

Oracle® Retail Price Management
Operations Guide Addendum
Release 11.0.9

February 2007

Copyright © 2007, Oracle. All rights reserved.

Primary Author: Nathan Young

The Programs (which include both the software and documentation) contain proprietary information; they are provided under a license agreement containing restrictions on use and disclosure and are also protected by copyright, patent, and other intellectual and industrial property laws. Reverse engineering, disassembly, or decompilation of the Programs, except to the extent required to obtain interoperability with other independently created software or as specified by law, is prohibited.

The information contained in this document is subject to change without notice. If you find any problems in the documentation, please report them to us in writing. This document is not warranted to be error-free. Except as may be expressly permitted in your license agreement for these Programs, no part of these Programs may be reproduced or transmitted in any form or by any means, electronic or mechanical, for any purpose.

If the Programs are delivered to the United States Government or anyone licensing or using the Programs on behalf of the United States Government, the following notice is applicable:

U.S. GOVERNMENT RIGHTS Programs, software, databases, and related documentation and technical data delivered to U.S. Government customers are "commercial computer software" or "commercial technical data" pursuant to the applicable Federal Acquisition Regulation and agency-specific supplemental regulations. As such, use, duplication, disclosure, modification, and adaptation of the Programs, including documentation and technical data, shall be subject to the licensing restrictions set forth in the applicable Oracle license agreement, and, to the extent applicable, the additional rights set forth in FAR 52.227-19, Commercial Computer Software – Restricted Rights (June 1987). Oracle Corporation, 500 Oracle Parkway, Redwood City, CA 94065

The Programs are not intended for use in any nuclear, aviation, mass transit, medical, or other inherently dangerous applications. It shall be the licensee's responsibility to take all appropriate fail-safe, backup, redundancy and other measures to ensure the safe use of such applications if the Programs are used for such purposes, and we disclaim liability for any damages caused by such use of the Programs.

Oracle, JD Edwards, PeopleSoft, and Siebel are registered trademarks of Oracle Corporation and/or its affiliates. Other names may be trademarks of their respective owners.

The Programs may provide links to Web sites and access to content, products, and services from third parties. Oracle is not responsible for the availability of, or any content provided on, third-party Web sites. You bear all risks associated with the use of such content. If you choose to purchase any products or services from a third party, the relationship is directly between you and the third party. Oracle is not responsible for: (a) the quality of third-party products or services; or (b) fulfilling any of the terms of the agreement with the third party, including delivery of products or services and warranty obligations related to purchased products or services. Oracle is not responsible for any loss or damage of any sort that you may incur from dealing with any third party.

Contents

Preface	v
Audience	v
Related Documents	v
Customer Support	v
Conventions	vi
1 Introduction	1
2 Technical Architecture	3
Conflict Checking	3
3 Batch Processes	5
Java Batch Processes	5
Java Batch Process Architectural Overview	5
Running a Java-Based Batch Process	5
Script Catalog	6
Scheduler and the Command Line	6
Functional Descriptions and Dependencies	7
Batch Process Scheduling	8
Threading and the RPM_BATCH_CONTROL Table	8
Return Value Batch Standards	8
Return Values	8
Batch Logging	8
ItemLocDeleteBatch Batch	9
MerchExtractKickOffBatch Batch Design	10
NewItemLocationBatch batch design	13
ZoneFutureRetailPurgeBatch Batch Design	15

Preface

Oracle Retail Operations Guides are designed so that you can view and understand the application's 'behind-the-scenes' processing, including such information as the following:

- Key system administration configuration settings
- Technical architecture
- Functional integration dataflow across the enterprise

Audience

Anyone with an interest in developing a deeper understanding of the underlying processes and architecture supporting Oracle Retail Price Management functionality will find valuable information in this guide. There are three audiences in general for whom this guide is written:

- Business analysts looking for information about processes and interfaces to validate the support for business scenarios within and other systems across the enterprise.
- System analysts and system operations personnel:
 - Who are looking for information about Oracle Retail Price Management's processes internally or in relation to the systems across the enterprise.
 - Who operate Oracle Retail Price Management regularly.
- Integrators and implementation staff with overall responsibility for implementing Oracle Retail Price Management.

