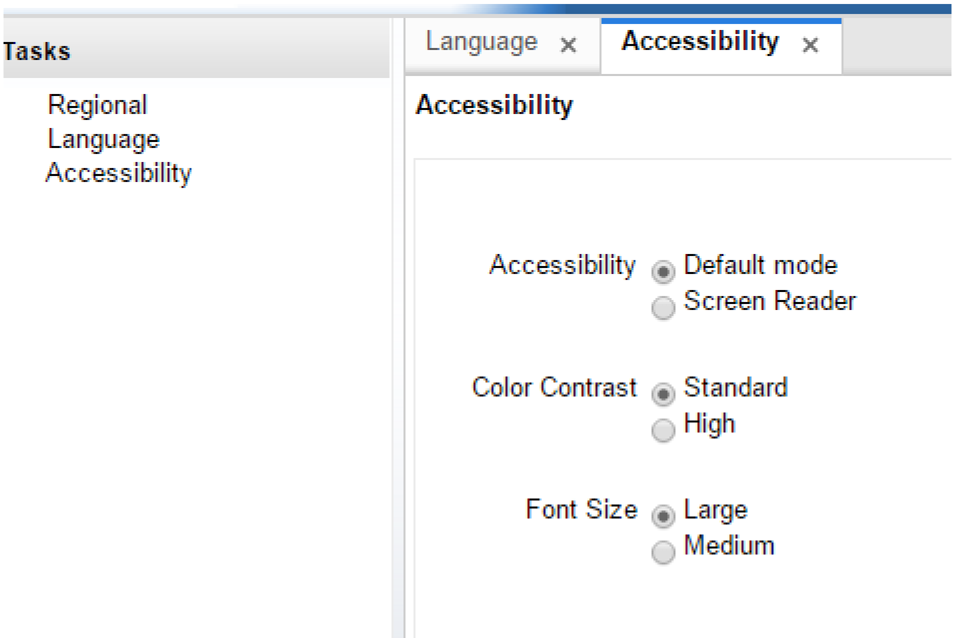
Accessibility ● Default mode ○ Screen Reader

Color Contrast ● Standard ○ High

Font Size ○ Large ● Medium

3. Select Default mode.

Figure 6–7 Accessibility Settings



Accessibility ● Default mode ○ Screen Reader

Color Contrast ● Standard ○ High

Font Size ● Large ○ Medium

4. Click **Save** to save the settings.

5. Click **Back to Home** to return to the ReSA home page.

   ReSA is returned to default mode and you can view graphical reports.

7 Oracle Retail Operational Insights

This chapter provides an overview of Oracle Retail Operational Insights (OI).

Overview

Oracle Retail Operational Insights dashboards and reports provide pervasive Business Intelligence. To provide a seamless user experience, they are designed to be embedded in a single sign on environment within Retail Merchandising Fusion Application like ReSA/ReIM/Allocation.

Operational Insights dashboards and reports have integrated data sources, including Oracle Retail Merchandising System (RMS) for foundation data and ReSA/ReIM/Allocation for transactional data from their corresponding applications. The standard reporting tool for Operational Insights dashboards and reports is Oracle Business Intelligence Enterprise Edition (OBIEE).

Operational Insights Reports and ADF Applications

Using OBIEE11g, a rich metadata layer is created hiding the complexities of the base transactional systems

The reports and dashboards are created using OBIEE Answers and embedded into Retail Merchandising Fusion Application ADF screens. The Oracle Retail Operational Insights dashboards and reports provide the ability to drill to lower level of detail and launch to application screens to execute tasks thus enabling actionable, and close loop BI.

The contextual BI reports in ADF application screens provides additional contextual insights related to the primary information displayed in the main dashboard report or in the contextual pane of application screen. The pervasive contextual BI reports give an user additional insight into the data that they see in a dashboard or application UI screen to make better decisions and drive efficiencies in the process.

The contextual BI reports for application screens other than the main dashboard screen appear in an expandable pane to the right of the main application screen.

Setup and Configuration

This section covers the following topics:

- Sizing Information
- Factors to Consider for Sizing
- Reporting Scenarios

Sizing Information

This section provides a list of factors that should be taken into account when making sizing plans. There are two major hardware components that make up the Operational Insights dashboards and reports physical environment:

- Middle Tier Application Server - The middle tier application server hosts software components such as the Oracle WebLogic Server and Oracle Business Intelligence Enterprise Edition (EE).
- Database - The Oracle Database stores large amounts of data that are queried while generating Oracle BI reports. The report query processing is heavily dependent on the hardware sizing decision. Operational Reports are a light reporting solution which leverages the Retail Merchandising Fusion Application application database schema and directly reports off it.

Sizing is customer-specific. The sizing of the Operational Insights dashboards and reports application is sensitive to a wide variety of factors. Therefore, sizing must be determined on an individual installation basis. The database sizing is driven mostly by the base Retail Merchandising Fusion Application requirements.

Testing is essential as with any large application. Extensive testing is essential for determining the best configuration of hardware.

Factors to Consider for Sizing

The following factors must be considered for sizing:

- Application Server
- Report Complexity - Reports processed through Oracle BI can range from very simple one-table reports to very complex reports with multiple-table joins and in-line nested queries. The application server receives data from the database and converts it into report screens. The mix of reports that you run heavily influence the sizing decision.
- Number of Concurrent Users - The Operational Insights dashboards and reports embedded in Retail Merchandising Fusion Application are designed for multiple concurrent users. When more users are running reports simultaneously, more resources are necessary to handle the reporting workload. For more details on Clustering and Load Balancing, see the *Oracle Business Intelligence Enterprise Edition Deployment Guide*.
- Hardware Architecture
- Number and Speed of Processors - More and faster processors speed report query processing in the database. The application server needs fewer resources than the database.
- Memory - More memory enables a larger SGA in the database and will reduce the processing time for report queries.
- Network - Since the data from the report queries needs to go from the back-end database to the application server, a faster network is better than a slower network, but this is a relatively low priority.

Reporting Scenarios

The Operational reports can be divided into following three categories:

1. The Dashboard Reports - The dashboard surfaces information in a manner that help users like Allocators, sales auditors, finance managers, etc, prioritize their

day-to-day activities as well as quickly identify areas of concern. The dashboard allows you to look at a consolidated view of their to-dos and then directly contextually launch into the corresponding application screens to take an action thus enabling actionable and closed loop BI.

2. Contextual BI Reports - Contextual BI Reports are displayed in the contextual pane of the Retail Merchandising Fusion Application windows. Contextual BI reports provide additional in context insights to the data being viewed in the application screen. This kind of pervasive BI reports enable you to take quicker and more informed actions thus driving workforce efficiencies.
3. Adhoc Reports - Adhoc reporting of Oracle Retail Operational Insights metrics and attributes through OBIEE Answers is supported in 15.0 release. Adhoc reporting privilege through standalone OBIEE Answers is limited to Administrator role in current release. Due to transactional nature of the underlying schema for Operational Insights, the adhoc reporting requirements and their performance impact on base transactional schema should be carefully analyzed by Administrator. Performance tuning of adhoc reports against Oracle Retail Operational Insights metrics and attributes is not supported in current release. For more information on details related to the metrics usage for adhoc reporting, see the *Oracle Retail Operational Insights User Guide*.

For more information, see the *Oracle Retail Operational Insights User Guide*.

> **Note:** The Operational Insights dashboard can be accessed in a standalone OBIEE environment. Use of the Operational Insights contextual reports in standalone OBIEE is not supported due to dependencies on application input parameters.

Security and SSO

Operational Insights dashboards and reports provide pervasive Business Intelligence when implemented in a Single Sign-On environment. To provide a seamless user experience, they are designed to be embedded within the Retail Merchandising Fusion Applications like Allocation, ReIM, and ReSA.

> **Note:** The Single Sign-On (SSO) implementation for Oracle Retail Operational Insights dashboards and reports is mandatory in production environments because it has contextual BI reports and in context launches into application screens apart from dashboard reports. Because accessing the Operational Insights reports from the application UI in the absence of SSO poses a security risk, the use of integrated Operational Insights reports in Retail Merchandising Fusion Application UI without SSO is not supported in this release. In the absence of SSO, the Oracle Retail Operational Insights dashboard can be accessed in a standalone Oracle Business Intelligence Enterprise Edition (OBIEE) environment. The use of the Operational Insights contextual reports in a standalone OBIEE environment is not supported due to dependencies on application input parameters.

For more information on security configuration and implementation, see the *Oracle Retail Merchandising Operations Management Security Guide*.

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. Operational Insights dashboards and reports has been internationalized to support multiple languages.

> **Note:** Operational Insights dashboards and reports use ISO codes for all the supported languages. Oracle Retail Operational Insights dashboards and reports look up RMS language codes. In the case a language supported by Oracle Retail Operational Insights dashboards and reports is not available in the source system, then the Primary language of RMS is used.

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 include the following:

- Graphical User Interface (GUI)
- Data translations for Item description, Store name, Tender Type Group, and item differentiators
- Report titles

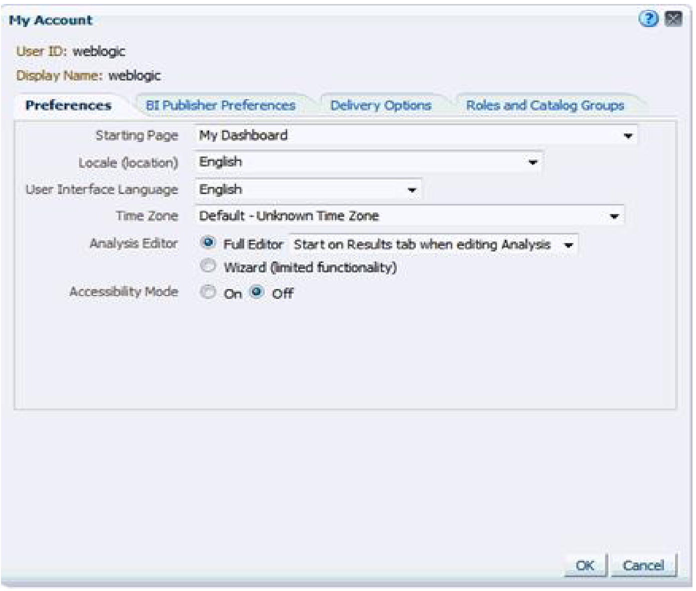
The user interface for Operational Insights dashboards and reports has been translated into:

- Chinese (simplified)
- Chinese (traditional)
- Croatian
- Dutch
- French
- German
- Greek
- Hungarian
- Italian
- Japanese
- Korean
- Polish
- Portuguese (Brazilian)
- Russian
- Spanish
- Swedish

- Turkish

Operational Insights Dashboards and reports displays the reports in the same language as the base Retail Merchandising Fusion Application by accepting the language and locale parameters from the application. The language and locale can be set for a user using the My Account OBIEE preferences as shown below. The language and locale setting in OBIEE My Account preferences is however overwritten by the language and locale parameter passed from the Retail Merchandising Fusion Application once the reports are accessed from within the application (embedded in application).

Figure 7–1 My Account Options



My Account

User ID: weblogic

Display Name: weblogic

Starting Page: My Dashboard

Locale (location): English

User Interface Language: English

Time Zone: Default - Unknown Time Zone

Analysis Editor: ● Full Editor Start on Results tab when editing Analysis
○ Wizard (limited functionality)

Accessibility Mode: ○ On ● Off

Multi-Language Setup

Oracle Retail Operational Insights dashboards and reports are supported in eighteen languages. This section provides details on the various scenarios that one may come across during implementation.

Since multi-language data support in Oracle Retail Operational Insights is dependent on the availability of the multi-language data in the source system, it is important to understand various scenarios one may encounter. Before proceeding review the following facts about multi-language support:

- Operational Insights dashboards and reports programs extract multi-language data from the source systems.
- Depending on the implementation, the source system may or may not have data for a particular supported language(s). For example, RMS supports Item Descriptions in multiple languages but the item's description may not be available in the translated languages. For source system released languages, see the *Oracle Retail Merchandising System Operations Guide*.

Scenario 1

All the supported languages are implemented in Operational Insights dashboards and reports and are the same set of languages that are supported in the source system as well. Multi-lingual data sets are enabled in both Operational Insights and the source Retail Merchandising Fusion Application.

Data Scenario 1a

Translated data exists for all records in Source System: This is an ideal scenario where the source system supports data for the same set of languages as Oracle Retail Operational Insights and data for the required column exists in all the languages in the source system. In this scenario, the attributes that are supported for multi-languages get all the multi-language data in Oracle Retail Operational Insights.

Data Scenario 1b

Translated data does not exist for some of the records in the source system. For the attributes for which data is not available in the source system, Operational Insights dashboards and reports display the attribute in the source system's primary language. For example, Operational Insights dashboards and reports request data in German and English languages. In RMS, the Item Attribute description is not available in the German language but is available in the English language. Operational Insights dashboards and reports display Item description in English to a user who is logged into Oracle BI EE (assuming English is the primary language of RMS for that implementation).

Currency Conversion

Operational Insights supports both primary currency and local currency. The Operational Insights reports support this in the following way:

- Oracle Retail Operational Insights for ReSA - For the following reports where amounts are aggregated across stores, if the auditor has stores assigned that have varied local currencies, the amounts are displayed in primary currency only. If all auditor stores have the same common local currency, the amounts are displayed in local currency.
  - Over/Short Summary Report on Dashboard
  - Over/Short History Report on Store Day Search Window

  For the following reports where amounts are displayed for a single store selected in the ReSA window, the values are displayed in the currency selected in the currency toggle of the corresponding ReSA window (Primary or Local). On window launch, Local Currency is the default.
  - Cashier/Register Over Short Report on Store Day Summary Window
  - Tender Summary Report on Tender Summary Window
- Oracle Retail Operational Insights for ReIM - In ReIM, data is displayed in primary or supplier currency depending on the report.
  - In the supplier detail report, data is displayed in supplier currency
  - In the invoice detail report, data is displayed in primary currency
- Oracle Retail Operational Insights for Allocation - In Allocation, top sales and bottom sales report are displayed in primary currency although the data stored in the database is in the local currency.

Number Format

This is not a user preference in OBIEE. This format is set at report level that is common for all users.

Date Format

The date format is based on the users language/locale and is independent of the date format selected within the Retail Merchandising Fusion Application.

Performance

Operational Insights dashboards and reports is a light weight reporting solution and provides efficient access to operational data through the delivered pre-defined reports and dashboards that are embedded in the application ADF User Interface along with adhoc reporting capabilities. For any transactional reporting solution, including Operational Insights, smart decisions on how to implement and run your operational reporting system ensure that you are getting the most out of it. This section contains information that helps you get the best performance out of Operational Insights and identifies common contributors that can weaken performance, as well as best practices that ensure Operational Insights is running in the most optimal manner. All implementations are unique and the factors that are beneficial for one implementation may not have the same effect for all the implementations. It is a good practice to test several settings/approaches for the factors and recommendations listed below and use the ones that work best for your environment. The factors listed in this chapter are the key factors that impact performance but no absolute values or settings can be provided for implementation purposes due to the uniqueness of each environment.

Operational Insights uses Oracle Business Intelligence (OBIEE) for operational reporting purposes. The recommendations in this section focus on front end (OBIEE) components of Operational Insights dashboards and reports.

Operational Maintenance and Performance Considerations

Operational Insights is an extensible and customizable solution and customers can customize the reports and dashboards in OBIEE. The pre-defined reports and dashboards that are delivered for the different Retail Fusion Applications are meant to be a reference implementation of how customers can leverage the power of BI integrated into the applications. The performance of these reports and dashboards may vary from one implementation to another and the reports can be easily modified to account for any performance concerns at the time of implementation. Customers are advised to consider their data profiles and volumes to decide how to best configure and customize the reports to meet their functional and operational needs and account for any performance issues.

Key Factors in Performance

The following key factors exist:

- Based on the complexity of the report, OBIEE sometimes generates complex SQL, causing the Oracle Database to pick a less than optimized execution plan. In order to avoid this scenario, it is recommended that you enable the SQL Plan Baseline functionality of the Oracle 11gR2 (it is disabled by default). For more details, see the *Oracle 11gR2 Performance Tuning Guide*.
- Operational reporting queries sometimes require retrieval of large amounts of data. Since Operational Insights dashboards and reports are embedded within application window, thus all caching is disabled at installation time. This enables any change to data in application to be immediately reflected in the reports. For example in the Oracle Retail Operational Insights for ReSA application, when one updates the item quantity for a particular store, the status of the store changes from partially loaded to fully loaded. Hence, to reflect this change immediately it is essential that the OBIEE cache is disabled. However if the implementation does

not require real time data updates for effective usage of Oracle Retail Operational Insights then caching can be a powerful performance enhancement technique.

The Oracle BI server can save the results of a query in cache files and then reuse those results later when a similar query is requested. Using the middle-tier cache permits a query to be run one time for multiple requests of a report and not necessarily every time the query is run. The query cache allows the Oracle BI Server to satisfy many subsequent query requests without having to access back-end data sources (such as the Oracle database). This reduction in communication costs can dramatically decrease query response time.

To summarize, query caching has the following advantages only when the same report is run repeatedly:

- Improvement of query performance
- Less network traffic
- Reduction in database processing and charge back
- Reduction in Oracle BI server processing overhead

For more details on caching, see the *Oracle Business Intelligence System Administration Guide*.

- Data Base Configuration/Partitioning strategy/ Purging and Archiving Strategy - Oracle Retail Operational Insights dashboards and reports are built on Retail Merchandising Fusion Application (Allocation/ReIM/ReSA/RMS) database which is based on Oracle Database 12c Enterprise Edition Release 12.1.0.1.4 and must be optimized and configured for a retailers' needs.
- Adequate Hardware Resources
- Database Connections

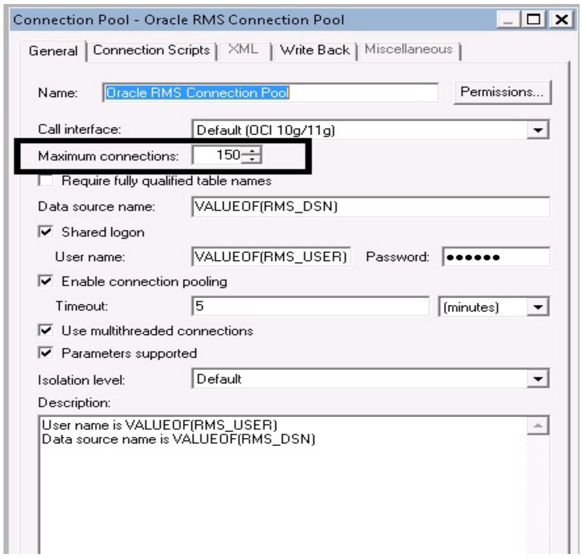
Establishing a optimum number of connections settings to the database is a step that must be completed prior to using any of the Operational Insights reports. The number of concurrent sessions set in the database must be according to the retailer's size and usage of the product. The number of connections within OBIEE rpd for Operational Insights must be set to an optimum.

Set the maximum connections for a Connection Pool to a reasonable level. In general, you can use the rule that one connection consumes about 1 MB of server memory. Note that connections are only started when needed. The maximum connections property specifies how many connections are maintained after they have been started. The time out property tells the Oracle BI Server how long to maintain unused connections in the pool.

It is recommended that you set the maximum connections to 20% of the simultaneous users multiplied by the maximum number of reports open on any given dashboard. For example, assuming you have 100 users and only 20% are going to be logged on at any one time and running reports, and the maximum number of reports on any given dashboard is four, then the maximum connections should be set to 80. Keep in mind that four reports could use more than four connections if any report (Logical SQL) generates multiple physical SQL queries, because one connection is used for each physical SQL statement. For more information, see Setup and Configuration.

For more information on calculating the optimum number of database connections, see the Max Connections Formula for Connection Pools by OBIEE (Doc ID 795846.1).

The maximum connections are set within the rpd connection pool as follows:



Name: Oracle RMS Connection Pool

Call interface: Default (OCI 10g/11g)

Maximum connections: 150

□ Require fully qualified table names

Data source name: VALUEOF(RMS_DSN)

☑ Shared logon

User name: VALUEOF(RMS_USER) Password: ●●●●●●

☑ Enable connection pooling

Timeout: 5 (minutes)

☑ Use multithreaded connections

☑ Parameters supported

Isolation level: Default

Description:

User name is VALUEOF(RMS_USER)
Data source name is VALUEOF(RMS_DSN)

Leading Practices

This section describes leading practices.

Customization

From release 15.0, changes to pre-packaged Oracle Retail Operational Insights dashboards and reports and Operational Insights rpd meta data model are subject to standard customization support policies. Any customizations made should follow an extensive analysis to ensure no performance impacts to transactional system users. Changes and modifications to the Operational Insights delivered code or development of new code is considered customization. Operational Insights does not support custom code developed by clients unless the issue related to customization can be recreated using Operational Insights delivered objects.

The following recommendations should help you in maintaining the Operational Insights code:

- Naming convention - It is recommended that you use a good and consistent naming convention when customizing Operational Insights dashboards and reports code or when building a new code in the Oracle Retail Operational Insights dashboards and reports environment.
- As a best practice, keep all the documentation up-to-date for capturing any changes or new code that has been developed at a site.
- While creating custom reports, refer to the following guidelines:
  - Understand capabilities of adhoc reporting. For more information, see the *Oracle Retail Operational Insights User Guide*.
  - Report developers should be trained in Oracle BI to learn how to design reports in the most optimal manner.
  - Design reports at the highest level possible and allow drill down to more detailed levels when required.
  - Design reports in a manner that allows multiple users to utilize a single report output rather than multiple users running the same report. A best practice is to run one report and distribute that report to multiple users. For more

information on how to distribute reports, see the *Oracle Business Intelligence Enterprise Edition User Guide* and the *Oracle Business Intelligence System Administration Guide*.

- Do not design reports to request data at a level lower than the minimum level that a metric can be reported. In addition, drilling must not be performed at these levels. This ensures that reports do not produce misleading or invalid results.
- Evaluate and purge reports periodically to eliminate any outdated or duplicate reports.
- Design reports to use the least amount of fact areas necessary. This reduces the number of fact table joins and in turn reduces the risk of poor report performance. Try creating separate reports with one or two fact areas on the report at a time and combining the results after these reports have run successfully.
- Design reports with the least number of metrics necessary.
- Schedule reports according to priority. This ensures that critical reports are available when needed. For more information on how to schedule reports, see the *Oracle Business Intelligence System Administration Guide*.

Database Configuration

Oracle Retail Operational Insights dashboards and reports is built on Oracle Database 12c Enterprise Edition Release 12.1.0.1.4 and must be optimized and configured for a retailers' needs.

Oracle BI EE Best Practices

The following are the Oracle BI EE best practices:

- Create aliases for the objects created in the physical layer for usability purposes.
- Any level key on ident's must be set to non-drillable.
- In the presentation layer, fact folders (presentation tables) must contain only metrics and dimension folders (presentation tables) must contain only attributes.
- For a development environment, it is recommended that you use a multi-user environment. For more information on setting up a multi-user environment, see the *Oracle Business Intelligence System Administration Guide*.

Frequently Asked Questions

The following issues may be encountered while implementing Oracle Retail Operational Insights dashboards and reports. The accompanying solutions should help you work through these issues.

Issue:

Why am I getting the Login Denied error with the following message when I try to run a report using Oracle BI Presentation Services?

ORACLE ERROR CODE: 1017, MESSAGE: ORA-01017: INVALID USERNAME/PASSWORD; LOGON DENIED

Solution:

Ensure that the repository connection pool has the right login credentials in the Oracle BI Administration Tool and check the tnsnames.ora file.

Issue:

I am getting the following error when I performed the Update all Row Counts task from the Oracle BI Administration tool.

UNABLE TO CONNECT DATABASE USING CONNECTION POOL

Solution:

Ensure the repository connection pool has the right login credentials in Oracle BI Administration tool or check the tnsnames.ora entry.

Issue:

Why can't I see the query activity at the individual user level in the NQQUERY.LOG file?

Solution:

Check the Logging Level field in the User window in the User tab. If the logging level is set to zero, the administrator may have disabled query logging. Contact the Oracle BI Administrator to enable query logging.

Issue:

When a user logins to the portal, the reports do not show data.

Solution:

Check if the user has stores assigned to him.

Accessibility Mode

Accessibility involves making your application usable for persons with disabilities such as low vision or blindness, deafness, or other physical limitations. This means creating applications that can be used without a mouse (keyboard only), used with a screen reader for blind or low-vision users, and used without reliance on sound, color, or animation and timing. Application Development Framework (ADF) Faces user interface components have built-in accessibility support for visually and physically impaired users. User agents such as a Web browser rendering to nonvisual media such as a screen reader can read component text descriptions to provide useful information to impaired users.

Operational Insights uses OBIEE for embedding dashboards and reports in Retail Merchandising Fusion Application ADF screens. Users need to set the accessibility parameter within the Retail Merchandising Fusion Application. This parameter is then passed to OBIEE which then renders all dashboards and reports in accessibility mode.

ADF Faces provides two levels of application accessibility support:

- **Default:** By default, ADF Faces generates components that have rich user interface interaction, and are also accessible through the keyboard. Note that in the default mode, screen readers cannot access all ADF Faces components. If a visually impaired user is using a screen reader, it is recommended to use the Screen Reader mode.
- **Screen Reader:** ADF Faces generates components that are optimized for use with screen readers. The Screen Reader mode facilitates the display for visually impaired users, but will degrade the display for sighted users (without visual impairment).

Additional fine grained accessibility levels as described below are also supported:

- **High-contrast:** ADF Faces can generate high-contrast–friendly visual content. High-contrast mode is intended to make ADF Faces applications compatible with operating systems or browsers that have high-contrast features enabled. For example, ADF Faces changes its use of background images and background colors in high-contrast mode to prevent the loss of visual information. Note that the ADF Faces high-contrast mode is more beneficial if used in conjunction with your browser's or operating system's high-contrast mode. Also, some users might find it beneficial to use large-font mode along with high-contrast mode. Faces high-contrast mode is more beneficial if used in conjunction with your browser's or operating system's high-contrast mode. Also, some users might find it beneficial to use large-font mode along with high-contrast mode.
- **Large-fonts:** ADF Faces can generate browser-zoom-friendly content. In default mode, most text and many containers have a fixed font size to provide a consistent and defined look. In large-font mode, text and containers have a scalable font size. This allows ADF Faces both to be compatible with browsers that are set to larger font sizes and to work with browser-zoom capabilities. Note that if you are not using large-font mode or browser-zoom capabilities, you should disable large-font mode. Also, some users might find it beneficial to use high-contrast mode along with the large-font mode.

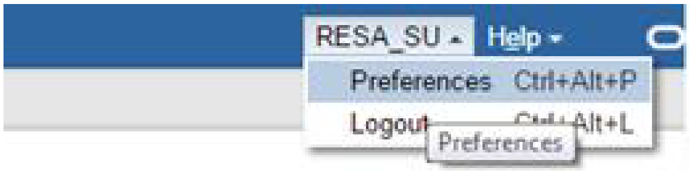
Applications provide the ability to switch between the above accessibility support levels in the application, so that you can choose the desired type of accessibility support, if required. Applications expose a user preferences window in which you can specify the accessibility preferences/mode which allows you to operate in that mode.

Configuring Application for Screen Reader Mode

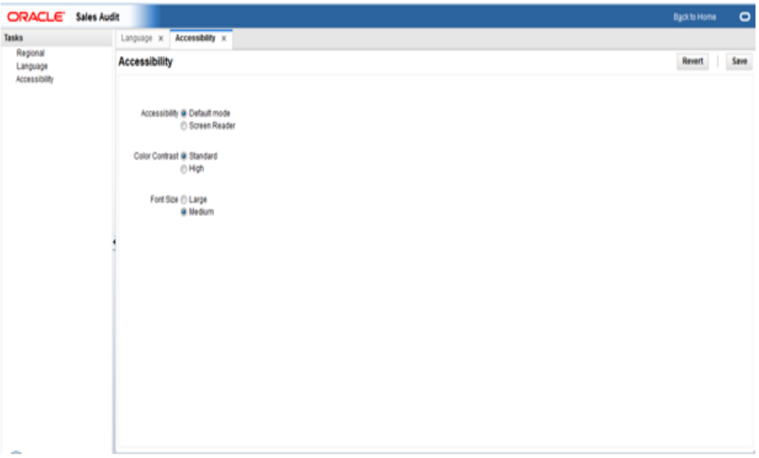
You can configure the session to the accessibility mode by setting user preferences on the home page of the application.

Perform the following procedure to configure a user preference in a screen reader mode:

1. Log into the Application Portal.
2. From the application home page, select **Preferences** from the logged in User menu.



3. From the Tasks pane, click **Accessibility** to open the Accessibility tab.



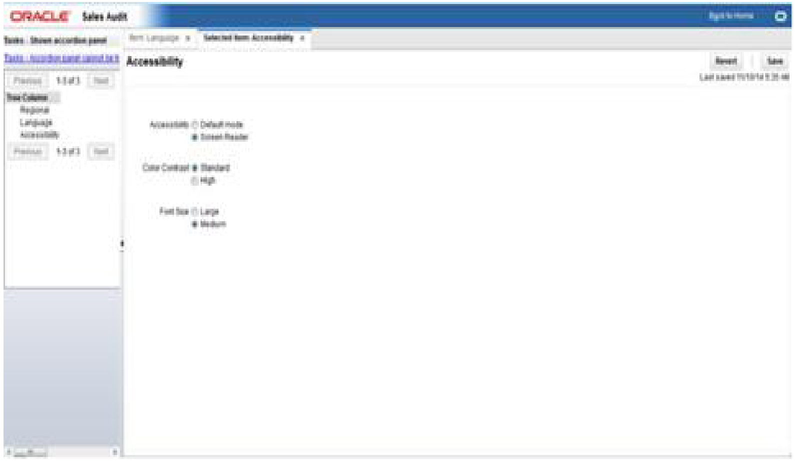
Accessibility

Accessibility ● Default mode
○ Screen Reader

Color Contrast ● Standard
○ High

Font Size ○ Large
● Medium

4. Select **Screen Reader** to enable accessibility mode and click **Save**.



Accessibility

Accessibility ○ Default mode
● Screen Reader

Color Contrast ● Standard
○ High

Font Size ○ Large
● Medium

5. Click **Back to Home** to return to the application home page.

The application is now enabled in the screen reader mode to assist a visibly challenged user. Some of the graphical content of Operational Insights is also displayed in a tabular mode.

Setting Accessibility to Default

To set Accessibility mode to Default mode, take the following steps:

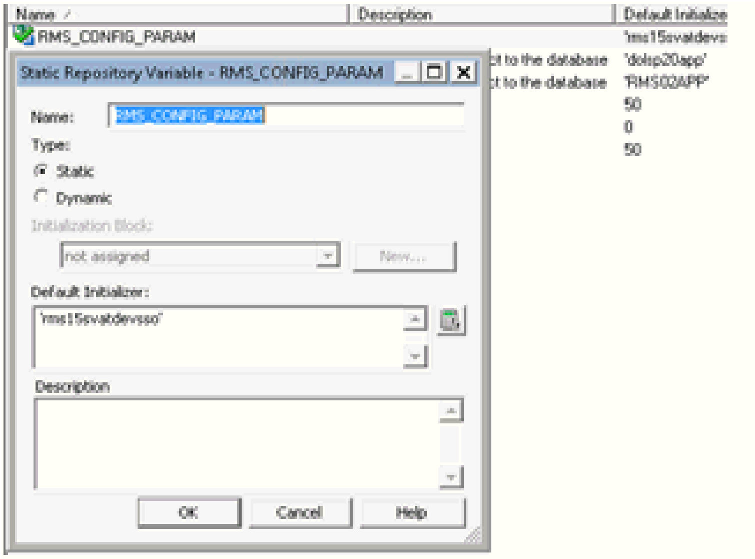
1. From the application home page, select **Preferences** from the logged in User menu.
2. From the Tasks pane, click **Accessibility** to open the Accessibility tab.
3. Select Default mode.
4. Click Save to save the settings.
5. Click **Back to Home** to return to the application home page. The application is returned to default mode and you can view the graphical reports.

Other configurations in Operational Insights

Set up the RMS_CONFIG_PARAM in the RPD for in-context launches into the RMS windows. RMS_CONFIG_PARAM is a repository variable that stores the configuration parameter of the RMS forms application. This variable is then internally used in reports to contextually launch into RMS forms applications from the Allocation dashboard.

To configure the variable:

Configure the Rpd variable for the RMS forms In-context launch. Change the value of Rpd Variable. Open the Rpd, go to **Manage -> Variable -> Repository -> Statics**, and change the value of Repository variable to **RMS_CONFIG_PARAM**.