Related Documents

For more information, see the following documents in the Oracle Retail Price Management Release 11.0.9 documentation set:

- Oracle Retail Price Management Release Notes
- Oracle Retail Price Management User Guide
- Oracle Retail Price Management Online Help
- Oracle Retail Price Management Installation Guide
- Oracle Retail Price Management Data Model

Customer Support

- <https://metalink.oracle.com>

When contacting Customer Support, please provide:

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

Conventions

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

Note: This is a note. It is used to call out information that is important, but not necessarily part of the procedure.

This is a code sample
It is used to display examples of code

[A hyperlink appears like this.](#)

Introduction

The information in this document reflects modifications and updates to the *Oracle Retail Price Management 11.0.4 Operations Guide*. (The RPM 11.0.4 Operations Guide is the most recent release of the full Operations Guide for the 11.0 release of RPM.) Using this document in conjunction with that guide provides retailers with a complete overview of the application.

For the Oracle Retail Price Management 11.0.9 release, there is one updated batch design and three new batch designs. The **MerchExtractKickOffBatch** batch design has been updated. Changes to the batch design appear in bold. The **ItemLocDeleteBatch**, **NewItemLocationBatch**, and **ZoneFutureRetailPurgeBatch** batch designs are new. Additional changes to the “Batch Processes” chapter appear in bold.

For more specific information regarding enhancements and modifications made to the previous Oracle Retail Price Management release, see the Oracle Retail Price Management 11.0.9 Release Notes.

Technical Architecture

Conflict Checking

The major components of Conflict Checking in RPM will be performed by PL/SQL and will be executed on the database server instead of the application server. This addresses performance concerns around this functionality. A number of PL/SQL package files need to be installed on the database.

Batch Processes

Retailers should refer to these sections in lieu of the sections in the RPM 11.0.4 Operations Guide, “Chapter 7 – Java and RETL Batch Processes”.

Java Batch Processes

This section provides the following:

- An overview of RPM’s batch processing
- A description of how to run batch processes, along with key parameters
- A functional summary of each batch process, along with its dependencies
- A description of some of the features of the batch processes (batch return values, and so on)

Java Batch Process Architectural Overview

The goal of much of RPM’s Java batch processing is to select business objects from the persisted mechanism (for example, a database) by a certain criteria and then to transform them by their state. These RPM Java-based batch processes remove some of the processing load from the real-time online system and are run periodically.

Note the following characteristics of RPM’s batch processes:

- RPM’s batch processes are run in Java. For the most part, batch processes engage in their own primary processing.
- They are not accessible through a graphical user interface (GUI).
- They are scheduled by the retailer.
- They are designed to process large volumes of data, depending upon the circumstances and process.
- They are not file-based batch processes.

Running a Java-Based Batch Process

Java processes are scheduled through executable shell scripts (.sh files). Retek provides each of these shell scripts. During the installation process, the batch shell scripts and the .jar files on which they depend are copied to a client-specified directory structure. See the Installation Guide for details. The batch shell scripts must be run from within that directory structure.

Each script performs the following internally:

- sets up the Java runtime environment before the Java process is run.
- triggers the Java batch process.

To use the scripts, confirm that the scripts are executable (using `ls -l`) and run “`chmod +x *.sh`” if necessary. The shell scripts take two arguments: username and password. The output can be redirected to a log file (as shown in the example below).

Note: The script `launchRpmBatch.sh` must be modified to include the correct environment information before any of the batch scripts run correctly.

The following is an example of how to use a batch shell script:

```
./locationMoveBatch.sh MyUsername MyPassword > log 2>&1
```

Additional Notes

- All the output (including errors) is sent to the log file.
- The scripts are meant to run in Bash. They have problems with other shells.
- If the scripts are edited on a Windows computer and then transferred to Unix, they may have carriage returns (^M) added to the line ends. These carriage returns (^M) cause problems and should be removed.

Script Catalog

Script	Batch program executed
itemLocDeleteBatch.sh	ItemLocDeleteBatch
locationMoveBatch.sh	LocationMoveBatch
merchExtractKickOffBatch.sh	MerchExtractKickOffBatch
newItemLocBatch.sh	NewItemLocBatch
priceChangeAutoApproveResultsPurgeBatch.sh	PriceChangeAutoApproveResultsPurgeBatch
priceChangePurgeBatch.sh	PriceChangePurgeBatch
priceChangePurgeWorkspaceBatch.sh	PriceChangePurgeWorkspaceBatch
priceEventExecutionBatch.sh	PriceEventExecutionBatch
priceStrategyCalendarBatch.sh	PriceStrategyCalendarBatch
promotionPurgeBatchbatch.sh	PromotionPurgeBatchbatch
purgeExpiredExecutedOrApprovedClearancesBatch.sh	PurgeExpiredExecutedOrApprovedClearancesBatch
purgeLocationMovesBatch.sh	PurgeLocationMovesBatch
purgeUnusedAndAbandonedClearancesBatch.sh	PurgeUnusedAndAbandonedClearancesBatch
worksheetAutoApproveBatch.sh	WorksheetAutoApproveBatch
launchRpmBatch.sh	The retailer does not schedule this script. Other batch programs call this script behind the scenes. Note that this script sets up environment information and takes as a parameter the name of the batch program to run.
zoneFutureRetailPurgeBatch.sh	ZoneFutureRetailPurgeBatch

Scheduler and the Command Line

If the retailer uses a scheduler, arguments are placed into the scheduler.

If the retailer does *not* use a scheduler, arguments must be passed in at the Unix command line.

The Java batch processes are to be called via the shell scripts. These scripts take any and all arguments that their corresponding batch process would take when executing.

Functional Descriptions and Dependencies

The following table summarizes RPM's batch processes and includes a description of each batch process's business functionality.

Batch processes	Details
ItemLocDeleteBatch	This batch program handles RMS deletions of item locations.
LocationMoveBatch	This batch process moves locations between zones in a zone group.
MerchExtractKickOffBatch	This batch process builds worksheets in RPM. MerchExtractKickOffBatch.java either creates or updates worksheets based on price strategies and the calendars attached to them.
NewItemLocBatch	This batch program ranges item locations by putting them into the future retail table.
PriceChangeAutoApproveResultsPurgeBatch	This batch process deletes old error message from the price change auto approve batch program.
PriceChangePurgeBatch	This batch process deletes past price changes.
PriceChangePurgeWorkspaceBatch	This batch process deletes abandoned price change workspace records.
PriceEventExecutionBatch	This batch process performs the necessary work to start (regular price change, clearance price change, promotions) and end (price change, promotions) pricing events.
PriceStrategyCalendarBatch	This batch process maintains calendars assigned to price strategies.
PromotionPurgeBatchbatch	This batch process deletes old and rejected promotions.
PurgeExpiredExecutedOrApprovedClearancesBatch	This batch process deletes expired clearances in 'Executed' or 'Approved' statuses.
PurgeLocationMovesBatch	This batch process cleans up expired/executed location moves
PurgeUnusedAndAbandonedClearancesBatch	This batch process deletes unused and rejected clearances.
WorksheetAutoApproveBatch	This batch process approves maintain margin strategy worksheets that have not been acted upon by the end of the review period. The strategies must be marked as auto-approve in order to be processed.
ZoneFutureRetailPurgeBatch	This batch deletes past zone/item price change actions.

Batch Process Scheduling

Before setting up an RPM process schedule, familiarize yourself with Batch Schedule document published in conjunction with this release.