Name	Description	Default Initialize
RMS_CONFIG_PARAM		'rms15svatdevs
	t to the database	'dolsp20app'
	t to the database	'RMS02APP'
		50
		0
		50

Static Repository Variable - RMS_CONFIG_PARAM

Name: RMS_CONFIG_PARAM

Type:

● Static

○ Dynamic

Initialization Block: not assigned

Default Initializer: 'rms15svatdevsso'

Description:

Note: If your RMS URL is as follows: http://hostname:port/forms/frmservlet?config= rms15svatdevsso, then the value of the **RMS_CONFIG_PARAM** variable is **rms15svatdevsso**.

Application Specific Configurations

Oracle Retail Operational Insights dashboards and reports are across Retail Merchandising Fusion Applications. This section details the application specific configurations required for use of Operational Insights for individual Retail Merchandising Fusion Application. The reports that are displayed in the application specific (Allocation or ReIM) dashboards as well as some data reported in some of the reports is configurable via the following database table.

Setting Up System Options in Allocation for Operational Insights Reports

The ALC_SYSTEM_OPTIONS_OI table drives the configuration parameters for the Allocation Operational Insights dashboards reports and is present in the Allocation application database schema. This following table describes the configuration parameters for OI reports for Allocation:

System Option	Definition	Column Name
Allocate By	The retailer may either choose to allocate based on incoming purchase orders or based on ASNs. Based on this value, either the PO Arrival Report or the Shipment Arrival report is the first report displayed on the dashboard. Values - PO; ASN	ALLOCATE_BY
Need Calculation Type	The PO Arrival/Shipment Arrival report, WH Inventory report and the Allocated to Plan/Forecast have the ability to support using either Forecast or Plan data.Retailers should configure this parameter based on the data that they have available for their items and wish to see in these reports. Values - P (plan); F (forecast)	NEED_CALC_TYPE
Simple Promo Only	When set to 'Y' the promotion flag is displayed only for simple promotions in the PO Arrival/Shipment Arrival reports. When set to 'N' the promotion flag is displayed for both simple and complex promotions in the PO Arrival/Shipment Arrival reports.	SIMP_PROMO_ONLY
PO Allocation Time threshold	Number of days before the not after date of the purchase order that the retailer expects a quantity greater than the 'Percentage Allocated PO Threshold' to be allocated.	PO_ALC_TIME_THRESHOLD
Percentage Allocated PO Threshold	The percentage of the warehouse order quantity against the given PO that is expected to be allocated within the 'PO Allocation Time Threshold' defined.	ALLOCATED_PO_THRESHOLD
ASN Allocation Time threshold	Number of days before the estimated arrival date of the ASN that the retailer expects a quantity greater than the 'Percentage Allocated ASN Threshold' to be allocated.	ASN_ALC_TIME_THRESHOLD
Percentage Allocated ASN Threshold	The percentage of the shipped quantity against the given ASN that is expected to be allocated within the 'ASN Allocation Time Threshold' defined.	ALLOCATED_ASN_THRESHOLD
Inventory Tolerance to Plan/Forecast	The tolerance percentage by which the inventory (current + inbound) can differ from the forecast/plan before raising an alert.	INV_TOL_TO_FORECAST_PLAN

Setting Up System Options in Invoice Matching for Operational Insights Reports

The IM_OI_SYS_OPTIONS table drives the configuration parameters for the REIM Operational Insights Dashbaords and reports and is present in the ReIM application database schema. The weightages are stored in a Configuration table in ReIM. The weightages must add to 100%.

System Option	Definition	Column Name
Weightage for exact match	Weightage in percentage for invoices to be of exact match.	WEIGHT_EXACT_MATCH
Weightage for non discrepancy	Weightage in percentage for invoices to be non descrepant.	WEIGHT_NON_ DISREPANCY
Weightage for timely shipment	Weightage in percentage for invoices to be timely shipped	WEIGHT_TIMELY_SHIP

8 Oracle Retail Allocation

This chapter provides an overview of Oracle Retail Allocation.

Information Maintained by Allocation

The following information is maintained in the Allocation application:

Standard Allocations - Oracle Retail Allocation provides the customer a number of sources from which to allocate products. These sources include:

- Purchase Orders
- Advanced Shipment Notification (ASN)
- Transfers
- Bill of Lading (BOL)
- Warehouse inventory
- Approved warehouse-to-warehouse allocation

Customers have more access and control to existing transactions as a result of the different item sources, which results in increased supply chain efficiencies.

What if Allocations - These allocations are similar to regular allocations with the exception that they have an infinite amount of product available to allocate. This allows a customer to determine the true need for the locations being allocated to, if they choose to create a purchase based on the allocation to fulfill the need. The user can also enter a finite available quantity during the creation of a 'what if' allocation, in which case this value gets treated as an upper cutoff.

ScheduledAllocations - Manually re-creating allocations that have the same criteria requires time. Oracle Retail Allocation allows users to define an allocation and schedule it to be automatically re-created on specific dates and time until the end date is met, warehouse stock is depleted, or threshold is met.

Allocation Templates - Oracle Retail Allocation users can create standard templates to apply to allocations to save time. Two types of templates can be created: allocation location groups and policy.

Policies - Oracle Retail Allocation requires the selection of a policy for the calculation of an allocation. The policy defines the source of the data used in the calculation of the allocation and other parameters that are used in the calculation.

Advanced Need Determining Calculations - Allocations are calculated in real time by advanced internal algorithms that calculate store or warehouse need based on the policy parameters established by the customer and the current perpetual inventory and sales for the items being allocated.

Split Allocations - Oracle Retail Allocation allows users to react to changes (such as short ships, delays, and cancelled product) by splitting an allocation. Using this functionality, users can split one or many items off from an existing allocation and onto a new allocation of their own.

Implementation Considerations

As part of the implementation of Allocation, customers may need to make specific choices about RMS item setup and location ranging. Customers may also need to populate tables used by Allocation in order leverage certain functionality, such as size profiles or auto quantity limits. This section provides details on the expected values to leverage these capabilities.

Allocation Item Types

The way items are classified in Allocation is different from RMS and most of the enterprise. The ALC_ITEM_TYPE is best explained with examples. This is not the exhaustive list of possible combinations, but is instead an illustration of possibilities. When examples of the data in the item_master table are presented, please be aware that the example only shows a small subset of the columns on the table. Not all items in the ITEM_MASTER table will be considered by Allocation. Thus, not all items in ITEM_MASTER will have the ALC_ITEM_TYPE column populated.

ALC_ITEM_TYPES

- Staple Item (ST)
- Sellable Pack (SELLPACK)
- Style (STYLE)
- Fashion Item (FA)
- Fashion SKU (FASHIONSKU)
- Pack Component (PACKCOMP)
- Non-sellable Fashion Simple Pack (NSFSP)
- Non-sellable Staple Simple Pack (NSSSP)
- Non-sellable Staple Complex Pack (NSSCP)
- Non-sellable Fashion Multi-color Pack (NSFMCP)
- Non-sellable Fashion Single Color Pack (NSFSCP)

Staple Item

A one level item is not related to any other items.

Item (ID, not null)	Item_ parent (nullable)	Item_ grandparent (nullable)	Item_level (not null)	Tran_level (not null)	Diff_1	Diff_2	ITEM_ AGGREGATE_IND	ALC_ ITEM_ TYPE
182920285	Null	Null	1	1	Null	Null	N	ST

Transaction level items of multiple level item families that have their aggregation indicator marked as N.

Item (ID, not null)	Item_ parent (nullable)	Item_ grandparent (nullable)	Item_level (not null)	Tran_level (not null)	Diff_1	Diff_2	ITEM_ AGGREGATE _IND	ALC_ ITEM_ TYPE
100001393	Null	Null	1	2	COLOR	SIZE	N	
100001828	100001393	Null	2	2	RED	SMALL	N	ST
100001561	100001393	Null	2	2	BLUE	SMALL	N	ST
100075018	Null	Null	1	3	null	null	N	
100075026	100075018	Null	2	3	null	null	N	
100075034	100075026	100075018	3	3	null	null	N	ST

Sellable Packs

All pack items that have ITEM_MASTER sellable_ind = Y are classified as Sellable Packs in Allocation.

Item (ID, not null)	Item_level (not null)	Tran_level (not null)	PACK_IND	SELLABLE_ IND	ALC_ITEM_ TYPE
110919650	1	1	Y	Y	SELLPACK
110919649	1	1	Y	Y	SELLPACK
111394648	1	1	Y	Y	SELLPACK

Fashion Item Families

These are item families where the transaction level is 2 and the aggregation indicator at level 1 is Y. Allocation introduces a new level between the level 1 item and the level 2 items. These intermediate level items (1.5) do not exist on the ITEM_MASTER table.

Level 1 items hold diff groups and level 2 items hold diffs in fashion item families

Item (ID, not null)	Item_ parent (nullable)	Item_ grandparent (nullable)	Item_ level (not null)	Tran_ level (not null)	Diff_1	Diff_2	ITEM_ AGGREGATE_ IND	DIFF_ 1_ AGGREGATE_ IND	DIFF_ 2_ AGGREGATE_ IND	ALC_ITEM_ TYPE
100001393	Null	Null	1	2	COLOR	SIZE	Y	Y	N	STYLE
100001828	100001393	Null	2	2	RED	SMALL	N	N	N	FASHIONSKU
100001561	100001393	Null	2	2	RED	LARGE	N	N	N	FASHIONSKU
100001465	100001393	Null	2	2	BLUE	SMALL	N	N	N	FASHIONSKU
100001721	100001393	Null	2	2	BLUE	LARGE	N	N	N	FASHIONSKU

The Allocation constructed items (FA) between the STYLE items and the FASHIONSKU items are a combination of the STYLE and the unique aggregation diffs of the FASHIONSKU items. In our example the STYLE is 100001393. The unique aggregation diffs (where the DIFF_X_AGGREGATION_IND is Y at the STYLE level): RED, BLUE. They are concatenated in this format {STYLE agg position~aggregation diff}

ITEM	ALC_ITEM_TYPE
100001393 1~RED	FA
100001393 1~BLUE	FA

The same example if DIFF_2_AGGREGATE_IND is Y instead of DIFF_1_AGGREGATE_IND

Item (ID, not null)	Item_ parent (nullable)	Item_ grandparent (nullable)	Item_ level (not null)	Tran_ level (not null)	Diff_1	Diff_2	ITEM_ AGGREGATE_ IND	DIFF_ 1_ AGGREGATE_ IND	DIFF_ 2_ AGGREGATE_ IND	ALC_ITEM_ TYPE
100001393	Null	Null	1	2	COLOR	SIZE	Y	N	Y	STYLE
100001828	100001393	Null	2	2	RED	SMALL	N	N	N	FASHIONSKU
100001561	100001393	Null	2	2	RED	LARGE	N	N	N	FASHIONSKU
100001465	100001393	Null	2	2	BLUE	SMALL	N	N	N	FASHIONSKU
100001721	100001393	Null	2	2	BLUE	LARGE	N	N	N	FASHIONSKU

ITEM	ALC_ITEM_TYPE
100001393 2~SMALL	FA
100001393 2~LARGE	FA

Non-sellable Staple Simple Pack

These are non-sellable simple packs that contain only 1 component item. The component item must be classified as a Staple item by Allocation.

In this example pack item 110919650 is a simple pack with 1 component item 110919649 which is a ST item.

Item (ID, not null)	Item_level (not null)	Tran_level (not null)	PACK_IND	SELLABLE_ IND	SIMPLE_ PACK_IND	ALC_ITEM_ TYPE
110919650	1	1	Y	N	N	NSSSP
110919649	1	1	N	Y	N	ST

Non-sellable Fashion Simple Pack

These are non-sellable simple packs that contain only 1 component item. The component item must be classified as a Fashion sku item by Allocation.

In this example pack item 110919650 is a simple pack with 1 component item (110919649) which is a FASHIONSKU item.

Item (ID, not null)	Item_level (not null)	Tran_level (not null)	PACK_IND	SELLABLE_ IND	SIMPLE_ PACK_IND	ALC_ITEM_ TYPE
110919650	1	1	Y	N	N	NSSSP
110919649	1	1	N	Y	N	FASHIONSKU

Non-sellable Staple Complex Pack

These are non-sellable complex packs that only contain staple items as components.

In this example pack item 110919650 is a simple pack with 3 component items which are ST items.

Item (ID, not null)	Item_level (not null)	Tran_level (not null)	PACK_IND	SELLABLE_IND	SIMPLE_PACK_IND	ALC_ITEM_TYPE
110919650	1	1	Y	N	N	NSSSP
110919649	1	1	N	Y	N	ST
110919648	1	1	N	Y	N	ST
110919647	1	1	N	Y	N	ST

Non-sellable Fashion Single Color Pack

These are non-sellable complex packs that only contain FASHIONSKU components. The component items must all share the same item_parent. The component items must all share the same aggregate diff values.

In this example, let's say pack item 110919650 has two components: 100001828 and 100001561. The two components share a common parent item (100001393). The two components are classified as FASHIONSKU in ALC_ITEM_TYPE. The aggregation diff for the STYLE the components both belong to is position 1 and both components have the same value for DIFF_1.

Item (ID, not null)	Item_parent (nullable)	Item_grandparent (nullable)	Item_level (not null)	Tran_level (not null)	Diff_1	Diff_2	ITEM_AGGREGATE_IND	DIFF_1_AGGREGATE_IND	DIFF_2_AGGREGATE_IND	PACK_IND	SELLABLE_IND	SIMPLE_PACK_IND	ALC_ITEM_TYPE
100001393	Null	Null	1	2	COLOR	SIZE	Y	Y	N	N	Y	N	STYLE
100001828	100001393	Null	2	2	RED	SMALL	N	N	N	N	Y	N	FASHION SKU
100001561	100001393	Null	2	2	RED	LARGE	N	N	N	N	Y	N	FASHION SKU
100001465	100001393	Null	2	2	BLUE	SMALL	N	N	N	N	Y	N	FASHION SKU
100001721	100001393	Null	2	2	BLUE	LARGE	N	N	N	N	Y	N	FASHION SKU
110919650	Null	Null	1	1	Null	Null	N	N	N	Y	N	N	NSFSCP

Non-sellable Fashion Multi Color Pack

These are non-sellable complex packs that only contain FASHIONSKU components. The component items must all share the same item_parent. The component items can have varying aggregate diff values.

In this example, let's say pack item 110919655 has two components: 100001828 and 100001721. The two components share a common parent item (100001393). The two components are classified as FASHIONSKU in ALC_ITEM_TYPE. The aggregation diff for the STYLE the components both belong to is position 1 and both components have different values for DIFF_1.

Item (ID, not null)	Item_parent (nullable)	Item_grandparent (nullable)	Item_level (not null)	Tran_level (not null)	Diff_1	Diff_2	ITEM_AGGREGATE_IND	DIFF_1_AGGREGATE_IND	DIFF_2_AGGREGATE_IND	PACK_IND	SELLABLE_IND	SIMPLE_PACK_IND	ALC_ITEM_TYPE
100001393	Null	Null	1	2	COLOR	SIZE	Y	Y	N	N	Y	N	STYLE

Item (ID, not null)	Item_ parent (nullable)	Item_ grandparent (nullable)	Item_ level (not null)	Tran_ level (not null)	Diff_1	Diff_2	ITEM_ AGGREGATE_IND	DIFF_ 1_ AGGREGATE_IND	DIFF_ 2_ AGGREGATE_IND	PACK_IND	SELLABLE_IND	SIMPLE_ PACK_IND	ALC_ ITEM_ TYPE
100001828	100001393	Null	2	2	RED	SMALL	N	N	N	N	Y	N	FASHION SKU
100001561	100001393	Null	2	2	RED	LARGE	N	N	N	N	Y	N	FASHION SKU
100001465	100001393	Null	2	2	BLUE	SMALL	N	N	N	N	Y	N	FASHION SKU
100001721	100001393	Null	2	2	BLUE	LARGE	N	N	N	N	Y	N	FASHION SKU
110919655	Null	Null	1	1	Null	Null	N	N	N	Y	N	N	NSFMCP

Items Not Supported By Allocations

Item below transaction level items

Allocation does not support Items where the item_level is greater than the tran_level.

Non-sellable complex packs that contain a mix of FASHIONSKU and ST components

Allocations does not support packs with both ALC_ITEM_TYPE = FASHIONSKU and _ITEM_TYPE = ST components.

Non-sellable complex packs that contain FASHIONSKU items with different parent items

Allocations does not support packs with component items that belong to more than one fashion item family.

Size Profile

Overview

All fashion, fashion pack, and fashion group allocations need to have size profile information. Size profile information is determined based on the selling ratio of the fashion items and is used to spread the quantity being allocated from Style/Color down to the SKU level. If an item / destination location does not have size profile information, it is excluded while performing the calculations.

In order to properly use Allocation for fashion items, customers need to populate the size profile data either through the UI or the database table. Each record should have hierarchy, location, and quantity information and should only contain details relevant to the target hierarchy level. For further information about size profile please see the Allocation and SPO section.

A combination of a merchandise hierarchy and location group can have multiple size profiles set up, which are termed as Generation IDs (GIDs). Size Profile Optimization application in RPAS provides size profile information for different seasons. These seasonal profiles are stored using three different tables in Allocation.

Table 8–1 ALC_GID_HEADER

Column	Required	Description
ID	Y	This is the primary unique ID generated from the sequence
GID	Y	This is the GID season code, populated from SPO
GID_DESC	Y	The description for the GID
CREATED_BY	Y	Indicates the user who created the record
CREATION_DATE	Y	The timestamp of the record creation date
LAST_UPDATED_BY	Y	Indicates the user who last updated the record
LAST_UPDATE_DATE	Y	The timestamp of the last updated date for the record
LAST_UPDATE_LOGIN	Y	Indicates the session login associated to the user who last updated the row.

Table 8–2 ALC_GID_PROFILE

Column	Required	Description
GID_ID	Y	This is the foreign key to the primary key of ALC_GID_HEADER
GID_PROFILE_ID	Y	This is the primary unique ID generated from the sequence
CREATED_BY	Y	Indicates the user who created the record
CREATION_DATE	Y	The timestamp of the record creation date
LAST_UPDATE_DATE	Y	The timestamp of the last updated date for the record
LAST_UPDATE_LOGIN	Y	Indicates the session login associated to the user who last updated the row

Table 8–3 Size Profile Table details

Column	Required	Description
SIZE_PROFILE_ID	Y	This column is auto-generated
LOC	Y	This column stores the location to which the size profile will apply. This column should always be populated
DEPT	N	This column stores the dept to which the size profile will apply. This column should be populated only for dept, class, and subclass level size profiles.
CLASS	N	This column stores the class to which the size profile will apply. This column should be populated only for class, and subclass level size profiles.

Table 8–3 (Cont.) Size Profile Table details

Column	Required	Description
SUBCLASS	N	This column stores the subclass to which the size profile will apply. This column should be populated only for subclass level size profiles.
STYLE	N	This column stores the Style Id to which the size profile will apply. This column should be populated only for Style, or Style Diff level size profiles.
SIZE1	N	This column stores the aggregated or non-aggregated Diff_1 information to which the size profile will apply. In the case of Style_Diff level, This column contains the aggregated or non-aggregated diff. In the case of Dept, Class, Subclass or Style Level; This column contains the non-aggregated diff.
SIZE2	N	This column stores the aggregated or non-aggregated Diff_2 information to which the size profile will apply. In the case of Style_Diff level, This column contains the aggregated or non-aggregated diff. In the case of Dept, Class, Subclass or Style Level; This column contains the non-aggregated diff.
SIZE3	N	This column stores the aggregated or non-aggregated Diff_3 information to which the size profile will apply. In the case of Style_Diff level, This column contains the aggregated or non-aggregated diff. In the case of Dept, Class, Subclass or Style Level; This column contains the non-aggregated diff.
SIZE4	N	This column stores the aggregated or non-aggregated Diff_4 information to which the size profile will apply. In the case of Style_Diff level, This column contains the aggregated or non-aggregated diff. In the case of Dept, Class, Subclass or Style Level; This column contains the non-aggregated diff.
SIZE_GROUP1	Y	This column stores the value 'X'
SIZE_GROUP2	N	This column stores the value 'null'.
QTY	Y	This column stores the individual size profile quantity.
CREATED_BY	Y	This column stores the created by user name
CREATION_DATE	Y	This column stores the creation date
LAST_UPDATED_BY	Y	This column stores the last updated by user name

Table 8–3 (Cont.) Size Profile Table details

Column	Required	Description
LAST_UPDATE_DATE	Y	This column stores the last updated date
LAST_UPDATE_LOGIN	Y	This column stores the last updated login
OBJECT_VERSION_NUMBER	Y	This column store the object version number
GID_PROFILE_ID	N	This column stores the GID_PROFILE_ID. This column is populated when the size profile is part of a Seasonal or Generation set. This column is null if it is a standard size profile. This column is a Foreign Key on the table ALC_GID_PROFILE
SIZE_PROFILE_LEVEL	Y	This column stores a numerical size profile level. Dept = 1, Class = 2, Subclass = 3, Style = 4, Style_Diff = 5

Auto Quantity Limits

Overview

Auto quantity limits provides users a way to store pre-defined quantity limits for multiple merchandise hierarchy levels, including item, style diff, style, department, class, and subclass levels. Auto quantity limits will automatically use the lowest available hierarchy level to apply to each item location. The quantity limits information retrieved from the auto quantity limits table will display in the existing quantity limits section of the UI. These values will then be saved to the Allocation as normal quantity limits, so once auto quantity limits are applied to an allocation, changes to the values in the auto quantity limits table will not be reflected. Allocations can use auto quantity limits through a checkbox in the quantity limits tab. Auto quantity limits can be used by default for every allocation through a system option. A new UI is now available to define the Auto Quantity Limits.

Implementation

In order to use the auto quantity limits, customers need to populate the auto quantity limits table manually or through the UI. Each record should have hierarchy and location information and should only contain information relevant to the target hierarchy level. Every record uses a start and end date to determine the effective dates of that record. Start dates must be populated, However, end dates are optional.

Not Supported

Overlapping dates for a particular hierarchy level location are not supported. If there is a record with no end date specified, then only records with start and end dates can be populated in the time period before the start of the original record which has no end date.

Table 8–4 Auto Quantity Limits

Column	Required	Description
AUTO_QUANTITY_LIMITS_ID	Y	This column is auto-generated

Table 8–4 (Cont.) Auto Quantity Limits

Column	Required	Description
LOCATION_ID	Y	This column stores the location to apply the quantity limits to. This column should always be populated
DEPT	N	This column stores the dept to apply the quantity limits to. This column should be populated only for dept, class, and subclass level quantity limits.
CLASS	N	This column stores the class to apply the quantity limits to. This column should be populated only for class, and subclass level quantity limits.
SUBCLASS	N	This column stores the subclass to apply the quantity limits to. This column should be populated only for subclass level quantity limits.
ITEM_ID	N	This column stores the Item id or Style Id to apply the quantity limits to. This column should be populated only for Item, Style, or Style Diff level quantity limits.
DIFF_1	N	This column stores the Diff_1 information to apply the quantity limits to. This column should be populated only for Style Diff level quantity limits where Diff_1 is the aggregate Diff.
DIFF_2	N	This column stores the Diff_2 information to apply the quantity limits to. This column should be populated only for Style Diff level quantity limits where Diff_2 is the aggregate Diff.
DIFF_3	N	This column stores the Diff_3 information to apply the quantity limits to. This column should be populated only for Style Diff level quantity limits where Diff_3 is the aggregate Diff.
DIFF_4	N	This column stores the Diff_4 information to apply the quantity limits to. This column should be populated only for Style Diff level quantity limits where Diff_4 is the aggregate Diff.
START_DATE	Y	This column stores the date when this record can start to be applied. This column should always be populated. (This Column may be depreciated in version 15.0)
END_DATE	N	This column stores the date that this record can no longer be applied. This column does not need to be populated, in which case, anything date after the start date is valid. (This Column may be depreciated in version 15.0)

Integration with Other Applications

A retailer that acquires Oracle Retail Allocation gains the ability to achieve more accurate allocations on a stable product. Having the right product in the right stores or warehouses allows for service levels to be raised, sales to be increased, and inventory costs to be lowered. By accurately determining which stores or warehouses should get which product, retailers can meet their turnover goals and increase profitability.

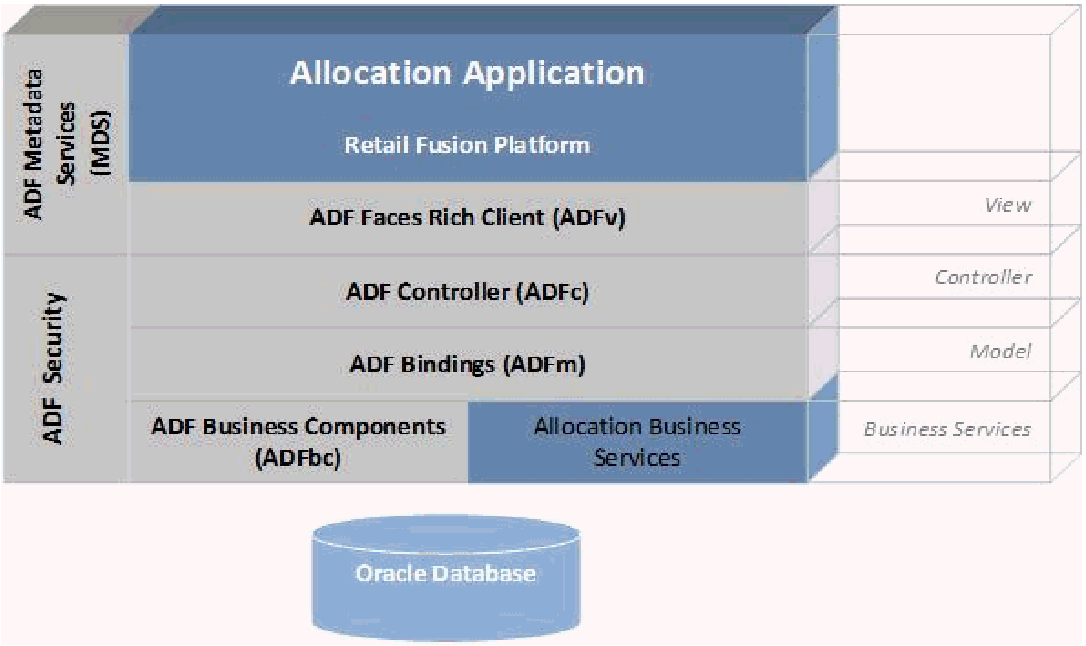
The Oracle Retail Allocation retailer benefits from the following capabilities:

- Built on ADF Technology stack, it allows the ability to quickly add UI based on ready to use design patterns, metadata driven tools and visual tools. Debugging can be performed more rapidly; maintenance and alteration costs are kept low using the Meta data driven Application Development.

- The application's interface takes advantage of the Java Database Connectivitiy (JDBC), ADF's built-in transaction management, along with connections to datasources handled in Weblogic server hence minimizing the interface points needed to be maintained.
- The application's robust algorithm executes rapidly and the call to the calculation engine has been ported over from C++ library to a Java Library thus minimizing the overhead/issues related to maintaining codebase consisting of two languages.
  - For retailers with other Oracle Retail products, integration with the Oracle Retail product suite means that item, purchase order, supplier, sales, and other data are accessed directly from the RMS tables, with no need for batch modules. The allocation information containing the item, location, and allocated quantity details is passed from RMS to a warehouse management system, such as the Oracle Retail Warehouse Management System (RWMS).
- Access Control to the System is better managed by using Fusion Security Architecture.
- The application allows for retailers to adjust to changing trends in the market by facilitating real time allocations.

The following diagram illustrates the Allocation n-tier architecture:

Figure 8–1 Allocation n-tier Architecture



Allocation n-tier Architecture

RMS owns virtually all of the information that Oracle Retail Allocation needs to operate and the information that Oracle Retail Allocation provides is of primary interest/use for RMS. As a result Oracle Retail Allocation has limited interaction with other Oracle Retail Merchandising Operations Management applications. For Oracle Retail Merchandising Operations Management applications that Oracle Retail Allocation does interact with, it is managed through direct reads from Oracle Retail Merchandising Operations Management application tables, direct calls to Oracle Retail

Merchandising Operations Management packages, and Oracle Retail Allocation packages based on Oracle Retail Merchandising Operations Management application tables.

For more information on the Merchandising Architecture, see *Retail Reference Architecture artifacts on My Oracle Support*.

Allocation and RMS

RMS provides the following to Allocation:

- **Foundation Data**-This information is essential to all areas of Oracle Retail Allocation including valid locations to allocate to and from, location groupings, valid merchandise hierarchies to allocate within, and so forth.
- **Item**-Allocations are generated at the item location level so it is necessary that the Allocation application understands what items and item/locations are eligible in the system.
- **Purchase Order**-One of the sources from which a user allocates is an approved Purchase Order. Oracle Retail Allocation relies on RMS to provide Purchase Order information.
- **Transfer**-One of the sources from which a user allocates is an approved Transfer. Oracle Retail Allocation relies on RMS to provide Transfer information.
- **BOL**-One of the sources from which a user allocates is a bill of lading. Oracle Retail Allocation relies on RMS to provide BOL information.
- **ASN** -One of the sources from which a user allocates is an ASN. Oracle Retail Allocation relies on RMS to provide ASN information.
- **Inventory** -In order to determine the correct need at an item-location level before performing an allocation, the application needs visibility to the current on-hand inventory at each location being allocated to. Oracle Retail Allocation relies on RMS to provide inventory information at the item/location level.
- **Shipping Information** - Once an allocation is shipped by the Warehouse Management System, this information is passed on to the RMS. This shipment information implies that the allocation has been processed and is not available for any further user edits from the UI. Oracle Retail Allocation relies on the RMS to provide shipment information.
- **Sales Information** - Oracle Retail Allocation uses historical sales, forecast sales, and plan in order to determine the need at an item/location level for an allocation. Oracle Retail Allocation relies on RMS to provide sales information, RDF to provide forecast information, and Assortment Planning (or external planning system) to provide plan information.
- **Item/Location Ranging** - Oracle Retail Allocation uses the item/location ranging data set up in RMS. The following logic is applied during the determination of valid item/location combinations:

  a. A location gets staple items allocated if and only if the SKU has a valid item/location status.

  b. A location gets fashion items (style/color or parent/diff) allocated if and only if the parent as well as all its child items being allocated have a valid item/location status.

  c. A location gets sellable packs (fashion/staple/simple/complex) allocated if and only if the pack has a valid item/location status.

d. A location gets non-sellable packs (fashion/staple/simple/complex) allocated if and only if the pack as well as all its components have a valid item/location status.

Valid item/location status are the ones not defined in the system options - 'LOCATION EXCEPTION REASONS - PRODUCT SOURCED ALLOCATION' and 'LOCATION EXCEPTION REASON - "WHAT IF" ALLOCATION'.

- **Allocation** - One of the sources from which a user allocates is an approved warehouse-to-warehouse allocation. Oracle Retail Allocation relies on RMS to provide this information.

> **Note:** The source identifier that is used is the one in Allocation tables and not the ones in RMS owned tables.

Allocation provides the following to RMS:

- **Allocations** - Once an allocation has been moved to Approved or Reserved status, the allocation is written to RMS tables to give visibility to the allocation results.
- **Purchase Orders created by What-If process in Allocation** - If the user selects the What-if option when creating an allocation, the allocation is created based on current need and then have RMS build a Purchase Order from the allocation to fulfill the need. Oracle Retail Allocation uses an RMS API to build the purchase order in RMS.

Allocation and RPM

RPM provides the following to Allocation:

- **Future retail** - Oracle Retail Allocation has the ability to get a real time price from RPM with which it is directly integrated. Allocation uses this data to provide the future retail price values of the set of items present in the entire allocation (based on their quantities). In addition, you can access future retail price values by item and by location.
- **Promotions** - You can associate active or future promotions present in RPM while creating an allocation.

Allocation provides no information to RPM.

Allocation and RTM

No information is exchanged.

Allocation and ReSA

No information is exchanged.

Allocation and ReIM

No information is exchanged.

Allocation and ARI

ARI is a monitoring system that interacts with any applications database (including Oracle Retail Allocation). As such, it does not use any information from Oracle Retail

Allocation; rather, it monitors the Oracle Retail Allocation database for events defined by a customer and notifies the customer when said events occur.

Allocation and AP

Allocation is integrated with RPAS Assortment Planning via RETL using the Receipt Plan feature. Assortment Plan data ideally represents what the store is expected to receive at the item level.