Threading and the RPM_BATCH_CONTROL Table

Some RPM batch processes use the RPM_BATCH_CONTROL table, which is a database administrator (DBA) maintained table and is populated by the retailer. This table defines the following:

- The batch process that is to be threaded.
- The number of threads that should be run at a time.
- How much data each thread should process (for example, 2 strategies per thread, 500 item/location/price changes by thread, and so on).

Each batch design later in this chapter states the following in its 'Threading' section:

- Whether the batch process utilizes the RPM_BATCH_CONTROL table.
- Whether or not the batch process is threaded.
- How the batch process is threaded (by strategy, by department, and so on).

Return Value Batch Standards

All batch processes in RPM conform to the Oracle Retail batch standards. They are executed and terminated in the same manner as other batch processes in the Oracle Retail suite of products. The following guidelines describe the return values that RPM's batch processes utilize:

Return Values

- 1 - The function completed without error.
- 0 - A fatal error occurred. The error messages are logged, and the process is halted.

Batch Logging

Relevant progress messages are logged with regard to batch program runtime information. The setting for these log messages is at the Info level in log4j.

For more information, see "Chapter 2 - Backend system administration and configuration" in the RPM 11.0.4 Operations Guide.

ItemLocDeleteBatch Batch

The ItemLocDeleteBatch program handles RMS deletions of item locations. When RMS deletes an item location, RPM now removes the Item/Location rows from the RPM_FUTURE_RETAIL table so that pricing events are no longer published out of RPM.

These item location deletions can be processed through either of two methods:

- The RMS_TABLE_RPM_ITL_AIR trigger
- A RIB message

These two options work the same as the batch and RIB modes for the New Item Location batch introduced in release 11.0.8.3. Documentation for switching between batch or RIB modes is found in the Release Notes for 11.0.8.3.

In the batch mode, the RPM_STAGE_DELETED_ITEM_LOC table is populated by the trigger RMS_TABLE_RPM_ITL_AIR. In RIB mode, the RPM_STAGE_DELETED_ITEM_LOC table is populated by subscribing to the itemlocmod and itemlocdel messages from RMS.

Usage

The following command runs the ItemLocDeleteBatch job:

```
ItemLocDeleteBatch.sh userid password
```

The first argument is the user ID and the second argument is the password.

Follow these steps to prepare to use this batch when the RIB is turned off:

1. Delete existing records from the table RPM_STAGE_DELETED_ITEM_LOC.
2. Enable the new trigger RMS_TABLE_RPM_ITL_AIR on table ITEM_LOC.
3. Stop the listener for Item/Location Creation RIB messages, as follows:
 - a. Log in to the Websphere administration console.
 - b. Select the RIBforRPM server from Servers > Application Servers.
 - c. Click the Message Listener Service link.
 - d. Click the Listener Ports link.
 - e. Select the check box next to ItemLocToRPMPort.
 - f. Click Stop. The listener is now stopped.
 - g. Click the ItemLocToRPMPort link to configure the port.
 - h. Select "Stopped" from the Initial State combo box.
 - i. Click OK.
 - j. Restart the RIBforRPM application server and verify that ItemLocToRPMPort is stopped.
4. Delete JMS subscriber in Egate, as follows:
 - a. Log in to the Egate Schema Manager.
 - b. Click the JMS Administrator button in the toolbar.
 - c. Expand the item etItmLocFromRMS.
 - d. Right-click on the RPM subscriber (it should have "RPM" in its name) and select "delete subscriber."

To reconfigure the system to process Item/Location deletion and modification messages through the RIB, follow these steps:

1. Modify the filter in the rib.properties file for RIBforRPM as follows, for RPM to subscribe to itemlocmod and itemlocdel:

```
tafr.types.filter.itemloc=ItemLocCre,ItemLocMod,ItemLocDel
```

2. Start the listener for Item/Location Creation RIB messages, as follows:
 - a. Log in to the Websphere administration console.
 - b. Select the RIBforRPM server from Servers > Application Servers.
 - c. Click the Message Listener Service link.
 - d. Click the Listener Ports link.
 - e. Select the check box next to ItemLocToRPMPort.
 - f. Click Start. The listener is now started.
 - g. Click the ItemLocToRPMPort link to configure the port.
 - h. Select "Started" from the Initial State combo box.
 - i. Click OK.
 - j. Restart the RIBforRPM app server and verify that ItemLocToRPMPort is started.
3. Create a JMS subscriber in Egate.