Allocation and SPO

Prior to Oracle Retail Allocation Release 13.3, the option to select and apply a specific set of store size profile data was not available to be used during the allocation process. Logic was set to always apply the lowest level defined in the ALC_SIZE_PROFILE table. Oracle Retail Size Profile Optimization enables you to create various seasonal sets of store size profile data in terms of generation IDs (GIDs). GIDs are commonly created in advance of a selling period and based off current trend. Oracle Retail Allocation will allow you to select a specific GID to be applied during the allocation process. This is illustrated in the following three business case examples:

- **Business Case 1**: When creating a pre-allocation for the upcoming fall season. You may want to use a forward out Subclass Fall 2015 set of size profile data. This helps to better align the planning and execution process.
- **Business Case 2**: When creating an in-season allocation, where the allocation is expected to be processed the next day, you may want to take advantage of a current trend GID and select the Style level GID set of size profile data. Results will be based off actual trend data.
- **Business Case 3**: For key items, you may want to apply key item level specific GIDs instead of the store size profile data created at higher levels, such as Class or Subclass.

> **Note:** The GID option applies only to stores. For a fashion allocation involving destination warehouses, there are other options to determine the size profile. For more details, see the *Oracle Retail Allocation User Guide*.

Allocation System Options

In order to increase operational efficiencies, Allocation has introduced the system properties settings into the UI. The Tasks menu > Allocation Foundation > Manage System Options allows you to view various system settings. Manage System Options displays two tabs System Properties and User Group Properties. System Properties will display global system settings which were configured and defined during install and implementation; the System Properties is controlled and maintained by the System Administrator user role. The User Group Properties displays settings in which the Allocation user group can manage as business trend, and business needs change due to a shift in season or a change in their business model. The User Group Properties is controlled and maintained by the Allocation Manager user role. Allowing allocation users to have view access of these settings allows for them to better understand the Allocation product, process and results.

System Properties

- CALCULATION QUEUE POLLING INTERVAL

  Indicates the calculation queue polling interval in milliseconds. This system option has been deprecated and does not impact the calculation queue due to the implementation of WLS JMS queue. This property requires a reboot/restart of Oracle Retail Allocation to take effect.

- ITEM LOCATION WARNING

  Indicates whether a warning message needs to be displayed to the user in case of, selection of an invalid item/location combination. This system option is important for customers to understand that invalid item/locations combinations have been added in an allocation. Once these are identified, the user can take necessary steps to rectify them before proceeding with the workflow.

- END OF WEEK DAY

  Indicates the day treated as the end of the week. This system option is vital for all customers implementing Oracle Retail Allocation. Any weekly rollups performed by the application during need calculations are based on this setting. For accurate results, this needs to be in sync with the setup within the merchandising system.

- BULK WAREHOUSE SETTING

  Indicates the non-finisher virtual bulk warehouse ID for PO creation for What If allocations. This is a non-finisher virtual warehouse where the customer would require the delivery of bulk purchase orders created out of What If allocations. It needs to be noted here that this warehouse would be considered only in cases where the destination stores do not have a designated default delivery warehouse in the merchandising system.

  Business example: Bulk Warehouse Setting = VWH1. For store S1, default delivery warehouse in the merchandising system = VWH2. For store S2, there is no default delivery warehouse in the merchandising system. In the above setting, a bulk PO raised for S1 would be sent to VWH2 and for S2 would be sent to VWH1.

- AUTOMATIC GROUP UPDATE

  Indicates whether the location groups need to be updated for worksheet allocations. This system option is important for customers who extensively use location groups. In cases where a location group undergoes modifications within the merchandising system, where there are stores that were added to or deleted from the group, the Allocation user would be alerted of such changes on accessing an allocation making use of the modified location group.

- ALL ORDERS

  Indicates whether the On Order quantities against open purchase orders are considered while calculating item stock on hand.

  On Order quantities against open purchase orders are considered while calculating stock on hand (SOH) for the items in the order only if this option is set to Yes. This setting needs to be taken into consideration while analyzing the net need quantity generated for a store by the calculation algorithm.

- SUPPLY CHAIN PATH SETTING

  Retailers intending to use date-based paths might consider using the distribution level of 2 whereas those using static time-based paths would want to use the distribution level of 1. Change in this value would invalidate all the allocations created in the previous distribution level setup and not allow the user to access them.

> **Note:** The distribution path set up by the retailer is unidirectional and cannot be used to move goods back to the deconsolidation center from the store. Currently, this field is defaulted to zero and disabled till Multi Level Distribution (MLD) functionality is introduced.

- WHAT IF ITEM SOURCE QUERY LEVEL

  For this property, the retailer would need to set the merchandise hierarchy at which the maximum number of item queries are likely to be carried out while creating a What If allocation. This would largely need to be a corporate decision during the implementation phase.

- LOCATION EXCEPTION REASONS - PRODUCT SOURCED ALLOCATION

  Indicates the item-location relationship status that needs to be excluded from product sourced allocations. Separate multiple statuses with a space. Example: Location Exception Reason Product Sourced = C D I. If you want to exclude a non-existing item location relationship, add NULL to the list.

  Within the merchandising system, there are multiple item-location relationships that may exist. During the implementation phase, it is very important that the retailer takes a decision around which of these relationships would be considered valid during the creation process for a regular allocation. Defining the set of invalid relationship status through this system option removes an additional overhead of having to individually examine each allocation and manually remove invalid item location combinations.

- LOCATION EXCEPTION REASON - "WHAT IF" ALLOCATION

  Indicates the item-location relationship status that needs to be excluded from What If allocations. Separate multiple statuses with a space. Example: Location Exception Reasons What If = C D I. If you want to exclude a non-existing item location relationship, add NULL to the list.

  Within the merchandising system, there are multiple item-location relationships that may exist. During the implementation phase, it is very important that the retailer takes a decision around which of these relationships would be considered valid during the creation process for a 'what if' allocation. Defining the set of invalid relationship status through this system option removes an additional overhead of having to individually examine each allocation and manually remove invalid item location combinations.

- DEFAULT "WHAT IF" IMPORT WAREHOUSE

  Indicates the default warehouse for import based purchase orders from What If allocations. This is a non-finisher virtual warehouse where the customer would require the delivery of purchase orders created out of What If allocations. It needs to be noted here that this warehouse would be considered only in cases where the destination stores do not have a designated default delivery warehouse in the merchandising system.

  Business example: Default What If Import Warehouse = VWH1. For store S1, default delivery warehouse in the merchandising system = VWH2. For store S2, there is no default delivery warehouse in the merchandising system. In the above setting, a What If PO raised for S1 would be sent to VWH2 and for S2 would be sent to VWH1.

- "WHAT IF" SUMMARY DEFAULT ACTION

  Indicates the What If Summary Default Action: Create or Update PO.

- FUTURE AVAILABLE FOR WHAT IF ALLOCATIONS

  Indicates whether or not to consider Future Available inventory for What If Allocations. True - Use the future SOH; False - Use the current SOH only.

  While raising purchase orders out of What If allocations, this system option gives the retailer the extra edge of being able to see inventory likely to be delivered within the time horizon of the allocation at the locations being covered by the allocation. The order quantity gets optimized as a result of this and it also safeguards the retailer against over-allocation and markdown scenarios.

- ENABLE SIZE PROFILE VALIDATION

  Indicates if the size profile validation should be done when the user hits the Calculate button.

- SIZE PROFILE VALIDATION LEVELS

  Indicates the levels at which the validation should be done. The valid values are STYLE, STYLE/COLOR, SUBCLASS, CLASS, and DEPT. If you want to specify more than one value, then use the comma as a delimiter.

  This needs to be set to the merchandise hierarchy levels at which the retailer is likely to store the size profile data.

- SISTER STORE SETUP

  Indicates whether the need of a similar store can be used during allocation calculation. If this is set to True, the system uses the sister store's need when the records don't exist for a store. If this is set to False, the system uses the sister store's need when the records don't exist for a store or when there are existing records but with zero need.

  This gives the retailer the option to use item sales data from a like store in case of no existing records from the store in the allocation or there is a new store receiving goods for the first time and which is unlikely to have any past history data.

- LOCATION LIST THRESHOLD

  Indicates threshold value to be used in SQL IN while fetching a location list.

- UNLOCK MINUTES

  Indicates the locking timeout in minutes.

- PROMOTION ASSOCIATION

  Indicates whether the system should allow you to link promotions with an allocation during the creation process.

- BATCH PROVIDER URL

  Indicates the WebLogic context Uniform Resource Locator (URL) that the Async process uses. This property requires a reboot/restart of Oracle Retail Allocation to take effect.

- ITEM SEARCH MAXIMUM

  Indicates the limitation on the number of rows that are returned by the item search.

- ALLOCATION RETENTION DAYS

Indicates the number of days you can retain allocations that are not linked to RMS allocations in the system without them being picked up by the purge batch. This is calculated based on the last modified date of the allocation.

- WORKSHEET RETENTION DAYS

  Indicates the number of days to keep worksheets not linked to any RMS allocations in the system. Purging occurs once this time frame is over.

- NOTIFICATION POLLING INTERVAL

  Indicates the notification polling interval in milliseconds.

- NOTIFICATION POLLING TIMEOUT

  Indicates the notification polling timeout in milliseconds.

- ENABLE FUTURE RETAIL

  Indicates whether or not the retail values of the items being allocated are displayed to the user.

- ENABLE PROMOTIONS

  Indicates whether or not the system must allow the user to link promotions with an allocation during the creation process.

- CALCULATION LOG PATH

  Directory path to hold calculation dat files. This property requires a reboot/restart of Oracle Retail Allocation to take effect.

User Group Properties

The User Group properties are assigned to the Allocation Manager standard user role.

- NUMBER OF DAYS BEFORE RELEASE DATE

  Indicates the number of days before the release date, that is used during the creation of a purchase order for a What If allocation. This field is set to three days by default.

- DEFAULT RELEASE DATE

  Indicates whether or not to use Default Release Date. 'Yes' indicates Allocations will have a default release date and 'NO' indicates Allocation will not have a default release date.

- DEFAULT AUTO QUANTITY LIMITS

  Indicates whether or not to use Default Auto Quantity Limits. 'Yes indicates Allocation will have Default Auto Quantity Limits and 'No' indicates Allocation will not have Default Auto Quantity Limits

- BAYESIAN SENSITIVITY FACTOR

  Indicates the plan sensitivity value used while using the Plan Reproject policy. The sensitivity factor is set to 0.3 by default. This value can be changed to any value between zero to one based on the requirements.

- SECONDARY

  Indicates whether to display secondary description of store or supplier in store field and supplier field respectively.

- BREAK PACK ENFORCEMENT

Indicates whether the break pack functionality is enabled or not.

- PRESENTATION MINIMUMS

  Indicates if presentation minimums are initially defaulted into the quantity limits UI. This field impacts the default setting of the Auto Quantity Limits check box in the Quantity Limits tab on the Policy Maintenance window.

- DEFAULT STORE CALCULATION/PO MULTIPLE

  Indicates the default store calculation multiple. Possible Values are:

  - EA - Each
  - IN - Inner
  - CA - Case
  - PA - Pallet

- ITEM SOURCE DEFAULT FOR ITEM SEARCH PAGE

  Indicates the Item Source that will be checked by default when entering the Item Search page.

  - P - Purchase order
  - S - Advanced shipping notification
  - T - Transfer
  - B - Bill of lading
  - W - Warehouse
  - A - Allocation

- DESCRIPTION LENGTH

  Indicates the maximum length to be used for display of Item descriptions in the user interface.

- RULE VISIBILITY

  Indicates the rule type for which the need value is displayed on the Allocation Maintenance window.

- PACK VARIABLE ACCEPTANCE THRESHOLD

  Indicates the Pack Variance Acceptance Threshold value.

- QL SPLIT METHOD FOR LOCATION GROUPS

  Indicates the method of splitting quantity limits across individual stores in a location group.

- MAXIMUM ITEMS IN ALTERNATE HIERARCHY SELECTION

  Indicates the maximum number of items per alternate hierarchy selection.

- ENABLE LOCATION PRIORITY

  Allows the user to upload and maintain a spreadsheet containing details on the priority value associated with the set of stores and warehouses being used in the system.

- PACK RANGING

  Indicates the option of performing pack ranging at either of the following levels:

  - Pack Level: Allows the retail to plan and execute at the same level.

  - Component Level: Allows each unique component within the pack to be ranged to the store. If a single component of the pack is not ranged, the pack cannot be allocated to the store.

Database Statistics

In order to ensure optimal performance of Oracle Retail Allocation, ensure that an appropriate database statistics strategy is in place for the Allocation database tables. Of particular importance is a strategy for managing the temporary tables used within the calculation processing and user online sessions.

The calculations tables can be identified by the naming standard, ALC_CALC_***_TEMP, and they represent the volatile tables that are subject to high concurrent users and application logic inserting data (and later deleting). Gathering stats against these temp tables has always been a challenge because the number of rows changes at a very high frequency. Cardinality is the estimated number of rows that will be returned by a query operation and is the key to getting representative statistics (and not necessarily the latest statistics). With this architecture, we have no way of knowing during the session if the upper boundary of a temp table is 100 rows,10000 rows or 1000000 rows and so our cardinality is unpredictable and the Oracle Optimizer Engine gives you a false estimate and produce bad query response times.

The following is the list of the volatile tables which must be set up for using dynamic sampling or a locked statistics approach:

- ALC_CALC_DESTINATION_TEMP
- ALC_CALC_NEED_DATES_TEMP
- ALC_CALC_NEED_TEMP
- ALC_CALC_QTY_LIMITS_TEMP
- ALC_CALC_RLOH_ITEM_TEMP
- ALC_CALC_RLOH_TEMP
- ALC_CALC_SOURCE_TEMP
- ALC_HEAD_TEMP
- ALC_MERCH_HIER_RLOH_TEMP
- ALC_SESSION_ITEM_LOC
- ALC_SESSION_ITEM_LOC_EXCL
- ALC_SESSION_QUANTITY_LIMITS
- ALC_WORK_SESSION_ITEM
- ALC_WORK_SESSION_ITEM_ALL
- ALC_WORK_SESSION_ITEM_LOC

So generically, we need to choose one of two options: using DYNAMIC_SAMPLING, or locking the statistics at an approximate average level.

Dynamic Sampling

Using DYNAMIC_SAMPLING (an Oracle system parameter), Oracle gathers just enough stats on the referenced table to give it a best guess. So if you select dynamic sampling, you must drop the current stats on the reference temp table(s) and then lock them. This prevents the auto stats from collecting them or a dba mistakenly creating them and also tells the Optimization Engine that these referenced tables are candidates for dynamic sampling collection. Then, it is recommended setting the DYNAMIC_SAMPLING to 6. It can be set either at the system or session level. See the following article for more details on dynamic sampling: *Optimizer Dynamic Sampling (OPTIMIZER_DYNAMIC_SAMPLING) (ID: 336267.1).*

Locking Statistics

Oracle recommends that if you are not using dynamic sampling, their size be monitored and the statistics gathered at, or close to, their peak size. The statistics should then be locked using dbms_stats.lock_table_stats().

> **Note:** The volatile tables need to be maintained by running the AlcDailyCleanUp batch process nightly. For more details on the batch, see the Oracle Retail Allocation Operations Guide

The following are additional recommendations on statistics relevant to Allocation:

- If statistics are currently being gathered manually, Oracle assumes that automatic statistics gathering has been disabled. If not, set DBMS_STATS.SET_GLOBAL_PREFS ('AUTOSTATS_TARGET', 'ORACLE'). This will still gather dictionary table statistics.
- The remaining ALC% tables should follow best practices for statistics gathering using DBMS_STATS.GATHER_*_STATS procedures, namely allowing ESTIMATE_PERCENT to default to AUTO_SAMPLE_SIZE. Running these statistics on a weekly basis should suffice unless a major change occurs in the characteristics of the data (e.g. size, density, etc).
- Finally, if not carried out previously, we recommend that fixed object statistics be gathered (DBMS_STATS.GATHER_FIXED_OBJECTS_STATS) while there is a representative load on the system. If that is not feasible, then these should be at least gathered after the database is warmed-up and v$session, v$access, vsql, vsql_plan, etc are populated.

Internationalization

For details on the language supported information see, Oracle Retail Merchandising System documentation for the current release.

9

Oracle Retail Invoice Matching

This chapter is an overview of Oracle Retail Invoice Matching (ReIM).

Information Maintained by ReIM

The following information is maintained in the ReIM application.

Multi-dimensional matching utilizes complex matching logic designed to maximize match rates and processing productivity for both invoice and credit note matching.

Discrepancy routing identifies cost and quantity discrepancies when a match has not occurred after a user-specified period of time and automatically routes discrepancies to the Discrepancy Review List.

Resolution dialog offers a powerful, streamlined approach to handling invoice discrepancies where reviewers can disposition a discrepancy based on a set of user-defined reason codes.

Self-billing and deals bill-back integration provides robust integration with the Oracle Retail Merchandising System that supports supplier billing for RTVs, rebates and other deals, consignments, direct store delivery, evaluated receipts, and other non-merchandise billings from obligations and customs entry.

Receiver adjustments integration provides direct updates of receiver cost and quantity adjustments initiated from the matching/resolution process to inventory valuation and positions in the RMS.

Best terms date uses payment terms rankings (predetermined by the user) to identify the invoice or purchase order term best supporting the retailer's cash management objectives. Payment terms and terms date information is exported to the retailer's accounts payable solution to support payment of the invoice, as well as processing of other documents..

Debit reversals allow the user to efficiently convert a supplier-disputed debit memo into an editable credit memo with supplier comments for resolution.

Matching tolerances offers the flexibility to set up tolerances by monetary range, nominal amount, or percent. Separate tolerances can be applied for quantity and cost and for discrepancies in either the retailer's or its supplier's favor. Tolerances created as individual entities and can be mapped to supplier site, supplier, supplier group, or department. One tolerance entity can be designated as the system default.

Integration with Other Applications

The Invoice Matching application's primary purpose is to match invoices so they can be exported to Accounts Payable (AP) to be paid. Invoice Matching has limited interaction with the Oracle Retail Merchandising Operations Management applications with the exception of RMS. RMS is the owner of the information that Invoice Matching needs to match invoices it receives.

Information from the Invoice Matching application is shared with Oracle Retail Merchandising Operations Management applications through direct reads from Oracle Retail Merchandising Operations Management application tables, calls to Oracle Retail Merchandising Operations Management application packages, Invoice Matching packages based on Oracle Retail Merchandising Operations Management application tables, and batch processes.

For more information on the Merchandising Architecture, see *Retail Reference Architecture artifacts on My Oracle Support*.

Invoice Matching and RMS

RMS provides the following to Invoice Matching:

- **Foundation Data** is essential to all parts of invoice matching, including valid locations for invoices to be executed, valid suppliers from which to receive invoices, supplier addresses to send credits and debits based on invoice matching results, and more.
- **Item** information is essential to the invoice matching process as item information ensures that invoices being received are valid for the business. For example, an item received on an invoice is carried by the client, is supplied by the supplier who sent the invoice, and is carried in the locations for which the item was received.
- **Purchase Orders** are used by Invoice Matching to facilitate the invoice matching process which is performed at the purchase order location level.
- **Shipments** information is used by Invoice Matching to determine if a PO has been received, which affects the matching algorithm used by the AutoMatch batch program in Invoice Match, as well as online matching process.
- **Deals and Rebate**-Invoice Matching creates credit memos, debit memos, and credit note requests based on deal and rebate information in RMS for processing by the financial (AP) system. This is performed by the ComplexDealUpload and FixedDealUpload batch processes that read from RMS staging tables.
- **Staged Accounts Payable transactions**-Accounts Payable documents created in RMS for consignment invoices, Obligations invoices, customer entries invoices, payment transactions sent through ReSA, and Return to Vendor chargebacks (either debit memos or credit note requests) are staged to Invoice Matching staging tables in RMS and extracted using the batch EDIDLINV to be loaded as EDI documents into Invoice Matching.

Invoice Matching provides the following to RMS/RTM/ReSA:

- **Invoice Matching results for shipments**-Shipment records are updated with the invoice matching results from the invoice match process. This involves updating the match status and quantity matched of the shipments in question. The matching process is handled by the AutoMatch batch process in Invoice Matching, which attempts to match all invoices in ready-to-match, unresolved, or multi-unresolved status, as well as online matching process.

- **Receiver Cost Adjustments**-An API executed when invoice matching discrepancies are resolved through a receiver cost adjustment. The API updates the purchase order, shipment, and potentially the item cost in RMS, depending on the reason code action used.
- **Receiver Unit Adjustments**-An API is executed when invoice matching discrepancies are resolved through a receiver unit adjustment. The API updates the purchase shipment in RMS to complete the transaction.
- **Closing unmatched shipments**-Invoice matching closes the invoice matching status for shipments in RMS after a set period of time (defined by the client in system options). This updates the invoice matching status of the shipment on the shipment table in RMS. This process is managed by the ReceiptWriteOff batch program.

Invoice Matching and RTM

RTM provides to Invoice Matching:

- **Finalized Customs Entry**-When Customs Entries are confirmed in RTM, a non-merchandise invoice is automatically created in Invoice Matching staging tables.
- **Approved Obligations**-when an Obligation is approved in RTM, a non-merchandise invoice is automatically created in Invoice Matching staging tables.

Note: Invoice Matching provides no information to RTM.

RTM data, as well as any other data staged for ReIM in RMS would need to be extracted from RMS using EDIDLINV RMS process to generate EDI files and loaded into ReIM using EDIInjector from EDI files.

Invoice Matching and ReSA

ReSA provides the following to Invoice Matching:

Store Level Purchasing-Payments for merchandise purchases done at store level are booked against a corresponding merchandise invoice. Payments of non-merchandise purchases or miscellaneous services availed at the store are booked against a corresponding non-merchandise invoice. These transactions are passed from the POS/OMS to ReSA as specially designated PAID OUT transactions. All these invoices are assumed paid. The batch program SAEXPIM transfers the specially-designated PAID OUT type of transactions to the Invoice Matching staging tables for extract to the Invoice Matching application.

Escheatment Processing - Unclaimed monies of outstanding, non-expiring vouchers are totaled after a defined period from the date of issuance of the voucher and posted to Invoice Matching staging tables as a non-merchandise invoice by SAEXPIM. The unclaimed amount is paid out as income to the issuing retailer or, in some U.S. states; it is paid out to the state. ReSA determines who receives this income and accordingly posts a non-merchandise invoice for the partner. These invoices are assumed not paid.

Note: Invoice Matching provides no information to ReSA.

Invoice Matching and RPM

Information is not shared between these applications.

Invoice Matching and Allocation

Information is not shared between these applications.

Invoice Matching and ARI

ARI is a monitoring system that interacts with any applications database (including Invoice Matching). As such, it does not use any information from Invoice Matching; rather it monitors the Invoice Matching database for events defined by a client and notifies the client when said events occur.

Invoice Matching and Financial Systems

Invoice Matching exports data to financial staging tables through the FinancialPosting batch program. However, clients using any other system for financials must create their own interface to deliver information to that system. Invoice matching validates accounts against the financial system.

Valid accounts can be pre-loaded into Invoice matching to improve performance of the posting process.

Payment terms and currencies need to accurately synced between the financial system and RMS.

Invoice Matching and PeopleSoft

Invoice Matching allows drilling forward to see the interfaced document in PeopleSoft application.

Note: Invoice Matching does not provide drillback functionality from PeopleSoft.

Invoice Matching and External Suppliers

Invoice Matching gets invoices from external suppliers in one of two ways: EDI or hardcopy. When EDI is used, the EdiInjector batch program is responsible for uploading the invoice details from the vendor using a standardized file format. When a hardcopy is used, the client needs to manually enter the invoice in the system before matching can proceed.

Notification to suppliers of charge backs and requests for credit notes is provided in a flat file extracted by EdiDownload batch process.

Internationalization

For details on the language supported information see, Oracle Retail Merchandising System documentation for the current release.

10

Oracle Retail Price Management

This chapter is an overview of RPM.

Functional Design

RPM is a highly configurable, strategy-based pricing solution that suggests and assists with pricing decisions. RPM empowers retailers to automate and streamline pricing strategies across the organization, yielding a more predictable and profitable outcome. It provides decision support through pricing-focused business information to validate and approve new retails and markdown suggestions. This approach results in improved margins and strengthened productivity, all while remaining competitive.

RPM supports the creation and execution of manual price changes and clearances. It also provides semi-automated pricing functionality through the execution of a pricing strategy. Pricing strategies allow the retailer to define parameters that will propose retail prices based on competitive information, margin targets, or clearance objectives.

RPM is part of the Oracle Retail Enterprise footprint. It uses several methods of integration: RIB, Batch, and RSB. RPM is the system of record for all retail pricing. SIM can request the creation/update or deletion of price events, but there is no import option from SIM.

System Options and Defaults

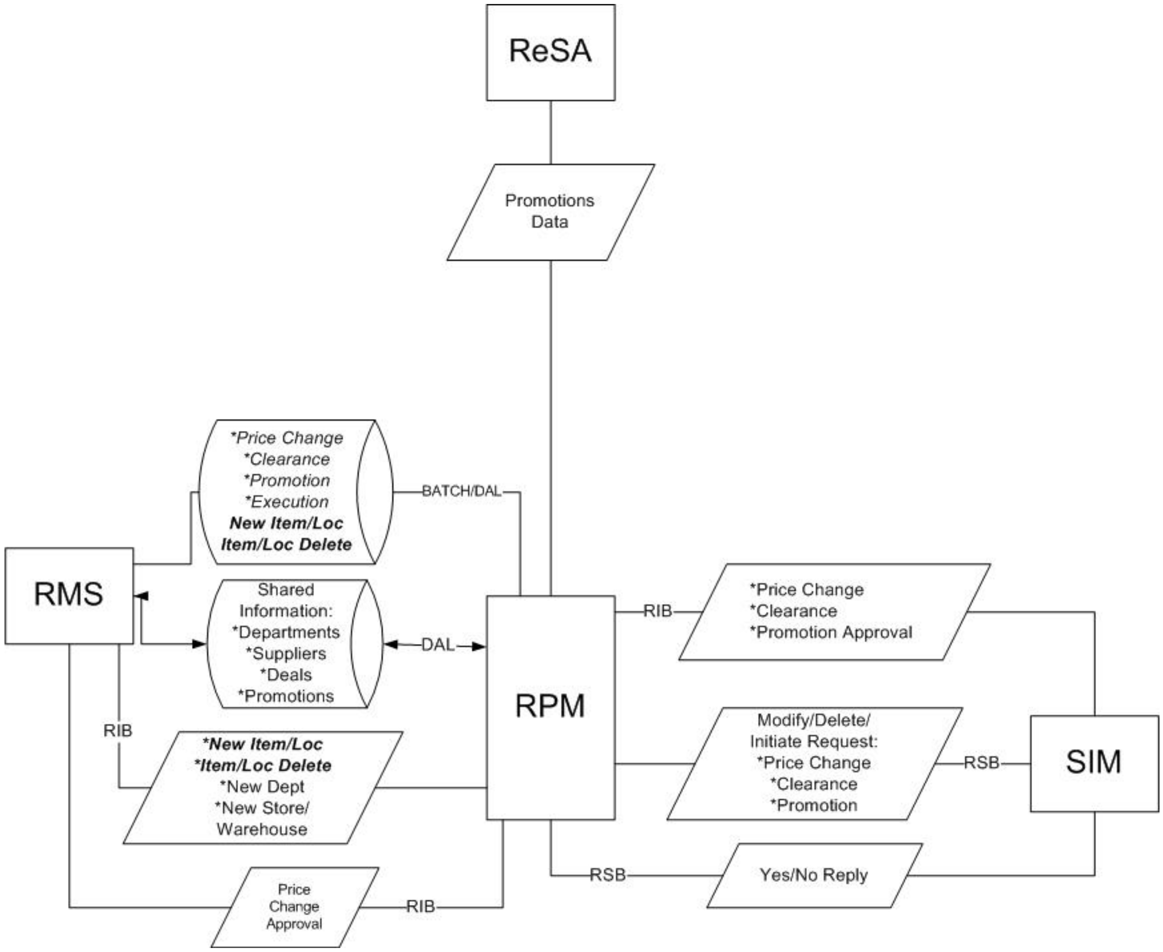
RPM includes a system options and defaults menu that allows the client to configure system settings as well as default values for certain dialogs. These are described in detail later in this chapter. Additionally, there are columns in the worksheet, worksheet status, price change and clearance dialog that can be displayed or hidden at the global level. This hide/show definition does not need to be supported with a GUI. It is important that during installation and implementation of the application, these system options are reviewed and set based on client business needs and/or preferences.

There are system options and system defaults that exist both in the table and the user interface. In addition, listed in each section is a default value for each system option/default field. This default value is populated upon entering the RPM application; however, values are not automatically stored in the database. The user must save the values in the system options user interface to commit the values to the database. This gives the user the option to change the values per business processes before committing any values to the RPM system options table.

RPM in the Oracle Retail Footprint

The following flowchart illustrates the position of RPM in relation to other Oracle Retail modules:

Figure 10–1 RPM Illustrated as Part of the Oracle Retail Footprint



Foundation Data

During installation and implementation, RPM requires some initial data setup to create and implement price changes, clearances, promotions and pricing strategies. This foundation data includes aggregation level, link codes, market basket codes, zone structures, price guides, calendar, and candidate rules. The following is a general overview of each functional area, including examples of lessons learned in the field. This information is to be used as reference when encountering issues or to avoid issues when implementing RPM.

Aggregation Data Functional Overview

The merchandise hierarchy allows the client to create the relationships required to support the product management structure of a company. Key information about how inventory is tracked, priced and reported is stored at the department level. It is very important that departments are set up in appropriate order for subsequent systems such as RPM to utilize. Departments are associated to zone groups, which assist in calculating the initial price for an item when created, approved and ranged to a location. The following is important information about using hierarchy information in RPM.

Adding a New Hierarchy in RMS

When creating a merchandise hierarchy in RMS, it is important that the department, class and subclass are set up correctly. Inappropriate setup will cause performance concerns in supporting applications such as RPM. The following are the correct and incorrect ways of creating Merchandise Hierarchical data.

The following is a suggested approach for creating Merchandise Hierarchy in RMS:

- **Department**-Next level below group in the merchandise hierarchy of a company. A group can have multiple departments. Key information about how inventory is tracked and reported is stored at the department level.
- **Class**-Next level below department in the merchandise hierarchy of a company. A department can have multiple classes. A class provides the means to group products within a department.
- **Subclass**-Next level below class in the merchandise hierarchy of a company. A class can have multiple subclasses. A subclass provides the means to classify products within a department/class combination.

Example:

- Department: Active Wear
- Class: Women's Active Wear
- Subclass: Women's Running Active Wear

The following is a less-effective, alternative approach to creating Merchandise Hierarchy in RMS:

In this example, the department is set up by designer; the department spans all locations and when it is used in RPM will cause performance issues. RPM was intended to support department data in order to create/implement price events under the Oracle Retail definition of what a department is in a retail store.

Example:

- Department: Prada (The department is set up by designer but spans many locations)
- Class: Prada Women's Apparel
- Subclass Prada: Women's Apparel-Active Wear

Deleting an Existing Department from RMS

If a department is deleted from the DEPS table in RMS, it should also be removed from the aggregation table in RPM. If the two tables are not in sync, the RPM application errors out with a fatal exception when entering the Aggregation Level user interface.

Markup Definition

For each Department (or Class or Subclass, if the Primary Zone Group has been defined at either of those levels), the client will need to specify the Initial Markup Percentage. This is used to determine the initial retail price for new items. It also necessary to specify whether this percentage should be applied as a Cost or Retail Markup.

The markup definition is defined in two places. The first is RMS in the DEPT form. The second is under the Maintain Primary Zone Group Definition in RPM. Even though there are two places where the setup can be defined, RPM reads and uses this information in the Maintain Primary Zone Group Definition, and all settings should be controlled there. Therefore, all modifications made to markup percentage should be done in RPM.

When the Markup Type and Markup percent is NULL for a department, or primary zone group does not exist, an embedded system default (DEF_MARKUP_TYPE) will default the markup type to RETAIL(R).

Link Codes Functional Overview

Link codes should be used for identical items that will always share the same retail price (for example, a retailer wants all frozen vegetables to carry the same price). They are used for ease of data entry and considered a point-in-time price change. When the link code price change is created and effective, it is considered complete and will not inherit new items/locations.

Link codes can be created at the Zone Group, Zone, or individual location levels and are held at an item/location level in RPM; only one can be assigned per item/location. For link code price changes, the Apply button validates the item/locations for which a price change is being created/edited, has link codes attached, and prohibits the user from applying the new/edited price change. When a price change is approved, this validation need not occur, as the state of the link when that price change was approved is respected. The state of the link means which items/locations were affected when the price change was approved at that point in time. If any of the item/locations in the link code fail conflict checking, the entire link code price change will fail conflict checking. The user will not be allowed to create a regular price change through the price change dialog that affects an item/location that has a link code assigned.

Link code functionality is not available in clearances. For worksheets, there are certain rules that need to be followed to determine how items with link codes should be handled in the worksheet. For each item/zone pulled by the extract program (assuming strategy is set up at a zone level), validation is performed to verify if there are any link codes that exist across all of the locations in the zone for the item. If there are varying link codes across the item/locations, the item will not be pulled into the worksheet. A NULL link code counts and is considered different than an item/location that has a link code assigned. Every item in the link code at the locations must also be represented in the worksheet, or the item will not be pulled into the worksheet.

For price changes created in SIM that affect a link code, the price change will need to be converted into a link code/location price change so the price change can be created in RPM (that is, the price change fails if it stays in a SKU/location form and affects a link code).

Link Codes and Inheritance

Link codes should be used as a grouping mechanism for items that should be priced the same for a particular price change and assumes all items that exist for the link code at the time the price is effective will be represented. Link codes do not inherit price events, nor is the link code dynamic and able to add details. As stated, it represents a point-in-time price change and should not be assumed or used to group items for future events, such as new item location, and location move.

- Proper use of a link code

Create a link with a grouping of cola that should be priced the same for an event. All items exist when price change is created. New retail price is reflected.

- Improper user of a link code

  Create a link with a grouping of cola that should be priced the same for an event. Create a new item and add to the link code. The link code price change already in effect will not reflect the new item unless a new price change is created.

Link Codes and Worksheets/Merchandise Extract

Link codes can be used within the worksheet/pricing strategy functionality. The link code must be the same for an item across all locations in the zone, and all items in the link code must be present.

Example:

Primary Area	Secondary Area
Link Code A	Link Code A
Item 1	Item 1
Item 2	Item 2
Item 3	Link Code B
Link Code B	Item 4
Item 4	Item 6
Item 5	Link Code C
Item 7	
Item 6	
Item 7	Item 3
Item 8	Item 8

From primary to secondary areas:

- A price change to link code A should propose a retail for link code A in the secondary area. This applies to items 1 and 2.
- A price change to link code B should propose a retail for link code B in the secondary area. This applies to items 4 and 6.
- Link code C does not exist in the primary area and therefore no retail will be proposed, it should be thrown out of the secondary area worksheet.
- Item 3 is in the link code in the primary area but is not in the link code in the secondary area. Because Item 3 was not an individual item in the primary area, it should not be an individual item in the secondary area and should be thrown out.
- Item 6 was a single item in the primary area, but is not in the link code in the secondary area. Item 6 should be represented within the link code in the secondary area and should have its retails proposed from the link code, not the item 6 change of the primary area.
- Item 8 is a single item in the primary area and single item in the secondary. No unique processing is needed.
- If an item does not exist in the primary area but exists in the secondary area, but is not a part of any link code in the secondary area, throw it out. There are multiple

errors that can be triggered by merchandise extract batch when a link code is not set up properly.

Error Descriptions

INVALID_SECONDARY_ITEM

This error occurs when the items in the secondary are mismatched with primary and, therefore, the secondary is not able to receive the primary's proposed retail for that secondary item. To rectify this error, ensure the item meets or does not meet any of the following conditions:

- Item does not belong to any link code, whereas the same item belongs to a link code in the primary area.
- Item does not exist in the primary area and does not belong to any link code in the secondary area.
- Item belongs to a link code that does not exist in the primary area.

MISSING_LINK_ITEM

This error occurs when one or more items from an item-link code/zone group are missing, or the entire group is excluded. To rectify this error, ensure the item meets or does not meet any of the following conditions:

- If any one item from a link code is present in the worksheet, all items in the same link code must also be present in the worksheet.
- Items sharing the same link code should have the same Basis UOM.
- Items sharing the same link code should all have the same Class VAT Indicator settings.
- For Margin and Maintain Margin Strategies, items sharing the same link code should have the same Margin Market Basket code.
- For Competitive Margin Strategies, items sharing the same link code should have the same Competitive Market Basket code.

VARIABLE_LINK_CODE

This error occurs when the item does not have the same link code at all locations in the zone. In order to rectify this, modify your link code to reflect the locations and ensure the link is setup with all locations used in the pricing strategy.

VARIABLE_LINK_MBC

This error occurs when the items sharing a market basket code do not have the same link code at all locations in the zone. In order to rectify this, compare the items in the market basket code to the item/locations in the link code.

VARIABLE_LINK_SELLING_UOM

This error occurs when the items sharing a link code do not have the same Basis UOM at all locations in the zone. In order to rectify this, verify the basis UOM for all the items in the link code at the locations in the zone.

VARIABLE_LINK_VAT_IND

This error occurs when the items sharing a link code do not have the same VAT indicator at all locations in the zone. In order to rectify this, check for the existence of the item on the VAT tables in RMS in comparison to the locations.

Link Codes and New Item Location Batch

The user must run the NewItemLocation batch to populate RPM_FUTURE_RETAIL with an item location record. When a user creates a link code, the user interface utilizes the item location table in RMS to verify the item/location exists. Potential issues can occur when a link code is created after the item is ranged in RMS and populated on ITEM_LOC but before the NewItemLocation batch has been run. If a price change is created against the link code before the batch is run, price changes will not be reflected since seed data does not exist on the RPM_FUTURE_RETAIL table. The correct approach is to range an item, run the NewItemLocation batch job and then create the link code.

Market Basket Codes Functional Overview

The maintain market basket codes area allows the client to assign market baskets codes to an item/zone. The items can be associated to the code through the merchandise hierarchy, at the item level, or through item attributes such as diff or diff type. They are used in Competitive Strategies to match, price above or price below the competitor price. Margin and Maintain Margin Strategies can set different targets by market basket code. The market basket codes are used to group items together with similar pricing characteristics. Only one market basket code per item/location can exist.

For example, price all items in the A market basket code at 30% margin while all items in the B market basket code items have a target of 20% margin. A market basket code could group highly competitive or margin-sensitive merchandise together. The user will have the ability to set up two market basket codes per item/zone: one to be used with the Competitive Pricing Strategy, and the other to be used with the Maintain Margin Pricing Strategy. When the merchandise extract batch program is run, the program will identify the pricing strategy being executed and associate the proper market basket code.

Market Basket Codes need to be created and maintained by the Database Administrator (DBA). There is no UI to create the list of values (LOV). The table is RPM_MBC_LOV_VALUES. The table includes MKT_BASKET_CODE (the market basket code), NAME (description of the code), and TYPE, which indicates if Market Basket Code (MBC) is for Margin strategies (1) or competitive strategy (0). The high level process is as follows:

1. User assigns market basket codes to item/zones in the Market Basket Code dialog.
2. User creates a Maintain Margin Pricing Strategy and assigns margin values to the market basket codes.
3. User creates a cost change and approves it in RMS.
4. Merchandise Extract is run and it identifies the cost change in RMS and generates a worksheet using the maintain margin pricing strategy and its corresponding market basket code.

Market Basket Codes and Link Codes

Items set up in a market basket code also can be in a Link code. The items in the market basket are considered a grouping of items with similar pricing attributes, such as competitive, and within that grouping the items are then instructed to be priced the same within a link code. In this scenario of using a link code/market basket code together, items must have the same Link Code at all locations in the zone in order to avoid merchandise extract batch errors.

Zone Structures Functional Overview

Zone structures in RPM allow you to define groupings of locations for pricing purposes and eliminate the need to manage pricing at a location level. At the highest level, these groupings are divided into categories called zone groups. While these zone groups might be flexibly defined, they are primarily defined by their pricing scheme. The three types of zone groups in RPM are regular zone groups, clearance zone groups, and promotion zone groups. In addition to being defined by pricing, zone groups are defined by the items being priced. The following are examples of zone groups:

- Regular price beverage zone group
- Regular price footwear zone group
- Promotion price beverage zone group

Within zone groups in RPM are groupings of locations (stores or warehouses) called zones. The function of these zones is to group locations together in a manner that best facilitates company pricing strategies. These zones might be flexibly defined. For example, the client might choose to create zones based on geographic regions, such as the following:

- U.S. East region
- U.S. West region
- Mexico stores

Similarly, the client can create zones with locations with similar characteristics, such as the following:

- U.S. urban stores
- U.S. rural stores

Contained within zones are locations. These locations can be stores or warehouses.

- **Regular Only**- Stores and warehouses allowed.
- **Clearance Only**- Stores and warehouses allowed.
- **Promotion Only**- Stores only, warehouse not allowed.

There are no restrictions on the number of locations a zone can contain. However, two rules apply to the relationship between locations and zones:

- A location cannot exist in more than one zone within a zone group. A location can, however, exist in multiple zone groups. For example, a New York City store might exist in the U.S. urban stores zone group as well as the U.S. East region zone group.
- All locations within the same zone must use the same currency.

When zone groups are created in RPM, users are able to assign them to primary zone group definitions. The primary zone group definition allows the user to specify the zone structure to use when pricing merchandise hierarchies, and how to initially price items in these hierarchies (markup percentage, markup type). These definitions can be created at the department, class, or subclass level.

When RMS publishes a new location or warehouse (if warehouses are recognized as locations per a system option) the message will include a pricing location. RPM will take the pricing location of the new location or warehouse and attempt to add the new location to every zone group/zone in which the pricing location exists. This will include Regular, Clearance and Promotion Zone Groups. If the Pricing location and the new location are not of the same currency, then the new location will be added to every Zone Group where the pricing location exists but the system will create a new zone for the location with the same currency as the new location.

This process works the same for the following location types: Company stores, virtual warehouses, franchise locations, stocking holding and non-stock holding.

Users can add, modify, or delete the primary zone group definition for a given merchandise hierarchy within RPM. There are limitations for deleting a primary zone group definition. For more information, see *Primary Zone Group - Delete Functionality* in the section below.

Primary Price Zone Benefits

When the primary price zone is used when creating a price event, location level data is stored at zone level versus location level. It is recommended to enter the item data at the highest level possible; Merchandise Hierarchy (department, class, or subclass), Parent Item, Parent Item DIFF-1, or using an Item List. This is also true for the selection of locations at primary zone level versus store level.

RPM supports storing future retails at higher than transaction item/location level i.e. style (parent), style/color (parent item/DIFF-1) and zone, only where the zone is part of the Primary Zone Group for a department. This rule is in place for all price event types, price change, clearance and promotions.

For example, if you create a promotion for a single parent item that has 50 color/size items associated and the primary zone used includes 500 store locations. The RPM future retail tables can be updated with as few as one record at the style (parent) and zone (primary zone) that represents all items and locations.

If you create the same promotion, but do not use primary zone, RPM can update the future retail tables with one record for each of the 500 store locations. The location rollup functions only when a department's primary zone group is used to create the price event. If non-primary zone groups are used, data will be stored at parent/location vs. parent/zone.

Examples:

- If you are setting up a clearance for an entire style, create the data at parent item, or item list level. Do not explode down to transaction item detail.
- When creating item lists, include parent items when you can instead of transaction level items.
- If you need to create price events at transaction item level, keep in mind the number of items you are selecting, because more items means slower response times.

Primary Zone Group — Delete Functionality

Primary zone groups are set up at one of the three levels of the merchandise hierarchy. The highest level of the merchandise hierarchy for a primary zone group cannot be deleted from the system.

Deletion rules include:

	Delete Rules		
Primary Zone Group Create Level	**Department**	**Class**	**Subclass**
Department Only	Not Allowed	n/a	n/a

Primary Zone Group Create Level	Delete Rules		
	Department	Class	Subclass
Department/Class	Not Allowed	Delete Allowed	n/a
Department/Class/ Subclass	Not Allowed	Delete Allowed if Subclass is Deleted	Delete Allowed

There is an ad hoc batch process that will process any merchandise hierarchy changes requested to the primary zone group - refer to the Operations Guide for more details.

Empty Zones and Price Events

A user can create an empty zone and add locations to the zone at a later date. They can create price events against the zone with no locations; however, conflict checking will not run and records are not generated on the future retail tables or RPM_ZONE_FUTURE_RETAIL. Those price events will be inherited after locations are added to the empty zone and the new item location batch is executed for items/locations in that zone.

Open Zone Use and Flexibility

The Open Zone Use System Option defines whether or not different Zone Group types can be used in all the pricing dialogs or if the type of the Zone Group will limit where it can be used. For example, if set to No (unchecked), then Promotion Zone Groups cannot be used in the clearance and price change dialogs or in the pricing strategies definition. It is recommended during implementation to set this as checked or Yes initially, so that when creating price events you can take advantage of rollup functionality. Setting this value to Yes allows flexibility in how/which zones can be used for certain price events. However, when checked or set to Yes, it cannot be unchecked.

Deleting and Adding a Location after Zone Exists

When a user adds a location to or deletes a location from an existing zone, the location move functionality is leveraged to move that location in or out of the zone. A location move is created in approved status and will move to scheduled when the scheduling batch is run. It is important to note that the location will not be added to the zone until it is properly moved in or out through a location move which is scheduled automatically when the user presses the Delete or Add button.

Price Guides Functional Overview

RPM price guides help users create a uniform pricing strategy. They are used to smooth proposed retails in order to maintain a consistent set of price points by rounding or applying ends in logic to retail values. Price guides can be set up at the corporate or department level. Department price guides can also link into corporate price guides. After price guides are defined, they can then be used when defining primary zone groups, creating price changes, clearances or promotions, and are very useful when performing what-if analysis for worksheets. The user can edit a price guide at any time, regardless of whether it is attached to a strategy, price change, promotion, or clearance. Edited price guide details will only affect retails derived by the price guide from that point on. It will not affect/overwrite any retails that have already been derived based on the old price guide details.

Price guides are optional. For customers that do not use price guides the following rules are enforced for "percent off" price events:

- Regular Price Changes: normal rounding rules are enforced
  - For example, last digit is between 0 and 4, round down; last digit between 5 and 9, round up
- Clearance or Promotion Events: retail will always round down

Calendar

Calendars are set up in RPM for the primary purpose of attaching them to pricing strategies. Calendars span a user-defined period of time and contain a review period that occurs once or many times over the duration of the calendar.

Candidate Rules Functional Overview

This set of rules is run against the items/locations being extracted from the merchandise system to determine if they should be flagged for review. They are defined at the corporate level and can contain variables at the department level. Candidate rules can be inclusive or exclusive. If they are inclusive, and the candidate rule is met, the item/location is flagged in the worksheet. When exclusive candidate rules are met, the item/location is excluded from the review when the merchandise extract program builds the worksheet. Candidate rules can also be active or inactive, allowing the user to suspend rules that are only needed at certain times of the year. Candidate rules are only run against the worksheet the first time the worksheet is created.

Exceptions:

Each review period has an indicator stating whether or not to run exceptions. If the indicator is set to Yes, the merchandise extract should tag those item/location records that are pulled into the worksheet with an exception flag if any of the following occur during a review period where exceptions are processed: competitor regular retail price changes, cost changes, and new item/location relationships.

For every item/location pulled into the worksheet, RPM attempts to propose a new retail based on the strategy attached to that item/location. When the worksheet is first created, the details of the strategy are saved. Updates to the strategy do not affect any worksheets that are currently being reviewed. The updates are only reflected in worksheets generated after the updates to the strategy are made. Until the worksheet has been locked, new retails will continue to be proposed using the strategy details every night the batch program is run.

Candidate Rules and Worksheets

Each review period has an indicator stating what kind of candidate rules to run, if any. The options on the calendar are to run only inclusion rules, only exclusion rules, both inclusion rules and exclusion rules, or none of the rules. Each item/location from the strategy to which the calendar is attached should be run against the rules. If the strategy is at the zone level, then any item/location within that zone that meets an exclusion rule should exclude the entire worksheet line item. Also, if there is a primary area that is being brought into the worksheet, the secondary areas attached to the primary area should run through all candidate rules as well.

Two types of candidate rules can be run:

- Inclusion (to flag an item on a worksheet as having met a rule)
- Exclusion (to prevent the item from making the worksheet)

For regular price strategies, when an item meets an inclusion rule it simply means that the rule column on the worksheet will populate and the user can see the rules are met. It should be noted that for regular price strategies they are simply flags or alerts, no processing takes place against them.

For the clearance strategy, when an item meets an inclusion rule the system is triggered to propose a markdown and populate the rule column. The markdown that is proposed is not related to the specific rule that was met, just that the rule was met and clearance proposed. The markdown is based on the clearance strategy and which markdown is next in that item's markdown lifecycle.

All items in the hierarchy level of the worksheet will make it into the worksheet (unless they meet an exclusion rule). Candidate rules are applied at the transaction item/location level. If any transaction/location meets a rule, markdowns are proposed for the transaction/zone. The user can decide if the rule met is inclusive of enough locations in the zone to take action or not. In order to take action against other SKUs in a parent or parent diff (or even a related style), it is important that other items are brought into the worksheet.

Candidate Rule Condition Definition Fields

The following table lists the fields, operators, and values.

Field Name	Operators	Values
Class	=, <, >, <=, >=, <>	LOV-Classes (Dept must be selected first)
Clearance	=	Yes / No
Current Margin %	=, <, >, <=, >=, <>	Numeric
Department	=, <, >, <=, >=, <>	LOV-Departments
Diff ID	=, <>	LOV-Diff IDs
First Received Date	=, <, >, <=, >=, <>	Date
Weeks since First Received Date	=, <, >, <=, >=, <>	Numeric
Item #	=, <, >, <=, >=, <>	LOV-(Dept, Class, or Subclass must be selected first)
Item List	=, <>	LOV-Item Lists
Last Received Date	=, <, >, <=, >=, <>	Date
Weeks since Last Received date	=, <, >, <=, >=, <>	Numeric
Markdown #	=, <, >, <=, >=, <>	Numeric
Margin Market Basket Code	=, <>	LOV-Margin market Basket Codes
Competitive Market Basket Code	=, <>	LOV-Competitive Market Basket Codes
Supplier	=, <, >, <=, >=, <>	LOV-Suppliers
Projected Sales	=, <, >, <=, >=, <>	Numeric
Promotions	=	Yes / No
Replenishment Indicator	=	Yes / No
Retail Label Type	=	LOV-Label type
Retail Label Value	=, <, >, <=, >=, <>	Numeric (Retail Label Type must be selected first)

Field Name	Operators	Values
Seasonal Sell Thru	=, <, >, <=, >=, <>	Numeric
Season Code	=, <, >, <=, >=, <>	LOV-Season Codes
Phase Code	=, <, >, <=, >=, <>	LOV - Phase codes (Season Code must be selected first)
Sell Thru	=, <, >, <=, >=, <>	Numeric
Package Size	=, <, >, <=, >=, <>	Numeric
Package UOM	=	LOV-UOMs (Package size must be selected first)
Store On Order	=, <, >, <=, >=, <>	Numeric
Store On Hand	=, <, >, <=, >=, <>	Numeric
Subclass	=, <, >, <=, >=, <>	LOV - Subclasses (Department and Class must be selected first)
UDA - Value Type	=, <>	LOV-UDAs (Value Type)
UDA - Date Type	=, <>	LOV-UDAs (Date Type)
UDA - Free Form Text	=, <>	LOV-UDAs (Free Form)
UDA Date Value	=, <, >, <=, >=, <>	Date (UDA Date Type must be selected first)
UDA Value	=, <, >, <=, >=, <>	LOV UDA values (UDA-Value Type must be selected first)
VPN	=, <, >, <=, >=, <>	LOV VPNs (Supplier must be selected first)
Weeks of Sales Exposure	=, <, >, <=, >=, <>	Numeric
WH On Order	=, <, >, <=, >=, <>	Numeric
WH On Hand	=, <, >, <=, >=, <>	Numeric

Price and Promotion Data

This section of the document provides details that pertain to all types of price events and helps provide insight on how to optimize and configure data to improve functionality in RPM.

Creating Price Events - Best Practices

When creating price events (price change, clearance, or promotions) for a group of items, it is recommended to enter the item data at the highest level possible: Merchandise Hierarchy (department, class, or subclass), Parent Item, Parent Diff Item, Item List, or using a user uploaded Price Event Item List. This is also true for the selection of locations at zone level versus store level. This provides the following advantages:

1. The future retail tables will have the ability to store data at the highest level possible which, in turn, will ensure that RPM will run as efficiently as possible, including response time for processing and screen flow. This means that if you can create price events at parent, parent/DIFF-1, or item list level versus transaction item level, data will be stored at the higher level. The same is true when selecting locations for your price event. If you create the price events at zone level and use the primary price zone, data will be stored at that level instead of having a record

at each store location within a zone. See more information in Primary Price Zone Benefits section.

2. Creating price events at parent differential level can also impact how data is stored on the RPM future retail tables. RPM will store DIFF-1 level data at higher than transaction level. However, RPM will not store at the higher level if you create your price event at DIFF-2 level.

    - Example: Parent item t-shirt with two differentiators: color and size. If you know that most of the time you are going to set the same price for items with the same color, you should make sure that color is selected as your DIFF-1 when setting up your items in RMS. If you create your price event using your DIFF-1, RPM will store data on the future retail tables at DIFF-1 level instead of transaction item (one row for each color).
    - Another example: Parent item yogurt with two differentiators: flavor and size. If pricing by size is most important, you would set up the item using size as the DIFF-1 in RMS. You would then create your price events using DIFF-1 (size).

3. Creating price events at higher than transaction item level will allow the system generated exclusion (SGE) process to be utilized during approval processing. See more details in the System Generated Exclusion section. Creating price events at the lowest level, transaction item or exploding an item list, will still identify conflicts, but will not automatically create exclusions, and your price event will remain in worksheet status until all conflicts are resolved.

    More examples:

    - If you are setting up a clearance for an entire style, create the data at parent item, or item list level. Do not explode down to transaction item detail.
    - When creating item lists, or price event item lists, include parent items when you can instead of transaction level items.
    - If you need to create price events at transaction item level, keep in mind the number of items you are selecting, such as the more items means slower response times.

Creating Price Events - Data Integrity

Data integrity is a critical component when using RPM. The system is data sensitive and it is important that you verify data integrity often, especially after a "go live" or an "upgrade". There is an existing batch job that can help with data validation. It is the status page batch program (statusPageCommandLineApplication.sh). This job, when run, performs data checks to verify that some of the assumptions that the application makes about the data are not violated. The checks are done with SQL counts and each check should return zero rows.

These are the data checks that are performed:

- **Missing department aggregations**-When departments are created in RMS, a row should be inserted into the RPM_DEPT_AGGREGATION table.
- **Missing primary zone groups**-Each merchandise hierarchy (department or lower) should have a row in the RPM_MERCH_RETAIL_DEF table.
- **Missing item/locations from future retail**-When an item is ranged to a location in RMS, a row should be inserted into the RPM_FUTURE_RETAIL table.
- **Duplicate future retail**-There should only be one row in the RPM_FUTURE_RETAIL table per item, location, and action date.

Note: For information see, Oracle Retail Price Management Operations Guide.

Viewing Full Column Detail for a Price Event

RPM was not designed to work as a reporting tool. Therefore, when customers use the "full column detail" feature to view details such as current cost or margin information, it is recommended that only one or two rows be selected before clicking the full column detail button. Using this functionality for events above the parent item level is not recommended.

This functionality should be used as a convenience to quickly see a snapshot of item data. Selection of more than a handful of rows will have an impact to system response times.

Price Event Exceptions/Exclusions

When creating a price event, users have the option to add an exception or exclusion to the price event.

- Exceptions allow for the detail of a price event to vary for a subset of the original price event detail at an item and/or location.

  For example you may want to create a clearance at style or parent item level for 25% off however you want some colors to have a greater reduction in price than the others. Creating an exception allows selected colors to be reduced at 40% off.

- Exclusions will remove the item and/or location from the price event. Below are details on exception/exclusion options by price event type.

The following price event types support item/location exceptions:

- Price Change (except Link Code type)
- Clearance
- Simple Promotions

The following price event types support item/location exclusions:

- Price Change (except Link Code type)
- Clearance
- Simple Promotions
- Finance Promotions

The following price event types support item exclusions only - location exclusions are not supported:

- Transaction Promotions

For more information on creating exceptions or exclusions by price event type see the *Oracle Retail Price Management User Guide*. RPM also has system generated exclusion functionality that can be setup using a system option. See detail in System Generated Exclusions section of this document.

System Generated Exclusions

When approving a price event, the conflict check process can automatically create system generated item exclusions based on a system option tolerance percent.

System generated exclusions may be created during approval process if conflict check is run and finds errors.

Price event types included in automatic exclusion process:

- **Price Change**-Regular and vendor funded (link code is not supported with this functionality).
- **Clearance**-Regular and vendor funded.
- **Promotions Components Supported: Simple or Threshold**-Including customer segment and vendor funded promotion components. (Multi-Buy, Transaction and Finance promotion components are not supported with this functionality.)

Note: Price event injector will support automatic exclusion creation.

The price event creation process remains unchanged until you reach the approval process. When Approve is selected, the conflict checking engine will automatically create exclusions based on a tolerance percent.

Price events must be created at higher than transaction-item level to be eligible for automatic exclusion creation.

Price events created at the following levels are eligible for automatic exclusion creation:

- Price changes and Clearance
  - Parent, Parent Diff/Location, or Zone
  - Item List/Location or Zone
  - Price Event Item List (PEIL)/Location or Zone levels
- Promotions- Simple or Threshold
  - Department, Class, Subclass/Location, or Zone
  - Parent, Parent Diff/Location, or Zone
  - Item List/Location or Zone
  - Price Event Item List (PEIL)/Location or Zone levels

To support and manage automatic creation of exclusions for price events, RPM will automatically create merchandise lists that will group exclusion details together into one detail row for display. The new system-created list is used only during the creation of the price event and will not be used with any other functionality.

System Generated Exclusions Tolerance %

A system option, System Generated Exclusions Tolerance % determines the percentage of transaction items on a price event that are allowed to error during conflict checking and still move forward with the approval of the price event. The system option will initially be set to default to 0% with a maximum tolerance value of 25%. The value can be updated in RPM.

The same tolerance percent value will be used for promotions, price changes, and clearances. If tolerance levels are exceeded, the price event approval will fail conflict checking, and the status of the price event will be set to Worksheet.

Tolerance values for conflict checking are based off of the number of transaction items that are processed. The number of locations does not impact whether or not the tolerance is met.

Maintain Approved System Generated Exclusions

System-generated exclusions will display in the multi-record block with one line representing all item exclusions. Users will review the auto-created exclusions to determine the next steps:

- **Option 1**-Fix data to eliminate the exclusion and re-process the data.
- **Option 2**-Leave the exclusions as created and no further action is required. When the price event becomes active, changes to the created exclusions are no longer allowed.

Upload a Price Event Item List

To support the process for creating price events at higher than transaction level, users have the ability to create 'ad hoc' price event item lists for one time use as they are keying a price event. This is done by uploading a spreadsheet during the item selection process of price event creation. The spreadsheet should be saved as a comma delimited file.

Price event item lists can be created for the following price event types: price change, clearance events, and promotion component types: simple, threshold, multi-buy, finance, and transaction, including customer segment and vendor funded.

When a price event item list is selected as the option for "item type", a new button named Upload Spreadsheet is enabled. This provides the ability to upload a previously created spreadsheet with the items that should be included in the price event item list. When the upload spreadsheet button is clicked, a pop up window will appear allowing the user to select a computer file using a browse feature or by manually keying in a spreadsheet file name for upload.

The price event item lists will be created for single use only at the time the price event is created or when a new component detail is added to an existing promotion. Users do not have the ability to re-use a price event item list in RPM. They do, however; have the ability to upload the same list multiple times.

The price event item lists are created in RPM based on the data from the uploaded spreadsheet. Users do not have the ability to make changes to the items associated to the item list once uploaded into RPM. If after uploading and prior to approving the promotion, users can remove the added list and re-upload the revised item list spreadsheet to create a new price event item list. If the price event has been approved, users will need to set the price event back to worksheet status and make necessary changes (remove old list and upload new list) or cancel the price event and create a new price event.

RPM will create price event item lists by validating the item numbers on the spreadsheet. For spreadsheets to be uploaded into RPM successfully you must ensure that no matter what other information resides on the spreadsheet, RPM requires valid item numbers in the first column. If the spreadsheet contains any other data in any other columns, they will be ignored. Valid item numbers require that the items have been set up in RMS and are in a status of "active".

The item number value can reflect parent items, transaction items, or a combination of both. Parent items with differentiators (for example, color/style) are not valid for price event item lists. In the event that a child item for a parent is included in the list along with the parent item, the child item will be removed from the list to ensure that there is no duplication of data. Price event item lists supports having multiple levels of the hierarchy represented in a single list. For example, users can create a file that has item numbers from more than one department, class, or subclass.

Note: Merchandise hierarchy levels (department, class, and subclass) are not valid values for the spreadsheet. Options are parent items or transaction items.

When the price event item list is uploaded, RPM will validate that all of the items included in the spreadsheet are valid RMS item numbers. If items are not found in RMS, or duplicate items are found on the spreadsheet during processing, they will be removed from the creation of the price event item list. RPM will also validate user security. If a user does not have valid security for any items on the spreadsheet they will be removed from the creation of the price event item list. Users will receive a message informing them that some of the items from the upload spreadsheet have been removed.

Warning: Not all items uploaded- duplicate or invalid items found.

Note: Items in error will not be displayed. Users are responsible for spreadsheet content and ensuring data accuracy.

Price event item lists in RPM do not exist as item lists in RMS. RPM will not send price event item list information to RMS.

Users will be able to search for price events (for price changes, clearances, and promotions) using an existing price event item list ID.

Price Event Item List Clean Up:

- **Unused price event item lists**-Batch logic searches for existing price event item list IDs and find any that are not tied to a price event. These price event item lists will be removed from the system.
- **Purging price events**-Batch logic that purges old price events will also purge any price event item lists that are connected to the price event being purged.

Note: For more information on the price event item list batch jobs, see the *Oracle Retail Price Management Operations Guide*.

Custom Attributes in RPM

This feature provides the ability to create and assign custom attribute values to price events. Customers have the ability to add/view/maintain attributes for the following price event types in RPM; price changes, clearances, and promotions. Custom attributes also are supported when creating price events via the price event injector. Custom attributes can be used to support expanded client functionality such as coupon information, clearance markdown phase or to capture additional information from legacy systems.

Customers are responsible for the set up and configuration of the custom attribute framework along with loading them to the RPM tables. Setting up the custom attribute framework is a backend process that will need to be managed by a system administrator. There is no UI access to create or maintain the custom attribute framework. See the *Oracle Retail Price Management Operations Guide* for information on setting up the framework.

Multiple tables support storing custom attributes for each type/level of price event:

- Price Change Custom Attributes
- Clearance Custom Attributes
- Promotion Event Custom Attributes
- Promotion Header Custom Attributes
- Promotion Component Custom Attributes
- Promotion Component Detail Custom Attributes

When creating the custom attributes framework, there are a maximum of 22 attributes that can be set up for each price event table. Of those 22 attributes, 10 are allowed to be character based attributes, 10 number based attributes and 2 date based attributes.

The following information needs to be defined when planning the attribute framework:

- **Custom Attribute ID #—**Identifies the type of custom attribute
  - ID's 1-10 are defined as character based
  - ID's 11-20 are defined as numeric
  - ID's 21 and 22 are defined as date based
- **Label Name—**Field name identifier that will display on the custom attribute window. The UI will support displaying 60 characters for the label name.
- **Display Sequence**—Indicates the order in which the attributes should be displayed in the attribute screen from top to bottom. Attributes will be displayed in a single column on the screen.
- **Enabled**–Indicates whether the attribute should be displayed in the UI or not; only enabled attributes will display for update in the attribute screen. If a system administrator changes an existing attribute from enabled to disabled or deletes the attribute, the UI will no longer display the values. The attribute, or any values assigned to the attribute during custom attribute creation will remain on the RPM tables. If required, custom reporting could be created to read all data from the tables.

Once the custom attributes framework has been added to the price event tables and enabled, they become available for use during price event creation. As price events are created or maintained, users will have access to assign custom attribute values by entering them manually in the UI.

When making changes to or adding custom attribute values to an already existing price event, maintenance is allowed based on the status of the price event:

- **Price Changes**—Must be in worksheet status
- **Clearance—**Must be in worksheet status
- **Promotion Event—**No status applies, changes are allowed anytime
- **Promotion Header**—All component details attached to the promotion header must be in worksheet status

- **Promotion Component—**All details attached to the component must be in worksheet status
- **Promotion Component Detail—**Must be in worksheet status

RPM will provide wrapper logic for custom validation of custom attribute values, providing the price event type along with the custom attribute values. Specific validation rules will require a custom package function to be written by the system integrator.

Below are some examples of possible validation rules that could be set up by a customer. Another validation option would be to tie into RPM conflict check by setting up custom conflict check rules. See the *Oracle Retail Price Management Operations Guide* for more information on setting up custom rules.

Validation examples:

- Validation on date requirements; customers may choose to only allow dates based on certain rules such as only allowing a future date versus a date in the past.
- A customer may choose to provide a list of valid values for an attribute, requiring validation to ensure the correct values are used. Although the user interface editor is freeform, the validation rule would need to contain a list of values to validate against.
- There may be a need to have one attribute dependent on another, requiring validation to ensure both fields are entered.

If any custom validation returns error messages, the UI will display all errors returned to the user. However, when this same validation is called by the price event injector batch, only the first error will be recorded on the associated staging table.

> **Note:** Reference the *Oracle Retail Price Management Installation Guide* for more information on configuring custom attribute framework.

Custom Attributes via the Price Event Injector

Attributes can be assigned during price event creation via the RPM price event injector, adding attributes to previously created price events is not supported (considered maintenance - can be done via the UI). All price event types that are supported by the price event injector will also support adding custom attributes. The following price event types will support custom attributes via the price event injector: price change create, clearance create and promotion create, assigning custom attributes at multiple levels; event, header, component and component detail. All promotion component types are supported; simple, threshold, multi-buy, transaction, and finance.

When adding custom attribute values via the price injector, no validation will be done to ensure the custom attribute framework exists on the price event attribute tables. Custom attributes loaded that do not match up with the custom attribute framework on the tables, will still flow to the underlying tables, however data will only display in the UI if the attributes structure is enabled. The values assigned to an attribute are free-form and will not be validated unless customer has set up specific validation rules.

Emergency Price Events

Emergency or Same Day Price Event Changes - There is a system option called price change processing days that is set to designate the number of days required between the create date and the effective date of a price event. This rule ensures that price changes, clearance and promotions are created with enough advance timing that stores and other process areas can react accordingly.

A separate security level has been created to give some users the ability to skip this rule and create a price change that is effective immediately. This is usually done as an emergency measure to update the price of an item that is incorrect. Users without the special security will receive an error message if the effective date of a new price event falls before the number of price change processing days.

For example, if the setting for price change processing days is "3" days. The system will enforce a conflict check rule that a price event cannot be created less than three days out unless user has emergency security

> **Note:** See more details on application security in the *Oracle Retail Price Management Operations Guide*.

When an emergency price event is created the information is passed to POS as soon as the price event is approved. The communication to RMS will also be done on approval; however there is an exception to the timing on when RMS will get the information. Batch processing will be done as a bulk or chunk process; if the price event is large the system will process it via the chunk batch. Price events processed by the chunk batch will update nightly unless the trickle feed batch is also run to update RMS. See the *Oracle Retail Price Management Operations Guide* for more information on the trickle feed batch process.

Price Changes

Price changes are the pricing events in RPM that affect the regular retail price. When a price change is created, the following information is specified:

- The item receiving the price change
- Where the price change is occurring
- How the price is changing
- When the price change will take effect
- Why the price change is occurring

When price changes are approved in RPM, they are made available to multiple systems for ticketing and inventory valuation purposes.

There are multiple options for creating price changes in RPM:

- **Regular Price Change**-Used to change the price of an item/location on a specified date.
- **Vendor Funded Markdown**-Deals created in RMS can be associated to price changes (including vendor funded promotions) in RPM.
- **Link Code Price Change**-Link codes should be used for identical items that will always share the same retail price (for example, a retailer wants all 8 oz. packages of frozen vegetables to carry the same price regardless of type). They are used for ease of data entry and considered a point in time price change. See more details in section Link Codes Functional Overview.
- **Multi-unit Price Change**-Multi-unit pricing allows you to manage prices that vary based on the number of units purchased by the customer. Retailers may give customers the option to purchase a single unit of an item at one price, and a case of

the same item at a lower price than the single item price times the number of units. For example: Fireplace logs- sold individually for $5.00 each, and case of 6 logs sold for $22.00 (versus $25.00).

> **Note:** This functionality is not the same as promotional pricing based on price reductions due to purchase quantity.

- **Price Change with Exceptions or Exclusions** -Both options are available for all price change types except link code price changes.

RPM provides flexibility on item and location selection when creating price changes. Users have the option to create price changes at the following levels:

- Item Selection
  - Parent Item
  - Parent Diff Item
  - Transaction Item
  - Item List
  - Price Event Item List
- Location Selection
  - Zone
  - Location (warehouse or store)

When multiple regular prices are passed for an item, the last regular price passed is the regular price that is used. Normally, only one price change for an item location on a given day is allowed to be created via the user interface in RPM. There can be more than one regular retail price for an item location on the same day only when an "emergency" retail price change is entered and executed by an authorized emergency user. At POS the latest price received (emergency price) is the only "regular" price.

If a price change is created and approved for an item/location that is on an active clearance, the new price change will not take effect in RPM until the clearance is reset (on the reset date). If the item/location that received the price change has a subsequent clearance that falls after the price change is approved, the clearance will be recalculated based on the new item regular retail.

If a price change is created and approved for an item/location that is on an active promotion, the new price change may impact the selling retail of the promotion. Updates to the promotion selling retail will be based on the type of promotion and the defined discount and applied on the effective date of the price change.

For example:

- Simple Promo - 10% off defined item

  Regular Retail = $10.00 - Promotional Retail = $9.00

  If a price change is entered for the defined item, raising the price to $12.00, the promotion selling retail will also change to $10.80 on the effective date of the price change.

Clearances

Clearances in RPM are defined as a reduction in permanent retail designed to increase demand and move inventory out of a store. These reductions are represented in RPM as clearances which may consist of a single markdown or a series of markdowns. When a clearance is created, the retailer is specifying the items and locations where the clearance is in effect along with the discount or set price for the markdown. Subsequent clearances always need to result in the price of an item decreasing. Clearances may be ended by using the reset function, but if never reset, the items remain on clearance in the specified location(s) indefinitely. If a reset is specified it is considered to be the final step in the clearance.

There are multiple options for creating clearances in RPM:

- **Regular Clearance** -Used to markdown the price of an item/location on a specified date, such as end of season holiday merchandise or swimsuits at the end of summer season. Clearance is also used to phase out a color or pattern for an item.
- **Vendor Funded Markdown**-A deal type of "Vendor Funded Markdown" created in RMS can be associated to clearances in RPM to offset the reduction of the permanent price change. The amount to be billed back to the vendor will be calculated based on the stock on hand when the clearance is executed on the effective date.
- **Clearances with Exceptions or Exclusions**-Both options are available when creating clearance. Exceptions allow item locations to have different clearance prices from zone. Exclusions remove the item location completely from a zone level clearance.

RPM provides flexibility on item and location selection when creating clearance. Users have the option to create a clearance at the following levels:

- Item Selection: Parent Item, Parent Diff Item, Transaction Item, Item List and RPM Price Event Item List
- Location Selection: Zone and Location (warehouse or store)

Deleting a Clearance

To end an executed clearance a reset date should be added to the clearance, the option to delete is not allowed.

To end an approved clearance, the option is available via the Maintain Clearance screen. The approved clearance must first be set back to worksheet status, which will kick off conflict check and send a message to consuming systems to remove the clearance. Once in worksheet status, the clearance can be deleted.

If a reset date has been set up for the clearance that is being deleted and only one clearance exists the reset date will be removed when the clearance status is updated to worksheet. If a reset date exists and there are multiple clearances for the same item/location, the reset date will be removed for the clearance that is being deleted however it will not be removed for any other existing clearances. RPM will not remove the reset date as there could be multiple dates that exist at the item location level. Customers will need to manually remove the reset date by clearance to meet their business process needs.

Below find examples of the communication sent to consuming applications for deleting a clearance.

Table 10–1 Scenario: Multiple Clearances -- with same reset dates

#	RPM User Interface	Integration to Consuming Systems
1	Newly created CLR 1 (1st markdown) Approved or Executed Effective Date = JAN/01 Reset Date = MAY/01	Create CLR message sent Create CLR Reset message sent Details at item/location level for both messages
2	Newly created CLR 2 (2nd markdown) Approved Effective Date = MAR/01 Reset Date = MAY/01 NOTE: reset date must be greater than the start date of the 2nd markdown	Create CLR message sent No message sent related to clearance reset data because the reset date remains the same. Details at item/location level for both messages
3	Un-approve CLR2 This step is required to delete	Delete CLR message sent at item/location level Note: CLR reset delete is not sent, the last applied reset date remains as MAY/01 for CLR1
4	Delete CLR2 (2nd markdown)	n/a - no messages are sent

Table 10–2 Scenario: Multiple Clearances with different reset dates

#	RPM User Interface	Integration to Consuming Systems
1	Newly created CLR 1 (1st markdown) Approved or Executed Effective Date = JAN/01 Reset Date = MAY/01	Create CLR message sent Create CLR Reset message sent Details at item/location level for both messages
2	Newly created CLR 2 (2nd markdown) Approved Effective Date = APR/01 Reset Date = JUL/01 NOTE: reset date must be greater than the start date of the 2nd markdown	Create CLR message sent Update CLR Reset message sent Details at item/location level for both messages
3	Un-approve CLR2 This step is required to delete	Delete CLR message sent at item/location level Note: CLR reset delete is not sent, the last applied reset date remains as JUL/01 for CLR1
4	Delete CLR2 (2nd markdown)	n/a - no messages are sent

Table 10–3 Scenario: Multiple Clearances with reset dates removed (CLR2)

#	RPM User Interface	Integration to Consuming Systems
1	Newly created CLR 1 (1st markdown) Approved or Executed Effective Date = JAN/01 Reset Date = MAY/01	Create CLR message sent Create CLR Reset message sent Details at item/location level for both messages

Table 10–3 (Cont.) Scenario: Multiple Clearances with reset dates removed (CLR2)

#	RPM User Interface	Integration to Consuming Systems
2	Newly created CLR 2 (2nd markdown) Approved Effective Date = APR/01 Reset Date = none (blank)	Create CLR message sent Delete CLR Reset message sent Details at item/location level for both messages
3	Un-approve CLR2 This step is required to delete	Delete CLR message sent at item/location level Note: CLR reset delete is not sent, the last applied reset date remains as "null" for CLR1
4	Delete CLR2 (2nd markdown)	n/a - no messages are sent

Clearance Reset Rules and Functionality

When a clearance is created, setting the clearance reset date is optional. Clearance reset is used for two functions. The first returns the item to the last regular selling retail price when the clearance has ended. If an item is being discontinued and will no longer be in stock after the clearance, a reset date may not be used. The item will remain at the last clearance retail as long as it exists in the system. The second function of the reset date allows purge programs to perform defined database maintenance.

Reset dates are stored at the item location level in RPM and will link to all clearances for the item/location. The functionality of adding, updating, or removing a clearance reset for an item location will be validated upon approval/apply via the user interface. The system will ensure that the reset date is not less than or equal to the last approved clearance's effective date for the item/location. All clearances set up for the item/location will be validated and the reset date will be linked to all clearances that exist for the item/location.

When entering/editing a reset date on an existing transaction/location level item, on apply the system will validate that the reset date for the item/location is not less than or equal to the last approved clearance's effective date for the item/location.

When entering/editing the reset date for any level above item/location (for example, parent item, item list or zone), on apply the system will ensure that the reset date is valid for every transaction level item/location combination represented by the higher level being updated. For each item/location the reset date must not be less than or equal to the last approved clearance's effective date for that item/location.

A reset date must be greater than the current system date (vdate) + the price change processing day's system option defined in RPM. This system option is set to designate the number of days required between the create date and the effective date of a price event. Since setting a clearance reset date sets the item location back to the last selling retail it is the same as a "price change" and must follow the system option rule. This rule ensures that price events are created with enough advance timing that stores and other process areas can react accordingly. A separate security level has been created to give some users the ability to skip this rule and create a price change or end a clearance effective immediately or same day. Users without the special security will receive an error message if the effective date of a clearance reset date before the number of price change processing days.

When a clearance reset date is met, the item/location will be set back to the last selling retail on the reset date. If an update to the retail is required, a price change will need to be entered in RPM to reflect the updated retail. If a price change for an item/location that is already on clearance is approved while the clearance is running, the new price change will take effect when the clearance is reset (on reset date). If the item/location that received the price change has a subsequent clearance that falls after the price change is approved, the clearance will be recalculated based on the new item regular retail. When the clearance is executed, the retail of the clearance cannot be recalculated.

If a clearance does not have a reset date associated with the items/locations, it will not be fully purged from the system. When conflict check runs, the future retail purge logic will find clearance related records and attempt to remove historic clearance data. Items/locations on the RPM_FUTURE_RETAIL that are outside the clearance retention period will be purged, however the purge process will retain one record that will remain on the RPM_FUTURE_RETAIL until a clearance reset date has been set.

Adding a Clearance Reset Date

Users have two options for managing the end of a clearance if they have not entered a reset date during the create process, Maintain Clearances and Create Clearance Resets. Below find more details on each process:

Maintain Clearances

Access the existing clearance via the Maintain Clearances screen and search for the clearance you want to update. Enter the necessary reset date information and click the apply button. You can add a reset date for an approved or executed clearance; it does not need to be set back to worksheet status. Adding or updating a clearance at this level will update each item location on the selected clearance with the reset date. If any of the item locations are also attached to another clearance, a reset date will be associated to the clearance for those item locations.

Create Clearance Resets

Access Create Clearance Resets and add/update the reset date by item(s)/location(s) independent of the clearance. This option allows you to select the item(s) and location(s) that require a reset date vs. setting the date for all items on a clearance. On apply the system will add/update the reset details for the selected item(s) and location(s). This may result in having a clearance where only some of the item locations will be reset.

The create clearance reset process provides flexibility on item and location selection. Users have the option to create a clearance reset at the following levels:

- Item Selection: Parent Item, Parent Diff Item, Transaction Item, Item List, Price Event Item List
- Location Selection: Zone or Location (warehouse or store)

Removing a Clearance Rest Date

If a reset date already exists for an item/location on clearance, users have the options to remove the reset date.

Maintain Clearances

When you view a clearance in the maintain clearance UI, after highlighting a row in the table, the reset date field in the Clearance Maintenance section is unprotected and blank. If a reset has been setup for one or more items on the clearance the Earliest Reset Date and Latest Reset Date fields will display the reset values. These dates could vary as resets can be set up by item location.

Note: The reset date will only display in the Reset Date column on the table if the clearance was created at transaction item/location level as dates can vary by item.

To remove the reset date(s) click the apply button at the bottom of the screen, leaving the reset date field blank. This will kick off conflict checking and if there are no conflicts found, will remove the reset date for all item locations on the clearance and any other clearances for the item location.

Create Clearance Resets

This function also supports removing a reset date for an item location independent of the clearance. To remove a reset date, leave the reset date field blank, enter item and location detail and apply the change. This function will remove the reset data for any clearance(s) set up for the selected item locations.

Below find examples of the communication sent to consuming applications when a reset date is removed from a clearance.

For this example, multiple clearances are set up for the same item(s)/location(s). This process works the same for both maintenance options; Maintain Clearance and Create Clearance Resets.

Table 10–4 Scenario: Removing a Reset Date a Clearance Level

#	RPM User Interface	RPM Internal Handling	Integration to Consuming Systems
1	Newly created CLR 1 Approved or Executed Effective Date = JAN/01 Reset Date = none (blank)	CLR record created CLR Reset record created with "null" effective date	Create CLR message sent Details at item/location level
2	Newly created CLR2 Approved or Executed Effective Date = APR/01 Reset Date = none (blank)	CLR record created CLR Reset record created with "null" effective date	Create CLR message sent Details at item/location level
3	Add Reset Date to CLR2 Reset Date = JUN/01	CLR Reset record updated to JUN/01	Create CLR Reset message sent at item/location level
4	Remove Reset Date from CLR2 Reset Date = none (blank)	CLR Reset record updated to "null"	Delete CLR Reset message sent at item location level Note: Reset delete impacts both CLR1 and CLR2

Promotions

Promotions are events in RPM that discount the price of an item for a defined amount of time. Promotions are set up to apply to the regular retail price, the clearance retail price, or both, and when the promotion ends, the price reverts back to the retail price or clearance retail if the item/location is on clearance. When a promotion is entered in RPM, the retailer specifies the duration of the promotional price, what kind of promotion takes effect, and to which items/locations the promotional price applies.

> **Note:** "Apply To" details assigned to a promotion will result in a filtered list of items that is sent to the consuming systems. This applies to all promotion types with an exception of Finance Promotions. Even though Finance Promotions have the Apply To value of a regular retail, clearance retail, or both, the item details are not filtered based on the setting. Systems that consume this promotion type need to filter promotion details based on the Apply To settings.

RPM supports multiple types of promotion component creation options which are grouped in categories of "simple" and "complex". Of the promotion component types available, only simple falls under the simple category. All other promotion component types; threshold, multi-buy, finance, and transaction are treated as complex. See more details on each promotion type in the following sections.

An additional feature when creating a promotion is the ability to attach a customer segment identifier to the promotion component. Customer segment set up is done in the RMS system. Customer segments can be used for multiple purposes; defining customer profiles such as soccer moms, college students, employee discount perks. When creating a promotion in RPM, you have the option to attach a customer segment type to the promotion. RPM will display a list of valid values via the LOV button or customer segment ID can be keyed on the user interface. This information is sent downstream to the POS system, so that when a customer has proper identification for the customer segment they will receive the promotion discount.

> **Note:** There is a conflict check rule that will enforce that there can be only one customer segment promotion for the same item/location/customer segment ID on any date. This rule is in place for all promotion component types (simple, threshold, multi-buy, transaction) with the exception of finance components. Finance components have no impact to the RPM_FUTURE_RETAIL table which is required to enforce the rule.

For simple promotions, RPM will calculate the promotional retail and update the data on the RPM tables along with making the information available to consuming systems. RPM does not calculate the promotional retail for complex promotions because the discount amount is determined by what is included in the transaction at the point of sale. RPM tables are updated with complex promotion details including the items/locations on promotion; however, no promotional retail is stored. Promotion details for complex promotions are made available to consuming systems for processing.

> **Note:** Because RPM does not calculate the promotional retail based on complex promotions, it is possible to create a complex promotion that results in a negative retail. Users must review the data being created to ensure the promotion does not result in a negative retail.

RPM provides flexibility on item and location selection when creating promotions. All of the following promotion types allow the following levels for item/location selection. See each promotion type for more details.

- Item Selection

  - Merchandise Hierarchy (Dept, Class, Subclass)

- Parent Item

- Parent Diff Item

- Transaction Item

- Item List

- Price Even Item List

- Location Selection

  - Zone

  - Store (warehouse not valid for promotions)

Simple Promotion Component

A simple promotion component consists of an item, item group, or merchandise level that receives a discount at a specific location or group of locations when the customer purchases an item. The discount can be defined as; amount off, percent off or by setting a fixed price.

- **Amount Off**-Buy Flat Screen TV @ $10.00 off.
- **Percent Off**-Buy Flat Screen TV @ 10% off.
- **Fixed Price**-Buy Flat Screen TV (regular retail = $500.00) for $450.00.
- **No Change**- Promote an item at its regular retail
- **Simple Promotion Components with Exceptions or Exclusions** - Both options are available when creating simple promotion components.

Threshold Promotion Component

A threshold promotion component consists of an item, item group, or merchandise level that receives a discount at a location or group of locations when the customer purchases a quantity or an amount of an item. Users must define the threshold levels before they can create the threshold component. See the *Oracle Retail Price Management User Guide* on details for setting up a promotion threshold.

It is important to understand that RPM will create a row in the item selection table for each item entered during the item selection process. The promotion component detail will be sent to consuming systems as individual promotions - one for each detail row.

For example:

When creating at transaction item level:

During item selection 3 transaction items are selected/entered with threshold parameters defined as buy a quantity of 3, get 10% off each item.

- If a customer purchases 3 transaction items - 1 of each item - the promotion discount will not be applied
- If a customer purchases 3 transaction items - 3 of one of the transaction items - the discount of 10% is applied

When creating at parent Item level:

During item selection 3 parent items are selected/entered with threshold parameters defined as buy a Quantity of 6, get 20% off each item.

- If a customer purchases 6 child items - 2 under each parent item - the promotion discount will not be applied.

- If a customer purchases 6 child items - all under the same parent item - the discount of 20% is applied.

In the examples above, the threshold parameter quantity must be met for each item or in other terms, for each detail row on the item selection table. To create a threshold promotion component for a group of items, enter them in the UI using an item list or a price event item list. This will create one row of item selection detail on the table which will create one promotion for all selected items.

Threshold level can be defined as a single threshold where a customer must purchase an amount or quantity to get a discount. Threshold promotions can also be created with multiple threshold levels where based on the amount or quantity purchased by the customer the discount increases based on threshold details. When creating the threshold details for a threshold promotion component the qualification type will be set to one of two values; threshold level or item level threshold. Below find examples on the different handling for each setting.

- Consider a regular price item of $10, buy 3 pay $7.50 each

  If the items selected for the promotion are Items A, B, and C;

Qualification Type = Threshold Level	Qualification Type = Item Level Threshold
The promotion is applied if I purchase;	The promotion is applied *only* if I purchase;
AAA, AAB, AAC, ABB, ABC, ACC, BBB, BBC, BCC, or CCC	AAA, BBB, or CCC

- Single threshold, qualifier = threshold level.
  - Buy six pairs of jeans; get 10% off each pair of jeans.
  - Any combination of designated items will qualify to receive the reward (discount) as long as a total of six units are purchased. The rules will work the same for single or multiple threshold promotions.
- Single threshold, qualifier = item level.
  - Spend $100 on jeans; get 10% off each pair of jeans.
  - Customer must purchase six pairs of jeans of at least one of the designated items, they cannot mix and match items to meet the threshold. The rules will work the same for single or multiple threshold promotions.
- Multiple threshold, qualifier = threshold level.
  - Buy four t-shirts; get 10% off each t-shirt.
  - Buy eight t-shirts; get 20% off each t-shirt.
  - Buy twelve t-shirts; get 30% off each t-shirt.

Multi-buy Promotion Component

A multi-buy promotion allows you to define the quantity of items or amount of purchase required for the customer to receive a discount or reward. For example, a promotion might include a reward of a third pair of shoes free with the purchase of any two pairs of shoes from selected items or item groups. Another promotion might allow purchase of any five items from selected item groups at a fixed price or discount. All multi-buy promotion components share common characteristics, and the user creates them in RPM using the same screen flow. Any multi-buy component can be considered as one of the following general types. See the *Oracle Retail Price Management User Guide* for more details on each of the following types:

- **Multi-buy "meal deal"**-Buy one hamburger sandwich, an order of French fries, and a soda for a fixed price of $5.00.
- **Multi-buy link saver**-Buy a cap and a pair of gloves; get $2.00 off of a winter scarf.
- **Multi-buy "cheapest free"**- Buy three pairs of shoes, get the cheapest pair free.
- **Multi-buy with multiple buy or reward (get) lists**-When you buy one jacket and one pair of slacks, you can get one shirt or two ties at a fixed price of $5.00.
- **Multi-buy with AND or OR conditions for buy and reward lists**- This example works the same as the example above: buy a hamburger or a chicken sandwich and an order of French fries, get a soft drink or ice cream free.

The following are some examples of multi-buy promotions that can be created using the "and/or" conditions, such as use the "and/or" qualifier between buy lists and/or reward lists. And/or conditions are required when creating more than one buy list or more than one reward list.

- Buy 1 or more items in item list A **or** item list B.

  Get discount on item list X **or** item list Z.
- Buy 1 or more items in item list A **and** item list B.

  Get discount on item list X **and** item list Z.
- Buy 1 or more items in item list A **or** item list B.

  Get discount of item list X **and** item list Z.
- Buy 1 or more items in item list A **or** item list B **and** item list C.

  Get discount on item list X **or** Z.

Multi-buy promotions will also support the following features:

- **Price Ranges**-This feature is optional and is available when creating multi-buy promotion component types with a single buy list and/or a single reward list. Price range is not valid if promotion has more than one buy list and/or reward list.

  The user interface allows entry of a minimum limit and/or a maximum limit for items associated to a buy list and/or reward list.

  - If min/max values are entered for a buy list, customers need to purchase either the amount or quantity specified on items that fall within the min/max price range in order to receive the reward.
  - If min/max values are entered for a reward list, customers will receive the reward if they select an item from the designated reward items that fall within the min/max value.

    The price range is defined based on the purchase of a single item; the reward or discount will not be applied on the sum total value of multiple items. For example, if the price range defined for the buy list items is $45-$100, and the reward discount is 10%, the reward is valid if the price of the units purchased fall within the defined price range. The reward will not be given if the purchase of combining two units of an item falls within the price range.

    RPM will not perform any additional validation between the item selection and the price ranges entered in the user interface. For example, RPM will not validate that the items in the selected list fall within the entered price ranges.

  - Buy one pair of jeans at regular price: price range = $45.00 - $100.00;

    Get one t-shirt free: price range = $0.00 - $25.00.

- **Promotion Limits**-This feature is optional and is available when creating multi-buy promotion component types. The discount limit field is a user entered numeric value to indicate the number of times a promotion is applied to a customer's purchase. For example:

  - Purchase a newly released DVD; get $5.00 off, limit one discount per customer. If customer purchases two of the same DVD, they will still only get $5.00 off one of them.
  - Buy two pairs of jeans and get one belt free. For this example, the customer is only allowed one free belt no matter how many pairs of jeans are purchased (and as long as two pairs of jeans are purchased).

Finance Promotion Component

This type of promotion provides users the ability to create promotions that use credit details along with threshold rules that would give the customers an option to pay back the credit card company without any interest (0%) in a defined timeframe. Customers would not be charged interest if paid in full within the defined duration.

Finance details are set up prior to creating the promotion including:

- Credit card type.

  Bank Name- For the credit card issuing bank.

  BIN- Bank index numbers.

  During promotion create users will add threshold details which is the total amount that the customer has to spend in order to qualify for the finance promotion.

  Spend $1000 using VISA; receive 0% interest, duration = 18 months.

- **Finance promotion component exclusions** - This promotion component type supports creating exclusions only. Item or location exceptions are not supported.

> **Note:** A Finance Promotion provides the ability to set Apply To value of Regular Retail, Clearance Retail or both (regular and clearance). But the item details tied to the promotion are not filtered based on the settings; all designated items are attached to the promotion. Customization is required if consuming systems that consume this promotion type require the promotion data to be filtered based on the Apply to setting.

Transaction Promotion Component

A transaction promotion component can have a discount type of either "amount off" or "percent off" that will be taken off the entire purchase (shopping basket) versus specific items in a transaction. This promotion has the same look and feel as setting up a multi-buy promotion; the main difference is that the reward given to the customer is a discount on their entire purchase amount. This type of promotion can be set up at storewide level or by designating specific items to purchase to receive the discount.

- **Storewide**- Spend $200 on any merchandise in store, get $20.00 off entire purchase or order $200.00, get 20% off entire purchase.

- **Merchandise Hierarchy**- Spend $100 on Jeans **or** Khakis, get 10% off entire purchase.

Both options above refer to purchasing an amount to receive the reward or discount. Users will also have the option to designate a "quantity" to purchase, such as, buy 10 items in the store, get 10% off entire purchase. This option is available for both storewide or by designating a lower level of item selection.

Transaction promotion component item exclusions - This promotion component type supports creating item exclusions only. Location exclusions along with Item or location exceptions are not supported. There are specific rules that must be met on the buy list details, the create/view exclusion button will enable/disable based on the rules.

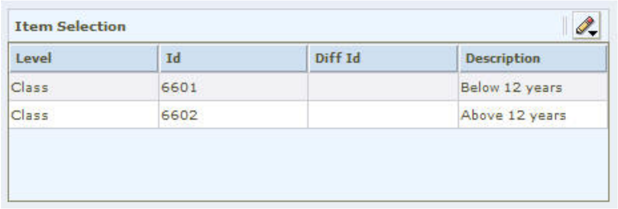
The button will enable when the following conditions are met:

- A single buy list has been created for the promotion component, a buy list must exist before exclusions can be created.
- The buy list can contain one or many detail rows however, if there are multiple details rows in the buy list item selection multi-record block, each row must have been created at the same merchandise hierarchy level.
- Buy list is created at higher than transaction item level.

The screen capture below shows an example of item selection for the buy list done at class level. The "level" field in the multi-record block must be the same to allow exclusion creation.

- In the example shown, the "level" column has Class for both entries--this example supports exclusion creation.

Figure 10–2 Item Selection for the Buy List at the Class Level



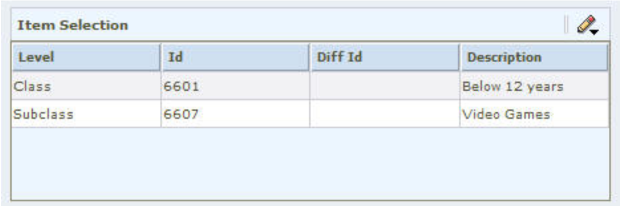
Item Selection

Level	Id	Diff Id	Description
Class	6601		Below 12 years
Class	6602		Above 12 years

Note: Assumption for this scenario is that the departments could vary as long as the "level" is the same (for the example shown, assume that class 6601 is tied to department 10 and class 6602 is tied to a different department.

- This next example is not "supported". The level columns shows Class and Subclass.

Figure 10–3 Item Selection for the Class and Subclass Level



Item Selection

Level	Id	Diff Id	Description
Class	6601		Below 12 years
Subclass	6607		Video Games

The button will be disabled under any of the following conditions:

- No buy list has been created for the promotion component.
- Buy list has been created that contains more than one row of detail in the item selection multi-record block with varying merchandise hierarchy levels (for example, one row at department, one row at class).
- Buy list is created at transaction item level.
- More than one buy list exists.
- Promotion component is in any status other than worksheet and no exclusions exist.

Once item exclusions have been created, changes or modifications to the buy list item selection detail are not allowed. When a check appears in the checkbox for "item exclusions assigned" the Delete button to remove a buy list detail and the Add Buy List will be disabled. To modify existing buy list item selection detail, exclusions must first be removed.

Time-Based Promotions

Retailers need the ability to set up time-based promotions, such as Happy Hour and Door-Buster promotions. Users can enter start and end times, in hours:minutes, when entering start and end dates while setting up or maintaining promotions. Adding this data at the component-level detail allows for business validations, such as conflict checking. When a promotion is approved, the information is sent to consuming systems. Start and end times can be added to the following promotions types, including customer segment and non-customer segment promotions:

- Simple promotions
- Threshold promotions
- Multi-buy promotions
- Transaction Promotions

You can also load time-based promotions using the price event injector batch. This functionality is supported for all promotion component types.

When creating promotions, the start and end times shown at header level are protected and defaulted to a start time of 12:00 A.M. and end time of 11:59 P.M. Time-based elements are only added at the component detail level. This allows users to create multiple component details for a promotion with varying start and end times. There is a column that indicates time-based promotion in the multi-records blocks for promotion search results, promotion component header, and the promotion component detail maintenance. The indicator is checked if the promotion has any components that are set up as time based.

Time-based elements for a promotion are based on the time zone for the store location. A promotion that starts at 8:00 A.M. in New York will also start at 8:00 A.M. in California.

Users are also able to enter start and end times when creating emergency promotions. However, RPM is not designed to recognize time as it does for selecting promotion dates (based on business rules), making it possible that the start time for an emergency promotion could be in the past; there are no validation checks to prevent this scenario.

For example, a user creates an emergency promotion to start at 8:00 A.M., ending at 1:00 P.M., but doesn't enter the data into RPM until 9:30 A.M. The system will reflect the entered time values of 8:00 A.M. - 1:00 P.M., sending the data consuming systems.

Start and end times will default at the component detail level to 12:00 A.M. start and 11:59 P.M. end if the user does not enter time values.

Price history in RMS only has the ability to reflect one price value for an item location and date. If multiple overlapping promotions exist for the same item location RMS will reflect the promotion with the latest start time. Because of these rules, it is possible that RMS and RPM current retail could be out of sync.

Example 1: An item location has two overlapping promotions:

- Early Bird promotion starting at 6:00 A.M. until 10:00 A.M.
- All day promotion starting at 8:00 A.M. until 10:00 P.M.

  RMS will reflect the retails for the all day sale because it starts later than the Early Bird sale.

Example 2: An item location has two overlapping promotions; only one is time based:

- Early Bird promotion starting at 6:00 A.M. until 10:00 A.M.
- Week long promotion, non-time-based 12:00 A.M. - 11:59 P.M.

  RMS will reflect the retails for the Early Bird promotion (during the overlap day), because it starts later than the week-long promotion.

  The only discrepancy to this rule is when an emergency promotion is created. The emergency promotion will override any other existing promotions.

> **Note:** Emergency promotions that are processed through bulk will follow this rule. If they are processed through chunk conflict check processing the updates will only be applied during the day if the Price Event Execution for Emergency Chunk data batch is run. If the batch is not run, data will not be applied until next day. See the *Oracle Retail Price Management Operations Guide* for more information on this batch.

Canceling a Time-Based Promotion

Users will have the ability to cancel a time-based promotion. When an active promotion is canceled in RPM, the system will end the promotion at 11:59 P.M. on the current date (Vdate) regardless of what time of the day the promotion was canceled. This process works the same as canceling a non-time-based promotion. It is not realistic for store locations to support real-time promotion cancellations as they would have to be able to react and update promotion signage in-store.

Overlapping Promotions

Overlapping promotions are two or more promotion details where the same item/location is on promotion on the same date/time. The following are details on each combination of overlap based on the promotion type.

Overlapping Simple Promotions

RPM provides the ability to select how simple overlapping promotions are handled through the system option Simple Promotion Overlap Rule. This system option will determine the rules for handling two or more overlapping simple promotions.

If the value is set to compounding, the selling price is calculated by compounding each (two or more) simple promotion for the overlapping period.

RPM will download the start date and end date for each promotion detail for the overlapping period so that consuming systems will be updated with the compounded selling retail for the overlapping period. This rule is true for all simple promotions created either with the online screens, price event injector, or new item location processing.

The start date and end date of the promotions overlapping period should be separately sent to consuming systems so that when the overlapping period is complete, the selling retail reflects the active promotion post the overlapping period.

It is important to note that if RPM is integrating with ORPOS, best deal will always be taken, even if the system option is set to compounding. Other POS systems will receive data based on the system option and can determine how to handle based on capabilities.

Note: Apply Promo Change Type 1st-This system option will only be used when selling retail calculations are compounded. It will not be used when the system option for overlapping simple promotions is set to non-compounding best deal.

If the value is set to Non-compounding Best Deal, the selling price is calculated for each (two or more) promotion separately. For the overlapping period, the selling retail with the best deal for the customer should be available in the RPM tables. This same value should be sent to consuming systems. This rule is true for all simple promotions created either with the on line screens, price event injector, or new item location processing.

RPM will extract the appropriate start and end dates for the selling retail based on the overlapping period of the promotions such that the best deal is available as the selling retail for the overlapping period. When the overlapping period is complete, the selling retail based on the active promotion for that period should be available in RPM. The same should be sent to consuming systems.

Note: Having more than one fixed price simple promotion for an item location will be valid if overlap rule is set to non-compounding best deal.

The following is a business example:

Simple Promotion 1	Simple Promotion 2
Start Date = 1/15/2013	Start Date = 2/1/2013
End Date = 2/15/2013	End Date = 2/28/2013
Discount = 10 % Off	Discount = 15 % Off

Overlapping period: 2/1/2013 to 2/15/2013

- Non-compounding Best Deal:
  - 1/15/2013 to 1/31/2013 - Discount 10%
  - 2/1/2013 to 2/15/2013 - Discount 15%
  - 2/16/2013 to 2/28/2013 - Discount 15%
- Compounding

  - 1/15/2013 to 1/31/2013 - Discount 10%
  - 2/1/2013 to 2/15/2013 - Discount 10% + 15%
  - 2/16/2013 to 2/28/2013 - Discount 15%

- Complex promotion with a simple promotion:

  In this scenario, RPM will assume these promotions will be compounded. RPM does not calculate the promotional retail for complex promotions, so it will send the retail for the simple promotion and promotion details for the complex promotion without the retail to consuming systems. ORPOS will apply the complex promotion reward (discount rule) after the simple promotion is applied. This means if there is a complex promotion and a simple promotion applicable for an item, the system will apply both.

  In cases where a retailer's POS system is a third party system, complex promotions will be applied based on retailer's implementation rules and what their POS system supports.

- Overlapping a complex promotion with a complex promotion:

  Two or more overlapping complex promotions will be handled as non-compounding best deal by ORPOS. RPM does not calculate the promotional retail for complex promotions so the overlap rules do not apply for this condition. In cases where a retailer's POS system is a third party system, complex promotions will be applied based on retailer's implementation rules and what their POS system supports.

Promotion Maintenance Functionality

Based on the status of a promotion, some maintenance functions are available. If a promotion has not yet reached active status, users will have the option to set the promotion status back to worksheet, make any necessary changes to the promotion and request approval of the promotion.

If a promotion status is active; meaning the start date is in the past and the end date has not yet been reached, the following changes are allowed:

- Update promotion end date:

  - RPM provides the option to change the end date at header level; however, changing the header level will not automatically change the component detail level. It is recommended to make end date changes at the component detail level; this level also allows changes to time elements if the promotion is time based or users want to add time elements.
  - SIM can also send an update for an end date on an approved promotion; RPM will unapprove the promotion detail, add the end date change, and reapprove the promotion detail. If there is a conflict an error message will be sent from RPM to SIM through the web service

- Cancel promotion:

  - RPM allows users to cancel one or more component details from an active promotion. Cancellation can also be done at the header level, which will cancel all components attached to the promotion. If a user wants to cancel a subset of items and/or locations from a component detail, they will use the following option.

- Cancel items and/or locations:

- RPM provides users the ability to cancel items and/or locations off of an active promotion component detail for the following price event types: Simple Promotion or Threshold Promotion.
- Canceling items and/or locations can be done at varying levels based on the original promotion detail setup. The item/location window will only allow entry/selection of items based on the original promotion component detail. For example, promotion detail created at Department/Zone level; cancellation is allowed at the following levels:
  - **Department/Store**-Stores within a zone can be canceled for the entire department.
  - **Class/Zone or Store**-An entire class of items can be canceled either at zone level or by store.
  - **Subclass/Zone or Store**-An entire subclass of items can be canceled either at zone level or by store.
  - **Parent/Zone or Store**-Parent items can be canceled either at zone level or by store.
  - **Parent Diff/Zone or Store**-Parent diff items can be canceled either at zone level or by store.
  - **Transaction Item/Zone or Store**-Transaction items can be canceled either at zone level or by store.
- For example, promotion created at Parent Item/Zone level; cancel is allowed at the following levels:
  - **Parent/Store**-Stores within a zone can be canceled for the parent item.
  - **Parent Diff/Zone or Store**-Parent diff item(s) can be canceled either at zone level or by store.
  - **Transaction Item/Zone or Store**-Transaction item(s) can be canceled either at zone level or by store.
- For example, promotion created with Item List at location level; cancel is allowed at the following levels:
  - **Parent Item/Store**-Parent items can be canceled at store level.
  - **Parent Diff/Store**-Parent diff items can be canceled at store level.
  - **Transaction Item/Store**-Transaction items can be canceled at store level.

Note: This feature is only applicable to active promotion price events created as simple or threshold including vendor funded and customer segment. The rules for multi-buy promotions are not valid as changes could compromise the integrity of the scenario. For example, buy sandwich and bag of chips, get drink free. Canceling an item from this type of promotion would make the buy/get rule invalid.

- Add a new promotion component detail:
  - RPM allows users to add a new promotion component to an existing promotion that has active details under it.
  - In the scenario where there is an approved promotion that has system created exclusions and the user is adding a new component, the system will process

the new component data the same as if it was a new promotion. If during the approval of the new component conflicts are found, exclusions may be automatically created based on the defined tolerance level. The status of the added component or component detail will be approved if tolerance is not exceeded.

Note: Tolerance checking will not take into account the previous exclusions it will start counting from zero.

Promotion Constraints

Users are stopped if they are creating a price change within a set number of days of the start of an approved promotion or vice versa. Conflict checking will stop the user from approving the price change or promotion. The number of days is determined by a promotion constraint variable that is stored at the subclass/location level.

When the user runs conflict checking on a price change record, promotion record, or worksheet status record, promotion constraint checks are run. If a promotion constraint is violated, the user will see a conflict in either the price change or promotion dialog and the price event will not be approved. The user is able to optionally select to ignore promotion constraints on individual price change, promotion, or worksheet detail records so promotion constraint checks are not performed when conflict checking is run.

Pricing Strategies

The pricing strategies front end allows a retailer to define how item retails are proposed when pricing worksheets are generated. The strategies are defined at department, class, or subclass in order to represent which items are affected.

RPM offers six types of pricing strategies:

- **Area Differential Pricing** allows clients to set prices for items at a particular zone or zone group differently than another zone or zone group. The price differential is based on the rules a retailer defines. Area differentials are used when a retailer creates a price change to ensure consistent pricing. Differential pricing cannot be applied to other pricing events, such as clearances or promotions.
- **Clearance Pricing** allows clients to define the method used to clear merchandise by creating markdowns or a series of markdowns.
- **Competitive Pricing** allows a client to define a pricing strategy for their items based on their primary competitor's prices. All locations in a competitive pricing strategy must use the same currency.
- **Clearance Default** allows the user to specify clearance defaults as low as the subclass level and apply the subsequent markdowns in the manual clearance dialog. Pricing worksheets are not generated for this pricing strategy.
- **Margin Pricing** allows a client to define a pricing strategy for items based on margin targets.
- **Maintain Margin Pricing** allows a client to define the pricing strategy for items based on future cost changes. The proposed retails can be based on current or market basket margin percentages.

Future Retail

The future retail tables are the most important tables within RPM. All price event activities need to access and update the future retail tables. Most retailers do their pricing above the item/location level (parent level, zone level, or parent/zone level).

The data in the future retail tables are held at higher levels whenever possible, which can enhance application performance:

- The future retail tables will retain retail information at levels above transaction item and location so that this information can be leveraged by the retailer when creating price events at higher levels. Users can create price events at any merchandise hierarchy and location hierarchy level as necessary, the data on the tables will reflect the events at the level created.
- The Conflict Review List window displays the price event at the level of the timeline where the conflict occurred, that is, potentially higher than transaction item and location level.
- A roll-up batch ensures the system holds future retail data at the highest level possible. This batch looks at lower-level timelines and attempts to roll up to a higher level where timelines match exactly between lower levels and higher levels.
- A batch is used to handle a change to a merchandise hierarchy's primary zone group (PZG) definition. This batch explodes all existing timelines in the future retail tables to the transaction item and location level, and then regenerates higher level timelines based on the new PZG definition, followed by roll-up processing to remove unnecessary, non-exceptional data.

Future retail logic includes:

- Parent-zone timelines for an item with parent and location, which is part of the primary zone group for the associated merchandise hierarchy.
- Item-zone timelines where the item does not have a parent and the location is parent of the primary zone group for the associated merchandise hierarchy.
- Parent-location timelines where the item has a parent. These timelines are generated regardless of whether the location belongs to the primary zone group for the associated merchandise hierarchy.
- Parent/Diff-zone timelines where the item is parent of diff_1 of the parent item and the location is part of the primary zone group for the associated merchandise hierarchy.
- Parent/Diff-location timelines where the item is parent of diff_1 of the parent item. These timelines are generated regardless of whether the location belongs to the primary zone group for the associated merchandise hierarchy.

Price Inquiry

Price inquiry is designed to allow retailers to retrieve the price of an item at an exact point in time. This price might be the current price of a particular item or the future price. Users can search for prices based on the following search criteria:

- Merchandise hierarchy
  - Dept., Class, Subclass
- Item, Item List, Price Event Item List
  - Parent, Parent/Diff, Transaction Item
- Zone group

- Zone
- Location
- Location (warehouse or store)
- Date/Time

Price Event Injector

This feature provides the ability to generate price events via a bulk process that can be loaded into RPM. Price events generated by an external system other than RPM can be staged on tables and processed through a batch job where conflict check and validation rules are processed as data is passed to RPM. Price events can be created in RPM, in either worksheet or approved status.

Functionality supported by the price event injector, includes the following features.

Overall Use

Overall use is as follows:

- **Zone Group Level Use:** This feature provides users the ability to load data at the zone group level versus loading at zone level. This can be done for price changes, clearance and promotions. Allowing this level of data entry will provide a streamlined approach for customers. For example, if a promotion is being created for a zone group that has 10 zones set up and all locations will have the same promotional retail. Users will be able to load one row of data at the zone group level vs. loading 10 rows of data at zone level. Validation will still be done at zone level and data will be set up in RPM at zone level. Once the original staged data at zone group level has been exploded to zone level, any necessary corrections found during conflict check will need to be corrected at zone level versus zone group level via the price event injector.
- **Automatic Exclusion Creation Details Included in Reporting:** This feature will provide details on the price event injector tables that show processing information. Any processed data that have had automatic exclusions created during processing will be flagged as such, providing customers visibility at a high level. For additional information on the details, customers will need to manually view the data in the RPM UI.
- **Vendor Funded Price Events:** This feature provides the ability to create a vendor funded price event by selecting a deal previously created in RMS and attaching it to a price event (price change, clearance or promotion) in RPM. Existing deal details will need to be included in the stage data when creating a vendor funded price event.
- **Ignore Constraints:** This feature defines whether or not promotion constraints are taken into account when conflict checking is performed. The functionality is available for the creation of price changes or promotions. For more information see the promotion constraints section of this document.
- **Price Guides:** This feature provides the ability to use price guides to create a uniform pricing strategy. Price guides will maintain consistent price points, for example, rounded in the same manner or all end in the same digits. Once price guides are defined, they can be used for the following functions: price changes, clearance, and promotions.
- **Custom Attributes:** This feature provides the ability to assign custom attribute values during price event creation, adding attributes to previously created price

events is not supported (this is considered maintenance and can be done via the UI). The following price event types will support custom attributes:

- Price changes
- Clearances
- Promotions at multiple levels: header, component and component detail
  - Simple, threshold, multi-buy, transaction, and finance promotion components

Adding custom attribute values via the price event injector:

- Validation will not be done to ensure the custom attribute framework exists on the price event attribute tables.
- Custom attributes loaded that do not match up with the custom attribute framework on the tables will flow to the underlying custom attribute tables. However, they will not display in the RPM user interface.

Attribute values are freeform and will not be validated unless customer has set up specific validation rule.

Price Changes

Price change use is as follows:

- **Create Price Change:** This feature provides the ability to create a price change via the price event injector. The following change values are supported: change by amount, changes by percent, fixed price change or link code price change along with creating a vendor funded price change. System generated exclusions may be created during approval process if conflict check is run and finds errors.

Clearance

Clearance use is as follows:

- **Create Clearance:** This feature provides the ability to create a clearance via the price event injector. The following change values are supported: change by amount, changes by percent, or fixed price change along with creating a vendor funded clearance. System generated exclusions may be created during approval process if conflict check is run and errors are found
- **Independent Clearance Reset:** This feature provides the ability to add a clearance reset date, change a clearance reset date or remove a clearance reset date for designated items/locations independent of the original clearance. Valid item selection includes; parent items, parent/diff items, transaction level items, or the use of an item list.

  When submitting clearance resets via the UI the conflict check process understands that there will always be only one date for all of the selected items/locations. Because of the current conflict check handling, a rule is enforced that when submitting clearance resets through the price event injector, you must submit and process records for a single date one at a time. For example, if you are resetting a group of item/locations with a date one week from now and another group of item/locations with a date of two weeks from now, each group should be staged and processed separately. Staging both groups of item/locations at the same time will result in all records in fail/error status.

  If duplicate or overlapping data is encountered during processing of clearance resets, the system will skip the duplicate/overlapping records and process all other data. This could happen if the same item/location/date record appears twice on the staging table, or if a parent item is submitted along with one of the child items for the parent. This condition will not be flagged as an error; all data will be processed as submitted.

If records are submitted at parent level, RPM will validate that all child items are set up for a clearance, any child items that are not on an active clearance will be skipped.

- **Not supported:** Manual exceptions, or exclusions.

Promotions

Promotion use is as follows:

- **Create Promotion Header:** This feature provides the ability to create a new promotion including the promotion header details via the price event injector. This functionality is available when loading the stage tables with NEW promotion details only. This feature also allows the ability to have multiple component details records attached to a single promotional header. Users will have the ability to group component details, allowing for a new promotion to be created with more than one component detail associated to the promotion. Promotion headers can be created for all promotion component types; simple, threshold, multi-buy, transaction, and finance.
- **Customer Segment Promotions:** This feature allows entry of customer segment information that has been previously created in RMS when loading promotion data to the stage tables. This functionality is available when loading the stage tables with NEW promotion details only for all promotion component types; simple, threshold, multi-buy, transaction, and finance.
- **Create a New Deal in RPM:** This feature provides the ability to create a promotion component and attach a newly created deal (not already existing in RMS) to the promotion component. This functionality is available when loading the stage tables with NEW promotion component details for the following promotion types; simple, threshold, multi-buy, and transaction. The feature is not supported for finance promotion components.
- **Time Based Promotions:** This feature allows for a promotion to be created with specific start and/or end time elements defined for the promotion component detail. The functionality is available for all promotion component types; simple, threshold, multi-buy, transaction, and finance. For more details on this promotion feature, see "Time-Based Promotions".
- **Simple Promotion Components:** This feature provides the ability to create a simple promotion component via the price event injector. The following change values are supported: change by amount, changes by percent, fixed price change, exclude, or no change. System generated exclusions may be created during approval process if conflict check is run and finds errors.
- **Threshold Promotion Components:** This feature provides the ability to create a threshold promotion component via the price event injector. The features include the ability to create a single threshold or multiple thresholds along with defining the qualifier as either threshold level or item level. The functionality does not support the set up and creation of the threshold structure tied to a threshold promotion component. The threshold structure must be created via the RPM UI before using the data in the price event injector. The threshold structure will designate the change values for the promotion component. System generated exclusions may be created during approval process if conflict check is run and finds errors.

- **Multi-Buy Promotion Components:** This feature provides the ability to create a multi-buy promotion component via the price event injector. All features supported via the UI will be supported via the price event injector including; various types of multi-buy promotions, multiple buy/reward lists, price ranges and a promotion limit value. For more details, see the "Multi-buy Promotion Component" section of this document. System generated exclusions may be created during approval process if conflict check is run and finds errors.
- **Transaction Promotion Components:** This feature provides the ability to create a transaction promotion component via the price event injector. All features supported via the UI will be supported via the price event injector including storewide level promotions. For more details, see the "Transaction Promotion Component" section of this document. System generated exclusions may be created during approval process if conflict check is run and finds errors.
- **Finance Promotion Components:** This feature provides the ability to create a finance promotion component via the price event injector. When injecting a finance promotion component additional details including credit details and threshold details must be included on the stage tables. For more details, see the "Finance Promotion Component" section of this document.
- **Not Supported:** Manual exceptions or exclusions for any promotion component type.

See the *Oracle Retail Price Management Operations Guide* for more information on running the price event injector batch jobs.

Coupon Handling

RPM does not currently support the setup or handling of coupons. Coupon setup and maintenance is done in RMS and the data is passed to consuming systems. See the *Oracle Retail Merchandising System User Guide* for more details on coupons.

Worksheet

The RPM worksheet functionality is designed to allow for the maintenance of automatically generated price change and clearance proposals resulting from the RPM merchandise extract batch program. These proposed price changes/clearances are the product of existing strategies, calendars and item/location information.

Location Moves

Location moves in RPM allow you to select a location that exists in a zone and move to a different zone within a zone group on a scheduled date. The user will choose to approve the location move. A batch will process all approved location move records, run them for conflict checking and update them to scheduled status. The batch will run immediately before the Location Move Execution batch.

After conflict checking is complete, the process also allows the location to persist most valid pricing events through the move and to smoothly transition out of their old zone pricing strategies into the new zones' pricing strategies. System options provide the user the flexibility to configure location moves. Exclusions may be created during approval process if conflict checking runs and finds errors.

Layering within the RPM Application

RPM uses a Java architecture built on a layering model. Layers of the application communicate with one another through an established hierarchy and are only able to communicate with neighboring layers. The application is divided into a presentation layer, a middle tier consisting of services and business objects, and a database access/driver layer.

The segregation of layers has the following advantages, among others:

- The separation of presentation, business logic, and data makes the software cleaner, more maintainable, and easier to modify.
- The look and feel of the application is easily updated because the GUI is not tightly coupled to the back end.
- A layered architecture has become an industry standard.
- Portions of the data access layer (DAL) can be radically changed without affecting business logic or user interface code.
- The application takes advantage of Java database connectivity (JDBC), minimizing the number of interface points that must be maintained.
- Market-proven and industry-standard technology is utilized (for example, Java EE, Swing, JDBC, PL/SQL, and so on).

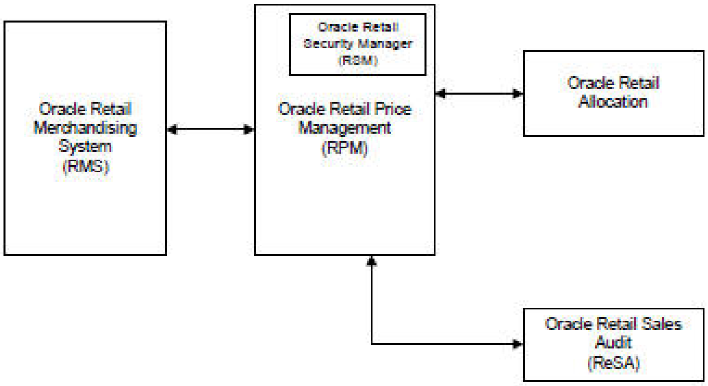
Integration with Other Applications

RPM exists on the same database schema as all the other Oracle Retail Merchandising Operations Management applications. As a result, there are a number of options for sharing information between applications.

Information from the RPM application is shared with and retrieved from other Oracle Retail Merchandising Operations Management (ORMOM) applications by reading directly from ORMOM application tables, by creating RPM views based on ORMOM application tables, by directly calling ORMOM application packages, and by allowing other ORMOM applications to call RPM packages, batch processes, and the RIB.

The following diagram illustrates the various applications with which RPM interacts in the Merchandising footprint:

Figure 10–4 RPM Interactions in Merchandising Footprint


Oracle Retail Merchandising System (RMS)
Oracle Retail Security Manager (RSM)
Oracle Retail Price Management (RPM)
Oracle Retail Allocation
Oracle Retail Sales Audit (ReSA)

RPM and RMS/RTM/ReSA

RPM and RMS provide a client with flexible options for how to implement both solutions. RPM exists on the same database schema as RMS which allows information to be shared between applications from direct database reads, package calls, and batch processes. RPM uses APIs to facilitate the exchange of information with RMS.

RPM provides the following to RMS:

- **Price Event Execution—**When regular, promotional, or clearance price changes are set to go into effect or end, the PriceEventExecutionBatch processes own the data flow. Once the pricing event has been processed by the batch program, it updates pricing in RMS by interfacing with the RMSSUB_PRICECHANGE API in RMS. Pricing is sent at item location level and supports all location types including wholesale/franchise locations.
- **Initial Pricing—**Initial pricing for items in RMS is dependent upon the primary zone group for the item defined in RPM and characteristics of that zone group. These characteristics include markup percent, markup percent type, and pricing guides. RPM provides this information to RMS through an API (MERCH_RETAIL_API_SQL).
- **Deal Creation—**RPM creates and associates Deals with regular, promotional, and clearance price changes. When this occurs RPM uses an RMS API (PM_DEALS_API_SQL) to create the deal in RMS.

RMS provides the following to RPM:

- **Foundation Data—**Foundation information is essential to RPM functionality. To successfully set up price events RPM needs to know the merchandise hierarchy, the organizational hierarchy, suppliers, and more. RPM is able to access this information through the RMS database.
- **Item—**Price events created in RPM ultimately relate to an item/location. RPM needs to know all approved items currently in the merchandising system, the active, discontinued, or inactive item/location relationships for those items, suppliers with which the items are associated, and more. RPM is able to access this information through the RMS database.
- **Competitive Pricing Information—**RPM has the ability to create price changes based off competitive activity in the marketplace. RPM is able to access this information through the RMS database.
- **Deals—**Deals can be associated to price events (including vendor funded promotions) in RPM. To associate price events to an existing deal RPM needs visibility to the deals currently available in the system. RPM is able to access this information through the RMS database.
- **Wholesale/Franchise Item Pricing—**RMS is responsible for determining wholesale/franchise item pricing which is maintained on the future cost table. RPM uses this information to generate recommended retails for wholesale/franchise business partners.
- **Store/Warehouse Creation—**New stores and virtual warehouses created in RMS must be added to a zone structure in RPM. To do this RMS provides RPM with the store and/or virtual warehouse being added, its pricing location, and its currency (to ensure it is the same as the zone it is being added to). RMS talks directly to RPM, RMS will call an RPM API that will trigger the update process.
- **Item/Location Creation—**When new item/location relationships are established RPM needs to verify that no future retail records currently exist, create an initial future retail record (for sellable items), and determine if there are any existing

price events that affect the item resulting in a future retail record for the price event as well. RMS talks directly to RPM, RMS will call an RPM API that will trigger the update process.

- **Item and Item/Location Deletion –** When an existing item or item location is deleted in RMS, RPM will remove all references to that item or item location including table clean up and removing them from existing price events. RMS talks directly to RPM, RMS will call an RPM API that will trigger the update process.

> **Note:** The following conditions do not automatically modify existing price events and should be considered manual maintenance.
> - Merchandise Hierarchy Deletion: Current base functionality in RPM does not process any changes when a merchandise hierarchy; department, class or subclass are deleted in RMS.
> - Location Deletion: Current base functionality in RPM does not process any changes when a location; warehouse or store, is deleted in RMS. RPM does not validate the status of a specific location during price event creation or maintenance. RPM will continue to calculate retail for items at locations that are closed until all item-location relationships for that location are removed in RMS and the necessary RPM batch processes are run to consume the resulting messages from RMS.

- **Item Modification** is used to notify RPM when an item has been reclassified. The details of the reclassification are written to an item modification table in RPM for the next batch processing run. RMS talks directly to RPM, RMS will call an RPM API that will trigger the update process.
- **Department Creatio**n is used to notify RPM when new departments are created in RMS. RPM creates aggregation level information for the new department using predefined system defaults. RMS talks directly to RPM, RMS will call an RPM API that will trigger the update process.

RPM and RTM

No information is exchanged.

RPM and ReSA

RPM provides ReSA with Promotion Information—RPM needs to provide ReSA with promotion information. ReSA gathers promotion data directly from the RPM promotion tables.

> **Note:** ReSA provides no information for RPM.

RPM and Oracle Retail Allocation

RPM provides the following to Allocation:

- **Future Retail Price Data—**Allocation uses future retail price data stored in RPM to provide the total retail value of an allocation. This interface is optional for a retailer and must be configured during implementation. Oracle Retail Allocation uses an API to access this information from RPM.

- **Promotions—**Users in Oracle Retail Allocation can select a promotion from RPM to associate an allocation with. Allocation OI functionality also uses exploded promotion data from RPM.

Note: Allocation provides no information to RPM.

RPM and ReIM

No information is exchanged.

RPM and ARI

ARI is a monitoring system that interacts with any applications database (including RPM). As such, it does not use any information from RPM; rather, it monitors the RPM database for events defined by a client and notifies the client when said events occur.

RPM Users and Security

RPM has its workflow and data security managed through an internal system, RSM. RSM is the part of the RPM application that provides a centralized method of authorizing system users.

RSM provides a two-tier security structure that consists of application and data level security. RSM provides centralized administration screens for system administrators to create application and data level permissions.

Additionally, RPM uses a directory service that complies with Light Directory Access Protocol (LDAP) to maintain and authenticate valid users. It is the system administrator's responsibility to ensure all users are set up in LDAP. If a client has an existing LDAP server where users are currently managed, RPM can be pointed to that server to eliminate the redundancy of maintaining multiple user names/passwords for the same user across a clients applications.

Application Level Security

Application level security allows applications to limit the business functionality users can access in the system and how they can interact with those functions. To determine the business functionality a user can access, RSM uses named permissions.

Named permissions are pieces of business functionality around which the application has security. For example, if RPM has Promotions functionality surrounded by security, RPM creates a promotions named permission. Named permissions data is sent to the RSM database during installation.

To determine what actions a user can perform for the named permissions, each named permission has defined actions.

Actions define how the user interacts with the functionality contained by the named permission. The types of actions attached to a named permission are as follows:

- **None**-Users associated to the role have access to the permission but no actions.
- **Edit**- Users associated with the role are allowed to see all secured information in a workflow.
- **View**-Users associated with the role are allowed to see all secured information in a workflow, but not make any changes to the data in the workflow.

- **Approve**-Users associated with the role are allowed to change the status of a workflow to Approved.
- **Submit**-Users associated with the role are allowed to change the status of a workflow from Worksheet to Submitted.
- **Emergency**-Users associated with the role are granted special access that goes beyond normal day-to-day access functionality. They can thus bypass normal delays in processing.

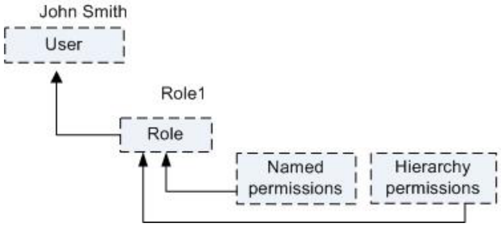
Using named permissions and actions, RSM provides a requesting application (RPM in this case) with the application access information it needs to determine what security is enforced for each functional area within the application.

Data Level Security

This allows applications to limit users' access to information based on hierarchy (merchandise and location) permissions. Applications either provide the details of these types up front with SQL scripts or dynamically by implementing an RSM interface and exposing it to the RSM service. RSM does not understand application-specific data (for example, RSM does not know the difference between departments and locations). To RSM, the data is a tag (for example, department) and a specific value (for example, 1000). This information is passed back to calling applications, and it is the applications responsibility to apply the data level permissions appropriately.

When application and data level security parameters are defined in RSM, they still need to be associated to users based upon their business roles. To facilitate this process RSM allows a system administrator to create security roles which application and data level security can be associated with. These roles can then be associated with users which causes the users to inherit the security permissions associated with their security role. When a user logs onto RPM, the user is authorized to use the business functionality and data associated with their role. The following diagram illustrates this concept.

Figure 10–5 Users are Associated with Functionality



System Options

All of the following system options are listed and can be viewed and edited through RPM user interface.

Allow Complex Promotions

System Option name:

RPM_SYSTEM_OPTIONS .COMPLEX_PROMO_ALLOWED_IND

When checked, this field is restricted/disabled in the user interface. This system option defines whether complex promotions can be created and maintained. If the indicator is checked or set to Yes, then all promotion types are available. If the indicator is set to unchecked or set to No, then only simple items and simple hierarchy type promotions can be defined, and threshold and multi-buy should not appear in the promotions dialog.

Business Example

This system option would be beneficial for those clients who choose to utilize only simple promotions. When indicator is unchecked, only promotion component type simple will appear to the user. If checked all promotion component types will exist.

Valid values: 0 - No (unchecked); 1 - Yes (checked)

Default value: 1 - Yes (checked)

Allow ItemList Explode

This system option gives the user the option to select whether the item list should be exploded to the item level or left at item list level.

Valid values are:

Y - Yes/Checked. Two options are provided in the Item Level field: Item List and Item (when Item Type = Item List).

N - No/Unchecked. Item Level could only be Item List (when Item Type = Item List)

The default value is N/unchecked.

Apply Promo Change Type 1st

System Option name: RPM_SYSTEM_OPTIONS .PROMO_APPLY_ORDER

When RPM is installed and system options are saved, this field is restricted/disabled. This system-level indicator is used to assist in the determination of appropriate application order for promotions and components. It is used when higher-level rules and indicators cannot determine which promotion or component should be applied before another and only used for the following scenarios:

- None of the components involved have a Fixed Price change type.
- Both of the components or promotions have the same ordering indicator setting (either primary or both secondary) and one component has a change type of % Off and one has a change type of Amount Off. The system level indicator will determine, across the company, whether promotion components with a Change Type of % Off or Amount Off should be applied first in relation to the other Change Type. This indicator will only be used for determining application order between components with these two change types, as components with other change types will be governed by other rules.

Business Example

If two promotions each have two components, and one of the promotions has two components with the preferred change type of amount off and the other promotion only has one component with the preferred change type of amount off, the promotion with two components with the preferred type of amount off is applied first.

The user will make a setting at the system level to specify whether percent off components will be applied before Amount Off components or vice versa. This is a setting that can be changed over time, and will be applied to situations at the time of component approval. However, it can only be updated when there are not any approved or active promotions in the system. If there are any approved or active promotions, this indicator is disabled. In the event a change is made to this system option setting, the change in ordering will not be applied to existing situations where multiple components are already approved. It will only be applied when new components are approved.

Valid values: % off; Amount off

Default value: Amount off

> **Note:** This system option will only be used when selling retail calculations are compounded. The rules will not be used when the system option for overlapping simple promotions is set to non-compounding best deal.

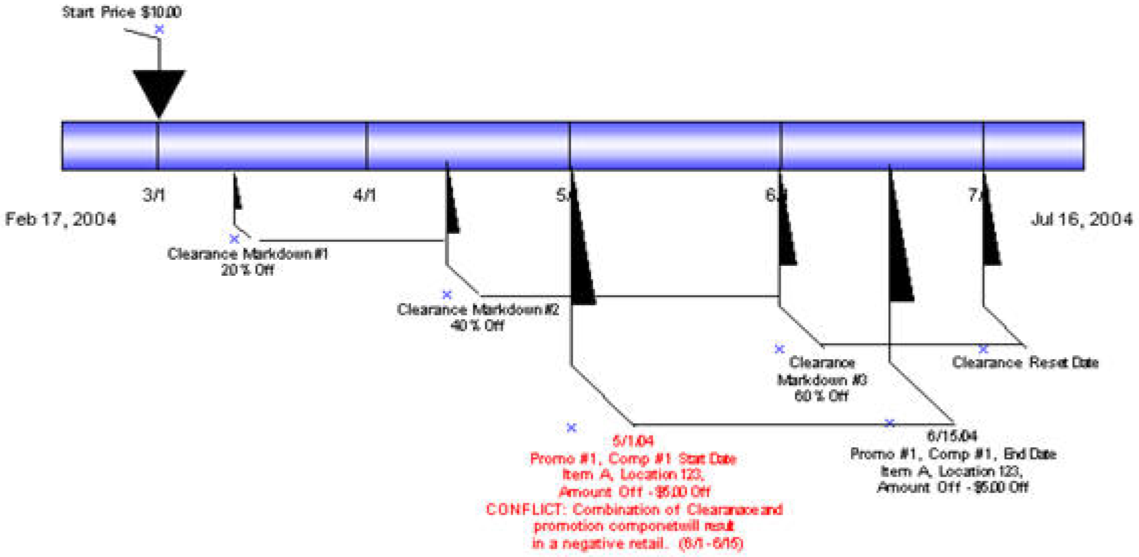
Clear/Promotion Overlaps

System Option name: RPM_SYSTEM_OPTIONS.CLEARANCE_PROMO_OVERLAP_IND

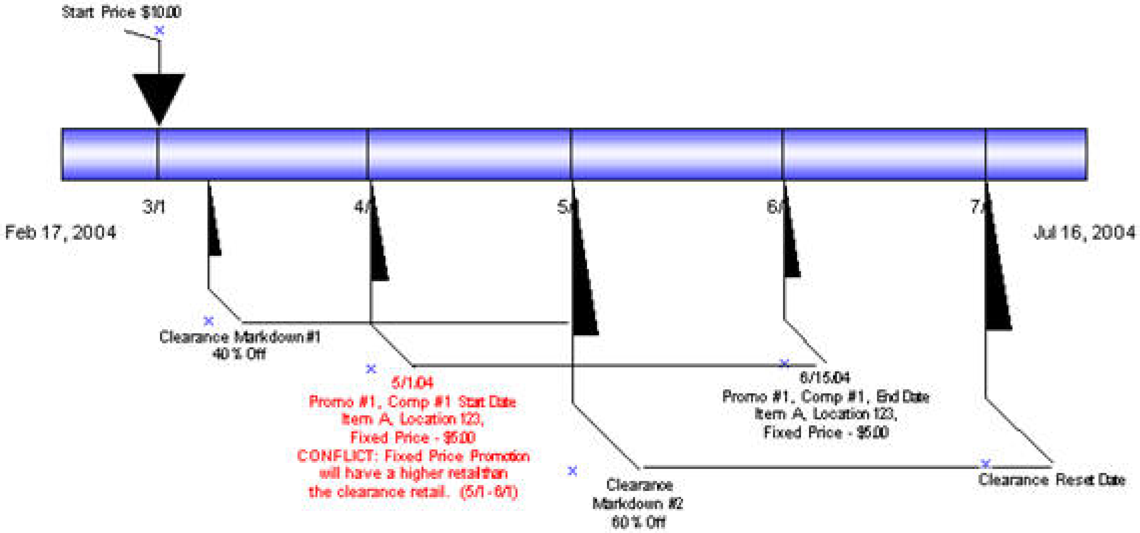
When checked, this field is restricted/disabled in the user interface. This system option defines whether or not an item/location can be on clearance and promotion at the same time. If the clearance/promotion overlap indicator is checked or Yes, the items that are on clearance will have the discount of the promotion applied to their clearance retail, not the regular price. If the promotion change type is set to Fixed Price, the item can only have the fixed price assigned that is lower than the clearance retail. If the clearance change takes place during the promotion then the promotional markdown will decrease when the clearance goes into effect. If the clearance/promotion overlap indicator is unchecked or No, clearance price changes cannot be submitted or approved where the effective date of the first markdown will fall on or after the start date or on or before the end date of an approved promotion, hence the clearance items cannot be promoted until the clearance resets. When this indicator is set to No, promotions cannot be submitted or approved where the item/locations currently exist on an active clearance or if the promotion start and end dates conflict with a pending approved clearance.

Business Examples

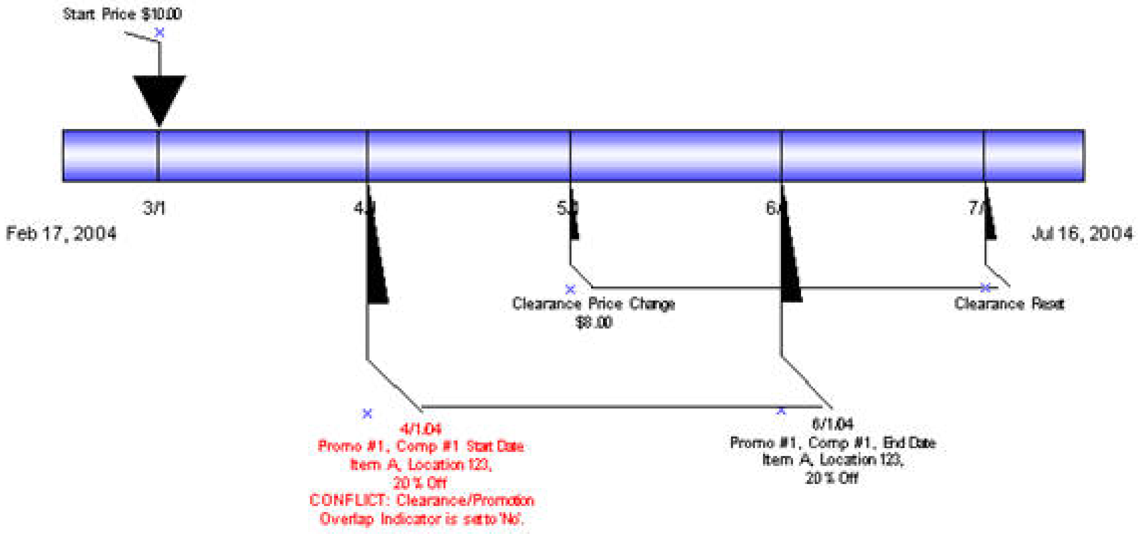
The Clearance/Promo Overlap Indicator is set to Yes. The user attempts to submit a promotion component that will result in the retail being negative, this will fail conflict checking processes. The approved Clearance exists for the same item/location as the promotion, therefore when both the clearance and the promotion component being submitted are applied; the retail will be less than $0.00 at some date and cause a conflict. This example illustrates that even with clearance/promotion overlap system capabilities, conflicts can be encountered.



Clearance/Promo Overlap Indicator is set to Yes. The user attempts to submit a Fixed Price promotion component with an Apply to Type of Clearance and Regular that will result in the promotional retail being higher than the Clearance retail at some date. Approved Clearance exists for the same item/location that has a lower Clearance retail than the Fixed Price Promotional retail of the component being submitted.



Clearance/Promo Overlap Indicator is set to No. User attempts to submit a promotion component with an Apply To Type of Clearance and Regular that will overlap dates with an existing approved Clearance for the same item/location fails Conflict Checking.



Valid values: 0 - No (unchecked); 1 - Yes (checked)

Default value: 1 - Yes (checked)

Clearance Reset Inquiry Max

System option name: RPM_SYSTEMS_OPTIONS.CLEARANCE_RESET_INQUIRY_MAX

This system option limits the data that is displayed in the clearance reset inquiry list.

Business Example

The Maximum Search Results-Clearances reset inquiry system option is set to 20. A user searches for clearance resets within five departments and returns 80 clearance resets. RPM returns records only up to that established value and displays the following message:

"The number of records that matched your search criteria is greater than the allowed maximum and therefore not all records have been returned. Refine your search criteria to narrow the search so that all valid records can be presented."

The user can then narrow the search criteria down further to one or two departments to show all clearances.

Valid values: 1-999

Default value: 100

Conflict History Days After

System option name: RPM_SYSTEM_OPTIONS.CONFLICT_HISTORY_DAYS_AFTER.

This system option defines the number of days after the vdate (current system date) that a pricing event is displayed to the user in the conflict review check screen.

Valid values: 1-99

Default value: 30

Conflict History Days Before

System option name: RPM_SYSTEM_OPTIONS.CONFLICT_HISTORY_DAYS_BEFORE.

This system option defines the number of days before the vdate (current system date) that a pricing event is displayed.

Valid values: 1-99

Default value: 30

Cost Calculation Method

System Option name: RPM_SYSTEM_OPTIONS .COST_CALCULATION_METHOD

The cost calculation indicator contains a drop down with the options: Average Location Cost and Highest Location Cost When zones are represented in a price event. This value is used to determine the cost for a zone. The cost is held at location level; therefore, either the highest cost across the locations in the zone or the average (non-zero) cost should be used.

Business Example

In RPM, the cost calculation method system option is set to Average Location Cost.

A price change is created at the zone level; there are five locations at a cost of 10.00 for four locations and 12.00 for the remaining fifth location. The cost in the price change apply block user interface is illustrated as an average cost or an average of the five locations and is totaled from the FUTURE_COST table in RPM.

Valid values: 0 - Average Location Cost; 1 - Highest Location Cost

Default value: 0 - Average Location Cost

Default Out of Stock Days

System Option name: RPM_SYSTEM_OPTIONS .DEFAULT_OUT_OF_STOCK_DAYS

This system option defines the number of days that should be added to an item clearance effective date in order to calculate the out of stock date. This default is only applied to generate the out of stock date when the clearance is first created. Existing out of stock values will be used for subsequent markdowns.

Business Example

The default out of stock system option is set to 30. A user creates a department zone level clearance. The VDATE is 1/1/09; the effective date is 1/3/09. The out of stock field will default with the date of 2/2/09**.**

Valid values: 1-999

Default value: 120

Default Reset Date

System Option name: RPM_SYSTEM_OPTIONS .DEFAULT_RESET_DATE

This system option defines whether a reset date should be defaulted when a clearance is created. When system option is set to No or unchecked, the reset date will appear NULL or blank in the create clearance dialog. However, the user can still manually enter a reset date. When the system option is set to Yes or checked, the reset date is defaulted to one day greater than the out of stock date, the user can edit or delete the defaulted date. Reset date will default one date per item/location.

Business Example

The default out of stock system option is set to 30 and default reset date system option is checked. A user creates a department zone level clearance. The VDATE is 1/1/09; the effective date is 1/3/09. The reset will default with the date to one day greater than the out of stock date, 2/1/09.

Valid value: 1

Default value: NULL

Display And/Or Condition

System Option name: RPM_SYSTEM_OPTIONS.DISPLAY_OR_CONDITION

In the multi-buy promotion type, the user will be able to use both And/Or selections in both the buy and get lists. Example: Buy a sandwich or a wrap and chips and get a soft drink for 50 cents off or a cookie for $1.00 off. The option to select the OR condition is controlled by the DISPLAY_OR_CONDITION system option. If system option is checked, the system will display the AND and OR conditions in the user interface. The system will default to AND. If system option is unchecked, it will display only the AND condition in the multi-buy promotions user interface.

Business Example

A multi-buy promotion is set up for the following scenario. In order to allow the OR condition the Display AND/OR Condition system option should be checked or Yes to add the OR condition to the multi-buy conjunction dropdown.

Buy a sandwich or a wrap and chips and get a soft drink for 50 cents off or a cookie for $1.00 off. The user would set up the above example in the following manner:

- Buy List 1(sandwich) OR Buy List 2(wrap) AND Buy List 3(chips) for -à
- Reward 1 (.50 cents off) OR Reward 2(cookie 1.00 off)

  Valid values: 0 - No (unchecked); 1 - Yes (checked)

  Default value: NULL

Display Conflicts Only

System Option name: RPM_SYSTEM_OPTIONS.DISPLAY_CONFILCTS_ONLY

When the system option is set for display conflicts only (the system option is checked), the record that caused the conflict is displayed in the lower portion of the Conflict Review List window, and no others.

When the system option for display conflicts only is unchecked and the days before/after are provided, RPM looks for all records, using the price event's effective date as the point to look forward and back. This returns all records within the window of time calculated.

Multiple records are displayed in the top portion of the Conflict Review List window so that, if one rule returns conflicts for multiple item/locations in a price event, all those conflicts are displayed. This is limited to 100 records maximum.

Days Event Displayed Before VDATE

System Option name:

RPM_SYSTEM_OPTIONS. CONFLICT_HISTORY_DAYS_BEFORE

This option defines the number of days before an effective date that a pricing event appears in the conflict review window. It enables only when the Display Conflicts Only indicator is not checked.

Valid values: 0 - No (unchecked); 1 - Yes (checked)

Default value: NULL

Days Event Displayed After VDATE

System Option: RPM_SYSTEM_OPTIONS.CONFLICT_HISTORY_DAYS_AFTER

This option defines the number of days after an effective date that a pricing event appears in the conflict review window. It enables only when the Display Conflicts Only indicator is not checked.

Valid values: 0 - No (unchecked); 1 - Yes (checked)

Default value: NULL

Dynamic Area Differentials

System Option name: RPM_SYSTEM_OPTIONS.DYNAMIC_AREA_DIFF_IND

Within the Area differential pricing strategy, this system option controls the ability to modify a secondary zone retail price after a primary zone has been approved. If the checkbox is checked (Y), the batch program Merchandise Extract will create all records, both primary and secondary areas, in New status. If a proposed retail is not available for the primary area, the secondary records should be calculated using the basis retail of the primary area or competitor retail (whichever is lowest).The worksheet will dynamically update the secondary locations based on changes to the primary location and the user will be allowed to edit the secondary areas regardless of the status of the primary records. If the checkbox is not checked (N), the batch program Merchandise Extract will create secondary locations in pending status and the user will not be allowed to edit the secondary locations in the worksheet until the primary area has been approved.

This system option is also responsible for controlling the ability to layer competitive strategies onto an Area Differential Strategy. If the option is checked (Y) then a competitor can be added to the area differential strategy for the secondary zones and the suggested retail price will be the lower price of the two strategies Area Differential and Competitive. The competitor information setup will not be applied in the Price Change dialog when calculating the proposed retail for a secondary area. The competitor information will only be used in the Merchandise Extract batch job and worksheet functionality.

Note: The proposed retails are displayed based on the percent higher or lower for the secondary locations.

Valid values: 0 - No (unchecked); 1 - Yes (checked)

Default value: 0 - No (unchecked)

Ends in Digits for Zero Decimal Currencies

System Option name: RPM_SYSTEM_OPTIONS. ZERO_CURR_ENDS_IN_DECS

When RPM is installed and system options are saved, this field is restricted/disabled. This value will be used in the Price Guides dialog when a currency is specified that does not contain digits following the decimal in the format. The number selected determines the number of digit fields that should be available in the Ends In definition area.

Valid value: 0-4

Default value: 0

Event ID Required

System Option name: RPM_SYSTEM_OPTIONS .EVENT_ID_REQUIRED

When unchecked, this field is restricted/disabled in the user interface, since the restriction cannot be added after promotions have been created. If this value is set to Yes (checked), then when a promotion is created an Event ID MUST be assigned to the promotion and validation will occur in the promotions definition dialog. If the value is set to No (unchecked), then promotions can be created without tying them to Events.

Valid values: 0 - No (unchecked); 1 - Yes (checked)

Default value: 0 - No (unchecked)

Exact Deal/Funded Promotion Dates

System Option name: RPM_SYSTEM_OPTIONS.EXACT_DEAL_DATES

When checked, this field name restricted/disabled in the user interface. This option defines whether or not the dates of a deal associated with a vendor funded promotion must match exactly. If the indicator is set to Y, then only deals with the same begin and end dates as the promotion component being created will appear in the deal LOV in the vendor funded promotions dialog. In the event the indicator becomes selected or checked, it becomes disabled and cannot be modified or unchecked since the restriction cannot be added after promotions and deals have been associated.

Business Example

If the Exact Deal/Funded Promotion Dates indicator is checked, the user creates a promotion with components start and end dates from 4/1/09 to 4/15/09. Any funding that is created for the component must fall within the promotion component start and end dates.

Valid values: 0 - No (unchecked); 1 - Yes (checked)

Default value: 0 - No (unchecked)

Filter Price Change/Clearance Results

System Option name: RPM_SYSTEM_OPTIONS.FILTER_PRICE_CHANGE_RESULTS
This system option will determine whether the Maintain Price Change/Clearance screen should go to the intermediate result screens or go directly to the Price Change/Clearance Editor screen.

Valid values: 0 - No (unchecked); 1 - Yes (checked)

Default value: Yes (checked)

Inherit New Zone's Regular Retail

System Option name: RPM_SYSTEM_OPTIONS. LOC_MOVE_PRICE_CHANGES

When a location move occurs, users can determine whether or not the moving location should inherit the new zone's retail based on the setting of this system option. When it is checked, the location will inherit the new zone's retails. If it is unchecked the zone will keep the existing retail as base. When the system option specifies that the location will automatically get the new zone's regular retail, the following will happen for the execution of the location move:

- A location level price change will be created with a unique reason code that will identify that the price change was created because of a location move being scheduled. The retail price on this price change will be equal to the items basis zone level retail of the new zone, if there is one. If the zone is not part of the primary zone group, there will be no zone level retail for the item and a price change will not be created. If the current locations retail already equals the zone level retail of the new zone, a price change will still need to be created in case location or zone level retail changes during the execution of the location move (see Execution notes).
- This price change will have an effective date equal to the location move execution date.
- The price change will be created and will go through conflict checking. If it passes conflict checking, it will be set to approved status. If it fails conflict checking, the price change will still be created but left in worksheet status and reported to the user that it could not be approved.
- When a price change is created in approved status, it will be reported to the store through the existing RIB message structure only if the new zones retail is different than the locations previous retail.
- The price change will be editable through the user interface after the move is implemented.
- Price change will have a system generated reason code.

  Valid values: 0 - No (unchecked); 1 - Yes (checked)

  Default value: No (unchecked)

Location Move Lead Time

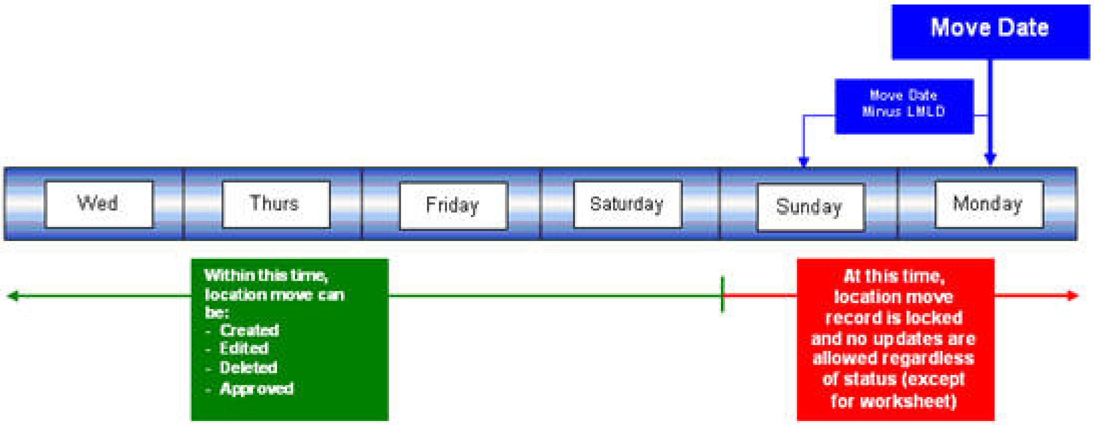
System Option name:

RPM_SYSTEM_OPTIONS .LOCATION_MOVE_PROCESSING_DAYS

This field is used to control the number of days from the current date the user is allowed to enter a location move effective date. The minimum value allowed is 1. The earliest effective date that can be selected as the effective date is VDATE + LMLD (Location Move Lead Days). For example, if Location Move Lead Days is set to 2 days, then the earliest location move effective date that can be set up is 2 days from today.

Business Example

If a location move in failed or approved status has a move date that has already passed the VDATE + LMLD (Location Move Lead Days), then the location move record is locked and no updates are allowed. When users try to update an expired location move record, error message appears, "Effective date of scheduled move cannot be less than today plus Location Move Lead Time." Updates to an expired location move record in any status is not allowed. The following is the illustration of the timeline:


Move Date
Move Date Minus LMLD
Wed
Thurs
Friday
Saturday
Sunday
Monday
Within this time, location move can be:
- Created
- Edited
- Deleted
- Approved
At this time, location move record is locked and no updates are allowed regardless of status (except for worksheet)

If a location move in worksheet status has a move date that has passed the lockout period (VDATE + LMLD), users get a warning message when trying to edit this record. The warning message displayed is "Date is out of range." Users are still able to make changes to this record but a valid new effective date needs to be re-entered before the batch can process it.

Valid values: 1-999

Default value: 1

Location Move Purge Days

System Option name: RPM_SYSTEM_OPTIONS .LOCATION_MOVE_PURGE_DAYS

This system option defines the number of days to wait when purging location move requests. The PurgeLocationMovesBatch program deletes location moves based on their effective date. Location moves are purged regardless of whether they have been run. Location moves are purged when their effective date is LOCATION_MOVE_PURGE_DAYS days in the past.

Business Example

If a Location Move Purge Days is set to 2 days, then the earliest location move effective date that can be deleted is 2 days in the past from today/VDATE.

Valid values: 1-999

Default value: 1

Maximum Number of Buy Lists

System Option name: RPM_SYSTEM_OPTIONS.MAX_BUY_LISTS

This system option gives the user the flexibility within the multi-buy promotion functionality to control the number of buy or reward lists within 1 promotion component.

Business Example

The maximum number of buy lists system option is set to 4. Therefore, a multi-buy/meal deal promotion can be set up for 1 component.

- Buy List 1 - Sandwiches
- Buy List 2 - Soft Drink

- Buy List 3 - Chips
- Buy List 4 - Cookies

The addition of Buy list 5- Candy would trigger this message to the user: "The maximum number of Buy lists has been exceeded for a single promotion component. A new list cannot be added."

Valid Value: 1-999

Default value: 4

Maximum Number of Promotion Component Details

System Option name: RPM_SYSTEM_OPTIONS.MAX_PROMO_COMP_DETAIL

This system option can restrict the promotion component detail user interface to a predetermined limit. The option is named Maximum Number of Promotion Component Details per Promotion Component. Limitations on the number of promotion component details affect any possible way in which a user can create promotion component details. Item lists, parent, parent diff, or transaction-level item/location combinations that are exploded to the item/location level are controlled through the value set for this system option.

The default setting is 500.

Maximum Number of Overlapping Promotion Component Details

System Option name: RPM_SYSTEM_OPTIONS.MAX_OVRLP_PROM_COMP_ DETAIL. The maximum number of promotions that can be created per item/location is controlled by a system option. If this existing system option is enabled, Maximum number of overlapping promotion component details is enabled-and the number of promotions will not pertain to exclusions, only inclusive promotions. The number of exclusions will not be restricted.

Note: All promotion component types with the exception of Transaction Promotion Components are included in the overlapping component count for an item/location.

Business Example

If Maximum number of overlapping promotion component details is set to six per item/location, the user will be able to create six promotion components for an item/location, If the user attempts to create a seventh promotion component for the item/location, a conflict will be created. Exclusions will not count toward this maximum number of allowed promotions.

Valid values: 1-999

Default value: 4

Maximum Number of Reward Lists

System Option name: RPM_SYSTEM_OPTIONS.MAX_REWARD_LISTS

This system option allows the user to control the number of buy or get lists in the promotion component.

Business Example

The maximum number of buy lists system option is set to 4. Therefore a multi-buy/meal deal promotion can be set up for 1 component.

- Reward List 1 - Shoes
- Reward List 2 - Shirts
- Reward List 3 - Hats
- Reward List 4 - Gloves

The addition of Reward list 5- Socks would trigger this message to the user: "The maximum number of Reward lists has been exceeded for a single promotion component. A new list cannot be added."

Valid values: 1-999

Default value: 4

Maximum Search Results - Clearances

System Option name: RPM_SYSTEM_OPTIONS.CLEARANCE_SEARCH_MAX

The following business example explains what happens if the number of rows for the search criteria exceeds the value entered in the new options.

Business Example

The Maximum Search Results - Clearances system option is set to 20. A user searches for approved clearances within five departments and returns 80 clearances.RPM returns records only up to that established value and displays the following message:

"The number of records that matched your search criteria is greater than the allowed maximum and therefore not all records have been returned. Refine your search criteria to narrow the search so that all valid records can be presented."

The user can then narrow the search criteria down further to one or two departments to show all clearances.

Valid values: 1-999

Default value: 300

Maximum Search Results - Price Changes

System Option name: RPM_SYSTEM_OPTIONS. PRICE_CHANGE_SEARCH_MAX

This option limits the data that is displayed in the price change list. When the user applies item/loc detail to a price change or clearance the system populates the detail table for that price event. In the case where there is a large number of item/locations that exceed the system option setting, on Apply, only a portion of the rows in the table will be populated. RPM returns records only up to that established value and displays a message when exceeded.

Business Example

The Maximum Search Results - Price Changes system option is set to 20. A user searches for approved price changes within 5 departments and returns 80 price changes. RPM returns records only up to that established value and displays the following message:

"The number of records that matched your search criteria is greater than the allowed maximum and therefore not all records have been returned. Refine your search criteria to narrow the search so that all valid records can be presented."

The user can then narrow the search criteria down further to one or two departments to show all price changes.

Valid values: 1-999

Default value: 300

Maximum Search Results - Price Inquiry

System Option name: RPM_SYSTEM_OPTIONS.PRICEINQUIRY_SEARCH_MAX

In the Price Inquiry screen, the user can search for a price of an item using the following criteria:

- Merchandise hierarchy
  - Dept., Class, Subclass
- Item, Item List, Price Event Item List
  - Parent, Parent/Diff, Transaction Item
- Zone group
- Zone
- Location
- Location (warehouse or store)
- Date/Time

Based on the criteria entered the by the user, a listing of item/locations together with the specified prices at a given date are displayed. The resulting information will be available to the user for export. The system option will limit the data that is displayed in the price inquiry search results list. When the Price Inquiry is populated, only a portion of the results are displayed determined by the value set in the Maximum Search Results -

Price Inquiry system option. If the user wants to review the details for the remaining columns, which tend to be the performance intense fields, the user can do so by selecting rows and requesting the additional data.

Valid values: 1-999

Default value: 300

Maximum Worksheet Search Result

System Option name: RPM_SYSTEM_OPTIONS.WORKSHEET_FILTER_SEARCH_MAX

When the user enables the Minimize Worksheet Data Filter screen system option, this system option will also be enabled and will default a value of 300 which is the recommended setting. Maximum Worksheet Search result is the maximum number of rows that will be displayed based on the respective search criteria.

Based on the criteria entered by the user, the user will be limited to how many item/zone rows are returned when the user performs a search for the worksheet filter functionality.

If the user attempts to bring back too many rows, they will be stopped and given an error message.

Valid values: 1-999

Default value: null

Minimize Worksheet Data Filter Screen

System Option name: RPM_SYSTEM_OPTIONS.WORKSHEET_FILTER_SCREEN

When this system option is enabled, the user is given an optional workflow that includes worksheet data filtering. The user can enter the following search criteria to filter out the items to review.

- Clearance Indicator
- Primary Competitor Retail Change Indicator
- Primary Competitor Alert
- Comp A Alert
- Cost Changes During Review Period
- New Item Location Indicator
- Item ID
- Parent ID
- Link Code
- New Retail
- Price Change Indicator
- Reviewed

The user also can choose the skip filter option from the worksheet status screen to bypass the filtering. If this system option is disabled, the worksheet filter screen is not accessible.

Valid values: 0 - No (unchecked); 1 - Yes (checked)

Default value: 0 - No (unchecked)

Multiple Item/Loc Promotions

System Option name: RPM_SYSTEM_OPTIONS .MULTI_ITEM_LOC_PROMO_IND

When checked, this field is restricted/disabled in the user interface. This system option defines whether an item/location can exist on more than one promotion and more than one component within the promotion. If the indicator is set to Yes (checked), an item can have its retail price affected by more than one promotion at a single time in a given location. If the indicator is set to No (unchecked), only one promotion or promotion component can exist at the same time for a given item/location.

Valid values: 0 - No (unchecked); 1 - Yes (checked)

Default value: No (unchecked)

Number (#) of Days Rejects Held (PC Clear)

System Option name: RPM_SYSTEM_OPTIONS REJECT_HOLD_DAYS_PC_CLEAR

This field defines the number of days after the effective date of a rejected price change or clearance when it should be purged from the system.

Valid values: 1- 999

Default values: 30

Number (#) of Days Rejects Held (Promo)

System Option name: RPM_SYSTEM_OPTIONS REJECT_HOLD_DAYS_PROMO

This field defines the number of days after the end date (or start date for promotions with no end date) of a rejected promotion when it should be purged from the system.

Valid values: 1-999

Default value: 30

Open Zone Use

System Option name: RPM_SYSTEM_OPTIONS .OPEN_ZONE_USE

When checked, this field is restricted/disabled in the UI. It defines whether different Zone Group types can be used in all the pricing dialogs or if the type of the Zone Group will limit where it can be used. For example, if set to No (unchecked), the zone group can only be used for its designated purpose; promotion zone groups can only be used to create promotions; price change zone groups can only be used to create regular price changes or used in strategies and clearance zone groups can only be used to create clearances or used in strategies.   If this option is set to yes, it cannot be changed. Open Zone enables total flexibility in the use of Zone Groups. When the indicator is set to Yes, all zone group types can be used throughout the system's functionality. In addition, clearance zone groups cannot be used to define the pricing strategies of Margin, Competitive, Relationship, and Area Differential.

Business Example

Creating a margin strategy with a zone group defined as promotion or clearance would require the open zone use system option to be checked or set to Yes.

Valid values: 0 - No (unchecked); 1 - Yes (checked)

Default value: No (unchecked)

Past Markup Events Display Count

System Option name: RPM_SYSTEM_OPTIONS.PAST_MARKUP_EVENTS_DISPLAY_CNT

This system option defines the number of past markup impact events displayed for margin visibility.

Valid values: 1-10

Default value: 1

Price Change Processing Days

System Option name:

RPM_SYSTEM_OPTIONS .PRICE_CHANGE_PROCESSING_DAYS

This value represents the number of days required between the create date and the effective date. It does not determine the communication processes or timing but simply allows the retailer to ensure that price changes are created with enough advance timing that stores and other process areas can react accordingly. This value will be used in the price change and pricing worksheet dialogs. Security will permit certain users to violate this timeframe and create same day price changes or emergency price events. Therefore, zero is not permitted at the system level.

Business Example

A user creates a price change effective tomorrow. The Effective Date is the date when the price change will go into effect at the point of sale. This date must be greater than VDATE + the PRICE_CHANGE_PROCESSING_DAYS system option.

Valid values: 1-999

Default value: 1 (day)

Price Change/Promotion Overlaps

System Option name:

RPM_SYSTEM_OPTIONS .PRICE_CHANGE_PROMO_OVERLAP_IND

When checked, this field is restricted/disabled in the user interface. The field value indicates whether an item/location can have a price change occur during the middle of an active location. If the indicator is set to Yes (checked), then a price change can take effect during a promotion. If the indicator is set to no (unchecked), then a price change cannot be approved that would take effect during an approved promotion--and a promotion cannot be approved if its dates will overlap an approved price change. If this option is checked, it is recommended that promotion constraints be used to take advantage of the suggested new date functionality. If this functionality is not used, overlaps will be highlighted in conflict checking and changes would have to be entered again using the correct effective dates.

Valid values: 0 - No (unchecked); 1 - Yes (checked)

Default value: No (unchecked)

Pricing Strategy Default Effective Day

System Option name: RPM_SYSTEM_OPTIONS.DEFAULT_EFFECTIVE_DATE

This system options allows the user to select the default day of the week for prices to be effective in the worksheet detail dialog for the clearance pricing strategies. When a specified weekday is selected during price strategy create, it will default the effective date in the worksheet detail to the specified day after the review period (plus price change processing days)-ultimately causing the effective date in worksheets to default to the desired day of the week.

Business Example

Consider the following clearance strategy example.

A client creates a clearance strategy with calendar starting on a Tuesday with a 2-day review period and 26 days between. In this example, it is assumed the batch is run Monday night. The batch will find calendars that have a review period that starts the next day.

Then the client creates a regular calendar starting on a Thursday with a two-day review period and five days between. The intention is to have all price events effective on the following Saturday after the review period. In this example it is assumed the batch is run Wednesday night. The batch will find calendars that have a review period that starts the next day.

Users review the clearance worksheet on Tuesday and Wednesday, and the margin worksheet on Thursday and Friday.

The default effective date functionality will allow the user to default the effective date to the Saturday after the review for both the clearance and margin strategies. When calculating the effective date for the worksheet detail dialog, this new functionality works as base does today in that effective date considers price change processing time when calculating a proper default effective date. In this example price change processing days is set to 1, and thus effective date would fall on the next day (Saturday).

Business Example Rules

The following business rules apply.

- The user can select only one day of the week as a default per strategy.
- Pricing Strategy Default Effective Day functionality applies to margin, clearance and competitive strategies only.
- The resulting effective date appearing in the worksheet detail defaults to day of the week after the current review period plus price change processing days.
- When the system option for the Pricing Strategy Default Effective Day is checked, it is not modifiable.
- If two pricing strategies for the same item/location default to the same effective date, conflicts will occur.

Valid values: 0 - No (unchecked); 1 - Yes (checked)

Default value: NULL

Pricing Strategy Review Period Overlap

System Option name: RPM_SYSTEM_OPTIONS. LOC_MOVE_PS_REVIEW_ OVERLAP This field indicates if a location move is allowed to overlap the review period of a pricing strategy. When this system option is checked, it allows pricing strategies with review periods to overlap a location move date. The validation of any worksheet review period will be bypassed and a location move can be scheduled else if unchecked validation will not be bypassed.

- If the system option indicates that a move cannot be scheduled if there will be a worksheet review period that overlaps the move date, and a pricing strategy exists with a calendar that will have a review period that overlaps the move date, the move will fail.
- If the system option indicates that a move can be scheduled regardless of any worksheet review period, the validation for worksheet review period overlaps will be bypassed.

Valid values: 0 - No (unchecked); 1 - Yes (checked)

Default value: No (unchecked)

Promotion End Date Required

System Option name: RPM_SYSTEM_OPTIONS.PROMO_END_DATE_REQUIRED

When checked, this field is restricted/disabled. If integration with ORPOS is required, this system option requires that an end date be assigned to promotions. It must be checked when integrating with ORPOS.

Valid values: 0 - No (unchecked); 1 - Yes (checked)

Default values: 0 - No (unchecked)

Promotion History Months

System Option name: RPM_SYSTEM_OPTIONS .PROMOTION_HIST_MONTHS

This system option defines the number of months after a promotion is completed that is should be purged. This parameter is also used to purge cancelled promotions.

Valid values: 1- 999

Default value: 6

Promotion Overlap

System Option name: RPM_SYSTEM_OPTIONS. LOC_MOVE_PROM_OVERLAP

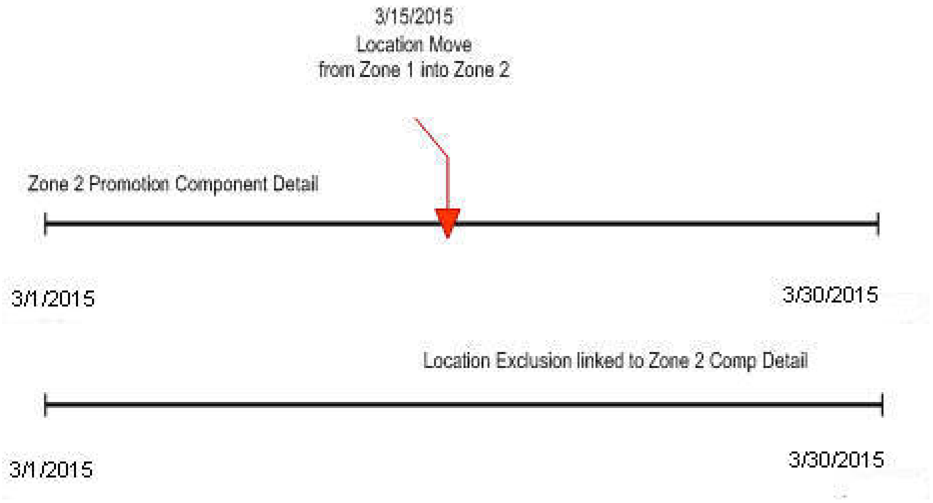
This field indicates if a promotion will overlap a location move. If checked, the LOC_MOVE_PROM_OVERLAP_BEHAVIOR system option will enable and allow the user to control how the promotion should overlap the location move during scheduling and execution.

RPM_SYSTEM_OPTIONS.LOC_MOVE_PROM_OVERLAP_BEHAVIOR

This option is within a dropdown box. If selected, it will have the following combinations enabled as options for the user to choose in a combination dropdown list. Only one of the following three options can be selected:

Extend Old Zone's Promotion and Do Not Inherit New Zone’s Overlapping Promotions

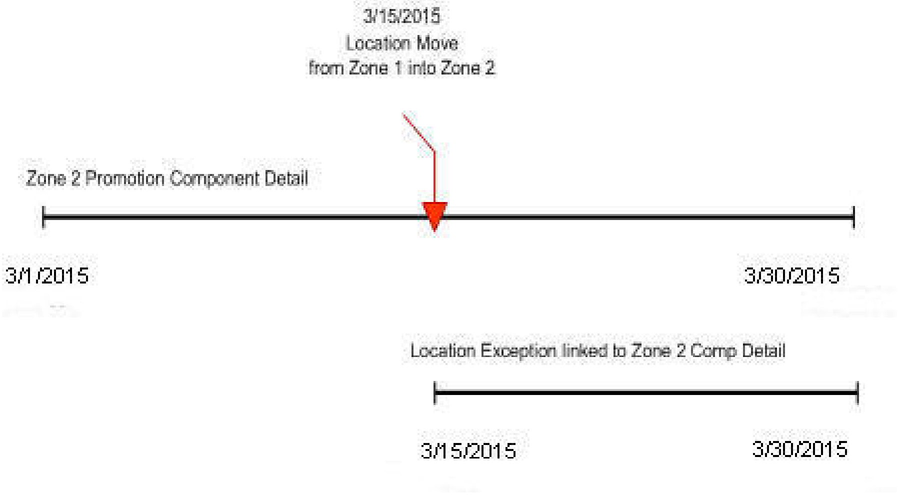
Location will keep running the old zones promotion. Location will not inherit any zone level promotion for the new zone if it overlaps the move date. Zone level promotions for the new zone that start before or on the location move date and end after or on the location move date and the promotion is currently pending, will have an exclusion created for the location that is being moved with a start date equal to the zone promotions start date and an end date equal to the zone promotions end date.


3/15/2015
Location Move
from Zone 1 into Zone 2
Zone 2 Promotion Component Detail
3/1/2015
3/30/2015
Location Exclusion linked to Zone 2 Comp Detail
3/1/2015
3/30/2015

Zone level promotions for the new zone that start before the location move date and end after or on the location move date and the promotion is currently active, will have an exclusion created for the location that is being moved with a start date equal to the current date + 1 and an end date equal to the zone promotions end date.

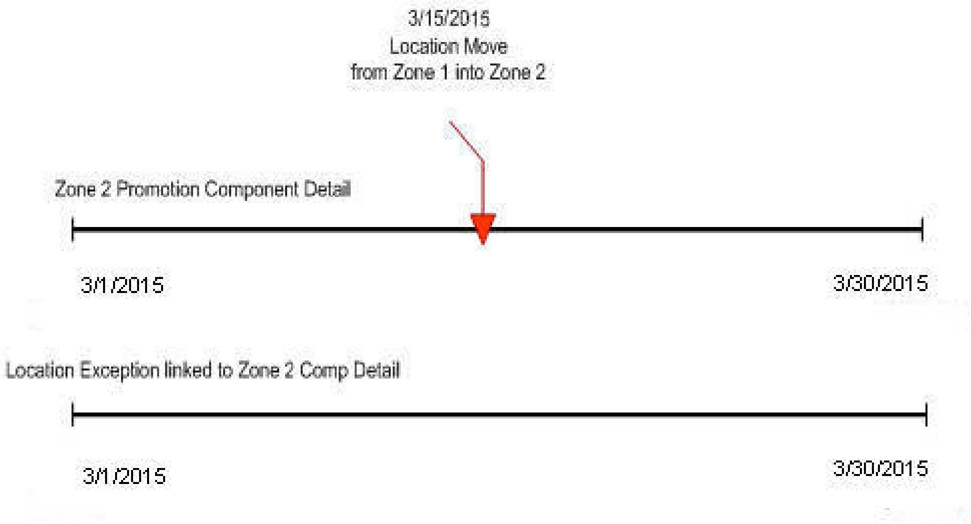
End Old Zone's Promotion and Inherit New Zone’s Overlapping Promotions

Promotion will end at the location the evening before the location move date. Location inherits the new zone’s promotion that overlaps the move date, but the promotion will only start on the location move date. Zone level promotions for the new zone that start before or on the location move date and end after or on the location move date will have an exception created for the location that is being moved with a start date equal to the move date and an end date equal to the zone promotion’s component’s end date.


3/15/2015
Location Move
from Zone 1 into Zone 2
Zone 2 Promotion Component Detail
3/1/2015
3/30/2015
Location Exception linked to Zone 2 Comp Detail
3/15/2015
3/30/2015

Do Not Start Old Zone's Overlapping Promotions and Inherit New Zone’s Overlapping Promotions

Location will not start the promotion if the zone promotion overlaps the move date. Location will inherit the new zone's promotion that overlaps the move date and will start the same day the zone level promotion starts or will start the day the move is scheduled if the zone level promotion is already active. Zone level promotions for the new zone that start before or on the location move date and end on or after the location move date will have an exception created for the location that is being moved with a start date equal to the zone promotion's start date or current date if the promotion is already active and an end date equal to the zone promotion's component's end date.


3/15/2015
Location Move
from Zone 1 into Zone 2
Zone 2 Promotion Component Detail
3/1/2015
3/30/2015
Location Exception linked to Zone 2 Comp Detail
3/1/2015
3/30/2015

Valid values: 0 - No (unchecked); 1 - Yes (checked)

Default value: No (unchecked)

Promotion Search Maximum

System Option name:

RPM_SYSTEM_OPTIONS.PROMOTION_SEARCH_MAX

This system option limits the data that is displayed when searching at promotion header level.

Business Example

The Maximum Search Results-Promotion search max system option is set to 20. A user searches for promotions within five departments and returns 80 promotions. RPM returns records only up to that established value and displays the following message:

"The number of records that matched your search criteria is greater than the allowed maximum and therefore not all records have been returned. Refine your search criteria to narrow the search so that all valid records can be presented."

The user can then narrow the search criteria down further to one or two departments to show all clearances.

Valid values: 1-999

Default value: 300

Purge Reset Clearance Months

System Option name: RPM_SYSTEM_OPTIONS.CLEARANCE_HIST_MONTHS

Expired clearance records must be purged from the system in order to manage the size of the database. The field holds the number of months past a clearance reset date that should be purged. If a clearance is never reset it cannot be purged, as the item/location still exists as a clearance. In addition, any RPM_FUTURE_RETAIL records associated with a clearance will not be purged until the clearance is reset.

Business Example

A clearance created prior to the VDATE is completed and now reset. This clearance would be purged with the execution of the clearance purge batch jobs.

A clearance is a candidate for purging if it has:

- Completed all markdowns and has been reset.
- Been cancelled.
- Been deleted.

Two clearance purge programs are run together:

- PurgeExpiredExecutedOrApprovedClearancesBatch purges expired executed and/or approved clearances.
- PurgeUnusedAndAbandonedClearancesBatch purges unused and/or abandoned clearances.

Valid values: 1-999

Default value: 1 (month)

Recognize WH as Locations

System Option name: RPM_SYSTEM_OPTIONS RECOGNIZE_WH_AS_LOCATIONS

When checked, this field is restricted/disabled. This system option controls whether warehouses will exist in RPM zones and assigned to regular and clearance price zones. They cannot be assigned to promotion price zones. When the system options is set to Yes, warehouses will appear in the Location LOVs and can be assigned to price zones, pricing strategies, and calendars. In addition, warehouse inventory will be taken into consideration when creating vendor funded markdowns and allows for the revaluation of inventory at a specific warehouse location. When the system option is set to No, warehouses are not available in RPM and will not appear in the Location LOVs. Stock on hand values will be zero and will not be calculated for warehouse locations when set to No..

Valid values: 0 - No (unchecked); 1 - Yes (checked)

Default value: Yes (checked)

Sales Calculation Method

System Option name: RPM_SYSTEM_OPTIONS SALES_CALCULATION_METHOD

The value specified is used by the merchandise extract to determine how to populate the Projected Sales column on the worksheet used together with Pricing Strategies. The Smoothed Average Calculation, which populates a table in RMS and is used by RPM to show the projected sales of an item for a time period representing a week. The intent of the smoothed average sales calculation is to represent on a weekly sales basis the number sales units sold at regular price and therefore not affected by a promotion or some sales anomaly.

Business Example

If a pricing manager in RPM wants to see the sales impact of changing the retail price of table salt by .20 per container, then the unit sales should simply be those that occurred when the item was sold at its regular price and will assume that the change will not slow demand but simply increase sales dollars by .20 per unit. The smoothed average value serves to provide this information in lieu of a more accurate forecast.

Clients using RPM who do not have access to an accurate sales forecast or are not able to accurately represent the regular (non-promotional, non-seasonal) sales of an item but need to use the value in RPM to calculate the sales values in the application, can use the value provided by a program that runs in RMS and calculates a representative value for unit sales of an item per week. If the RPM_IND on SYSTEM_OPTIONS is set to Y, meaning the client is using RPM with RMS, the IF_RPM_SMOOTHED_AVG table will be populated with a sales value in units and a counter value for each item/location combination. The program looks at sales data from Sales Audit and pulls the unit sales values for days when only regular price sales were recorded. If promotional sales were found during the day we assume the unit sales for that day do not represent the typical regular price sales needed in the calculation. The goal is to represent only those sales that occurred at regular price and only on days when no other factors where involved. The sales unit's values for days when only regular price sales were recorded are pulled into the calculation.

Components of the Calculation

- **Average Tolerance Percent** is a value used to determine if a sales amount received falls within a reasonable range to be considered in the calculation. Values that fall outside the range would be considered outliers and are capped at the high or low of the range.
- **Max Counter Value** provides a way to weight the existing sales value on the table against new values received. If the item has a relatively minimal seasonal swing this value can be set higher so the value will remain stable and takes many anomalies to move the value. If the item has a relatively strong seasonal swing the counter should be set to a lower number so that the value is more easily moved to reflect a trend or seasonal shift.

These values are defined in RMS on the Department Maintenance (dept) screen in the Max._Average Counter and Average Tolerance % fields. The values set by the user are store on the DEPS table in RMS and held in the following columns: AVG_TOLERANCE_PCT, MAX_AVG_COUNTER

Valid value: 1

Default value: 1

SIM Indicator

System Option name: RPM_SYSTEM_OPTIONS. SIM_IND

This field Indicates whether RPM is interfacing with SIM.

Note: The UI system option name is External Prices Allowed.

Valid values: 0 - No (unchecked); 1 - Yes (checked)

Default value: 0

Simple Promo Overlap Rule

System Option name: RPM_SYSTEM_OPTIONS. SIMPLE_PROMO_OVERLAP_RULE

This system option will determine the rules for handling two or more overlapping simple promotions. When the value is set in the system it will become restricted/disabled. DBA assistance will be required if the system option needs to be changed. Changing the value will impact new promotions or promotions that are unapproved maintained and reapproved. Changes to the value will not impact approved promotions.

See the Overlapping Promotions chapter for more information on Simple Promotion Overlap Rules.

Skip Conflict Checking During Submit

This option allows the user to skip the conflict checking process during the price event Submit action. If this system option is checked, RPM does not perform the conflict checking process (and does not show conflicts, if there are any) when the user performs a price event Submit action.

The price event status is switched to Submit immediately. If this system option is unchecked, RPM runs the conflict checking process (and shows conflicts found) when the user performs the price event Submit action.

Skip Conflict Checking During Complex Promotion Approval

This option allows the user to skip the conflict checking process during complex promotion (multi-buy and threshold) approval and disapproval. When the user approves or unapproves a multi-buy or threshold promotion, RPM does not update the future retail table and does not perform conflict validation. RPM populates the payload tables, however, so that RPM can communicate the complex promotion to other systems (through RIB or the flat file extract, for example).

If this system option is selected, RPM does not perform conflict checking, and does not show any conflicts, when the user approves or unapproves a complex promotion. If this system option is not selected, RPM runs a complete conflict checking process and shows conflicts found when the user approves or unapproves a complex promotion.

Choosing this option can dramatically improve the performance of an approval of a large multi-buy or threshold promotion (for example, at department/zone level). You should consider the use of this option carefully, because some functionality is lost with this option. For example:

- For overlapping promotions, an item/location in a complex promotion is not considered when validating overlapping promotion counts.
- For price change promotion overlaps or clearance promotion overlaps, an item/location in a complex promotion is not considered when validating these types of overlapping.
- Complex promotion information is not available in the future retail tables.
- Deals cannot be attached to Complex promotions.
- Apply To selection during promotion create will not be taken into consideration when applying the promotion details, system will assume Apply To Both Regular and Clearance.
- For fixed price promotions, the rule that only one can exist for an item/location/date will not be enforced. For example you could have one fixed price simple promotion and one fixed price complex promotion for the same item/location on the same date.

Although this system option can be changed from one status to the other in the System Options Edit screen, changing this system option could produce unpredictable results when there are active or pending complex promotions.

System Generated Exclusions Tolerance %

System Option name: RPM_SYSTEM_OPTIONS. SYS_GEN_EXCLUSION_ TOLERANCE

This system option will determine the percentage of transaction items on a price event that are allowed to error during conflict checking and still move forward with the approval process of the price event. The option can be found on the general tab under System Options; System-Generated Exclusions Tolerance %.

The tolerance setting will result in the percentage of transaction item exclusions created that the users will need to review and/or fix. The tolerance percent value will be set up as a system option that will be maintained by the retailer and can be updated as necessary. The default tolerance value will be set to 0%. The maximum tolerance value will be 25%.

If the tolerance percent value is exceeded during conflict checking the price event status will be set to worksheet and error details will be displayed for review by the users. The conflict check process verifies errors by item location; however, the tolerance will be measured based on transaction item only.

Business Examples

- Tolerance quantity set to 10%, price event has 200 items, if conflict checking finds an error with a five transaction items at 100 locations the tolerance rule counts as five items against the 200-this scenario passes the tolerance rules resulting in an approved price event.
- User creates promotion using an item list with 300 items; the system tolerance is set at 5%. If more than 15 transaction items return errors during conflict check the entire event remains in worksheet status and errors are displayed for review. If less than 15 transaction items return errors the approval process will move forward with exclusions automatically created for any items not passing conflict check.

Unique Unit of Measure

System Option name: IS_UOM_UNIQUE and UOM_VALUE

This option provides the ability to inform RPM that there is only one unit of measure (UOM) to be used throughout RPM. This is particularly useful to improve performance when a fixed-price price event is created. Previously, when the user created a fixed-price price event without entering a new UOM, RPM validated that all item/locations affected by the price event had a unique UOM. For a very large price event (for example, a promotion at department/zone level), this validation process could cause performance degradation.

For this functionality, there are two system options:

- A unique UOM must be used for all items (true/false)
- The actual unique UOM that will be used for all items (for example, EACH, LBS)

The Unique UOM is used for all items system option informs RPM that there is only one UOM in RPM. When this system option is selected, RPM does not perform UOM validation when the user creates a fixed-price price event. If this system option is selected, the user must provide a valid UOM to be used throughout RPM as the value of the Unique UOM to be used for all items system option. These two system options are set during the RPM installation. For an existing RPM environment, these system options should be updated from the back end by the database administrator.

Update Item Attributes

System Option name: RPM_SYSTEM_OPTIONS UPDATE_ITEM_ATTRIBUTES

This value is used by the merchandise extract to determine if the following attributes should be updated during each extract that occurs during a worksheet review period. If the Update Item Attributes indicator on System Options is set to Yes, then for every existing worksheet that is in New or Update status and is not ending today, the merchandise extract will update the following information on the worksheet. When this option is set to yes, worksheets will become dynamic and update daily for the entire review period. If this option is set to no, worksheets are static.

- **Primary Competitor Retail** stores the retail price at the primary competitor's store. This value is stored in the COMP_PRICE_HIST table in RMS.
- **Primary Competitor Store** is the Competitor Store associated with the Primary Competitor. This value is stored on the COMP_STORE table in RMS.
- **Primary Competitor Retail Selling UOM**-Primary competitor's retail selling UOM for the item/primary competitor store.
- **Primary Competitor Retail Standard UOM**-Primary competitor's retail UOM or the item/primary competitor store.
- **Primary Competitor Multi-Units**-Multi-unit value at the primary competitor's store. This value is stored on the COMP_PRICE_HIST table in RMS.
- **Primary Competitor Multi-Unit Retail**-Multi-unit retail at the primary competitor's store. This value is stored on the COMP_PRICE_HIST table in RMS**.**
- **Primary Competitor Multi-Unit Selling UOM**-Primary competitor's retail multi-unit UOM for the item/primary competitor store**.**
- **Primary Competitor Boolean**
- **Competitor A Retail**-Competitor Store associated to Competitor A. This value is stored on the COMP_STORE table in RMS.
- **Competitor B Retail**-Competitor Store associated to Competitor B. This value is stored on the COMP_STORE table in RMS.
- **Competitor C Retail**-Competitor Store associated to Competitor C. This value is stored on the COMP_STORE table in RMS.
- **Competitor D Retail**-Competitor Store associated to Competitor D. This value is stored on the COMP_STORE table in RMS.
- **Competitor E Retail**-Competitor Store associated to Competitor E. This value is stored on the COMP_STORE table in RMS**.**
- **Competitor A Store**-Retail price at Competitors A's store. This value is stored on the COMP_PRICE_HIST table is RMS**.**
- **Competitor B Store**-Retail price at Competitors B's store. This value is stored on the COMP_PRICE_HIST table is RMS**.**
- **Competitor C Store**-Retail price at Competitors C's store. This value is stored on the COMP_PRICE_HIST table is RMS.
- **Competitor D Store**-Retail price at Competitors D's store. This value is stored on the COMP_PRICE_HIST table is RMS.
- **Competitor E Store**-Retail price at Competitors E's store. This value is stored on the COMP_PRICE_HIST table is RMS.

- **Pending Cost Change Indicator**-Indicates whether there is a pending cost change at any of the locations within the zone on today's date or after. The information on the FUTURE_COST table in RMS will be used to determine the value of this indicator.
- **Cost Change Alert**-Alerts the user if there is a cost change in the x number of days that is set in the aggregation level details.
- **Current Cost**-Current cost of the item/location (value stored on the FUTURE_ COST table in RMS). If the location is a zone the cost calculation method from the System Configurations table will be used.
- **Basis Base Cost**-Base cost of the item/supplier/country at the given location. This value is stored on the FUTURE_COST table in RMS**.**
- **Reference Warehouse Stock on Hand**-Summary of the current stock on hand for the item from the reference warehouses on the strategy (or in the zone). This value is stored on the ITEM_LOC_SOH table in RMS.
- **Reference Warehouse on Order**-Summary of the current on order for the item from the reference warehouses on the strategy (or in the zone). This value is stored on the ORDLOC table in RMS.
- **Location on Order**-Summary of the current on order for the item from the reference stores on the strategy (or in the zone). This value is stored on the ORDLOC table in RMS.
- **Location Inventory**-Summary of the current stock on hand for the item from the stores on the strategy (or in the zone). This value is stored on the ITEM_LOC_SOH table in RMS.

  Valid values: 0 - No (unchecked); 1 - Yes (checked)

  Default value: Yes (checked)

Zone/Parent Ranging

System Option name: RPM_SYSTEM_OPTIONS .ZONE_RANGING

If this system option is checked, when a user creates a price change/clearance/promotion at a level higher than transaction/location (tran/zone, parent/zone, parent/loc, and so on.) RPM will ensure there is at least one item/location ranged underneath the event. If there is not at least one, then the system will not allow the user to create the event.

If this system option is not checked, then ranging will only be performed on price changes/clearances/promotions created at the transaction/location level and ranging checks will not be performed on events created at any other higher level.

This system option is used so that when setting up promotions at zone and parent level, it checks that locations and parent items are ranged to zones and locations respectively to ensure that promotions are not set up on items where they are not sold.

Valid Value - 0 - No (unchecked), 1 - Yes (checked)

Default value: 1 - Yes (checked)

System Defaults

Unless otherwise noted, the following system defaults can be viewed and edited through RPM user interface.

Currency

System Default name: RPM_SYSTEM_OPTIONS_DEF.DEF_CURRENCY

This system default defines the currency code that should appear in each of the dialogs in the system where the user is required to enter a currency. This is used in the pricing guides and price zone dialogs. The values are selected from RMS CURRENCIES table.

Valid values: Currency value extracted from the CURRENCIES table located in RMS
Default value: USD

Diff Type

System Default name: RPM_SYSTEM_OPTIONS_DEF.DEF_PRICE_CHANGE_DIFF_TYPE

This field holds a value for the desired default item level in the pricing event dialogs.

Valid values and default value are dependent on the Diff Values set up in RMS.

Ignore Constraints

System Default name: RPM_SYSTEM_OPTIONS_DEF.DEF_WKSHT_PROMO_CONST_IND

This system default determines whether constraint checking is defaulted as checked in the worksheet dialog.

Valid values: 0 - No (unchecked); 1 - Yes (checked)

Default value: Yes (checked)

Item Level

System Default name:

RPM_SYSTEM_OPTIONS_DEF .DEF_PRICE_CHANGE_ITEM_LEVEL

This is the default indicator to define the item level that will be displayed when the user enters the price change and clearance dialogs.

Valid values: 0 - Parent; 1- Parent/Diff; 2 - Single Item

Default value: 2 - Single Item.

Maintain Margin Strategy Method

RPM_SYSTEM_OPTIONS_DEF DEF_MAINT_MARGIN_METHOD

This field stores the default method type for the Maintain Margin price strategy.

Valid values: M - Market Basket Margin; C - Current Margin

Default value: NULL

Pricing Strategy

System Default name: RPM_SYSTEM_OPTIONS_DEF DEF_PRICING_STRATEGY

This field holds a value for the default pricing strategy type that displays when the user creates new pricing strategies.

Valid values: 0 - Area Differential; 1 - Clearance; 2 - Competitive; 3 -Margin

Default value: 2 - Clearance

Type

System Default name: RPM_SYSTEM_OPTIONS_DEF .DEF_PRICE_CHANGE_TYPE

This field holds a value for the default price change type that will be displayed when the user is creating either a price change or a promotion.

Valid values: 0 - Change by Percent; 1 - Change by Amount; 2 - Fixed Price

Default value: 1 - Change by Amount

Implementation Considerations

Carefully consider system options setup and system rules.

Conversion/Configuration Business Processes

- The rationale for determining the Zone Group, Zone and Location groupings might be based on any factors of the retailer's choosing.
- No promotions or pricing events should exist for a Zone prior to deletion.
- If a location is not included in a member of the merchandise hierarchies' Primary Zone Groups, zone price is driven from the merchandise hierarchy member's Primary Zone Group's base zone.
- A Primary Zone Group can be defined for members of each level of the merchandise hierarchy.
- A new store is assigned to a zone in RPM through designation of its Pricing Store at the time of the location's creation.
- An item should never be unique compared to the rest of its merchandise hierarchy level in terms of price zones. A category of items should follow the same zone group, ensuring manageability and accountability within the process.
- It is an assumption that all locations are assigned to a zone group for regular pricing.
- Security control should be considered around zonal maintenance to ensure that restrictions are put in place where necessary.
- Security control around Emergency Price Change functionality should be limited.
- The more granular level chosen to manage retail prices (for example, location traits, competitive pricing, demographics) the more complicated it can be. Managing numerous Zones can affect overall profit objectives.

Internationalization

For details on the language supported information see, Oracle Retail Merchandising System documentation for the current release.

11

Oracle Retail Active Retail Intelligence

This chapter is an overview of Oracle Retail Active Retail Intelligence (ARI).

Exception Reporting

ARI takes business rules that users are applying to reports in their daily tasks and applies them directly to the retail systems. This allows the correct people within the retail organization to be alerted when the business rule conditions have been met and informs them of what the appropriate actions might be to resolve it. This enables a retailer to apply the best practice to their entire organization by tying actions to information in an intelligent way.

A business example of an exception reporting rule is a rule that looks for orders that are past their ship date but have not been shipped yet. Rather than having an analyst scan a report to identify which orders have been shipped late a rule could be created in ARI to identify these orders for the analyst and the recommended steps to take to resolve the late shipping issue with the vendor.

Workflow Management

ARI maps out a retailer's business processes and notifies the users responsible for those processes whenever new tasks require attention. ARI also lists the appropriate actions that might be taken to complete these tasks and allows users to take action directly from the screen where they receive the alert.

If the alert is not acted on within an acceptable timeframe, ARI can define an automated action that needs to take place. This can be to raise the priority of the alert or send it to a supervisor or take an automated system action to ensure that single users do not become a bottleneck.

A business example of a workflow management rule is a rule that notifies an inventory manager when orders are ready for approval. This gives the manager instant visibility to all orders that need to be approved for the day rather than requiring the manager to search for orders meeting this requirement. Additional logic can also be added to notify the manager's supervisor when orders awaiting approval are not approved after a pre-determined timeframe.

Enterprise Process Modification

ARI allows a retailer to modify process within Oracle Retail to more closely match their business processes. This is achieved in most cases without modifying the underlying retail system.

A business example of an Enterprise Process Modification is a client changing its transfer approval process from analysts being able to approve transfers to only managers being able to approve transfers. Rules could be developed in ARI to send alerts to both the analyst and manager when transfers are submitted to educate them on the new business process to help ensure a smooth transition.

Integration with Other Applications

ARI is a monitoring system that interacts with any applications database. As such it does not use any information from other retailing applications; rather, it monitors the retailing application databases for events defined by a client and notifies the client when said events occur.

The only complication that can result is when a client is making database changes that affect the data mapping for rules already defined in ARI. In this case the rule needs to be changed to reflect the new data mapping.

ARI Security

Active Retail Intelligence provides no special security features or safeguards. Addressing any site-specific security issues involving Active Retail Intelligence is the customer's responsibility. Security settings in other applications with which Active Retail Intelligence interacts are not overridden or circumvented by Active Retail Intelligence. Whereas this is generally desirable, it is a consideration when determining to whom to route Active Retail Intelligence alerts. Sending an alert to a user who does not have the privileges to take the actions necessary to resolve the event might prove frustrating and counter-productive. Users should be educated about this issue so that they avoid forward events that have actions with limited access as well.

At a data level, Active Retail Intelligence detection is necessarily done with full access privileges to all data. Individual users with data level security might see different values for some parameters (in particular those involving sums) than the values seen by Active Retail Intelligence. This might cause adverse effects such as a user looking at an event automatically causing it to close because the user's limited data access causes the event to see values that make Active Retail Intelligence think the exception is no longer an issue when in fact it still is. For this reason, Oracle urges extreme caution when designing Active Retail Intelligence processes that involve users with limited data access. The consequences of missing alerts are great in an exception driven enterprise, so extra care is needed in the technical analysis of how such Active Retail Intelligence processes behave.