Note: No action is needed.

4. Disable the new trigger RMS_TABLE_RPM_ITL_AIR on ITEM_LOC.
5. Run the new batch (ItemLocDeleteBatch.sh) one time to process any records remaining in the RPM_STAGE_DELETED_ITEM_LOC table.

Scheduling Notes

This batch can be run ad hoc.

MerchExtractKickOffBatch Batch Design

Overview

The MerchExtractKickOffBatch.java batch program builds worksheets in RPM. MerchExtractKickOffBatch.java either creates or updates worksheets based on price strategies and the calendars attached to them.

Usage

The following command runs the MerchExtractKickOffBatch job:

```
MerchExtractKickOffBatch userid password
```

Where the first argument is the user id and the second argument is the password.

Detail

Setup: clean up expired worksheets and prepare for creation of new worksheets.

- Delete worksheets that are at the end of their review period.
- Get list of all strategies that need to be processed today. Create copies of the strategies as needed.
- Determine what strategies need to be grouped together based on the RPM_DEPT_AGGREGATION. WORKSHEET_LEVEL.

- Deal with overlapping strategies. Strategies can be setup at the DEPT, CLASS, or SUBCLASS. MerchExtractKickOffBatch.java ensures that only one strategy affects a given item.
- Start threads based on the values in RPM_BATCH_CONTRL for MerchExtractKickOffBatch.java.

Worksheet Creation:

- Call RPM_EXT_SQL, a PL/SQL package, to extract RPM information. The package is called at the strategy and RPM_DEPT_AGGREGATION. WORKSHEET_LEVEL. level. It pulls large amounts of data from various RMS tables and populates the RPM_WORKSHEET_DATA table. **The RPM_MERCH_EXTRACT_CONFIG table is used to exclude certain families of data from being included in the population. If this table is not populated all values are included in the population of RPM_WORKSHEET_DATA.**
- Read in the RPM_WORKSHEET_DATA records created by RPM_EXT_SQL. For each RPM_WORKSHEET_DETAIL record perform the following:
 - Use the price strategy to propose a retail value.
 - Apply candidate rules.
 - Apply price guides.

Potential reasons item/locations do not make it into a worksheet:

- The item/location falls under an exclusion type candidate rule.
- The item/location does not have a cost on RMS's FUTURE_COST table.
- The item's market basket codes vary across locations in a zone.
- The item's link code varies across locations in a zone.
- If a link code is identified on an item/location, and there is any item within that link code (at that location) that has not been brought into the worksheet, all of the item/locations with that link code are excluded from the worksheet.
- The item's selling unit of measure varies across locations in a zone.
- The item is part of an area differential item exclusion.
- Item/locations in a single link code have varying selling unit of measures.

If an item does not make it into a worksheet, a row is inserted into the RPM_MERCH_EXTRACT_DELETIONS table for each item location along with a reason that the item location was not included in the worksheet.

Assumptions and Scheduling Notes

The following programs must run before PriceStrategyCalendarBatch:

- PriceStrategyCalendarBatch
- LocationMoveBatch

Primary (RPM) Tables Involved

- RPM_WORKSHEET_STATUS
- RPM_WORKSHEET_DATA
- RPM_STRATEGY
 - RPM_STRATEGY_CLEARANCE
 - RPM_STRATEGY_CLEARANCE_MKDN
 - RPM_STRATEGY_COMPETITIVE
 - RPM_STRATEGY_DETAIL
 - RPM_STRATEGY_MARGIN
 - RPM_STRATEGY_REF_COMP
 - RPM_STRATEGY_WH
- RPM_AREA_DIFF
 - RPM_AREA_DIFF_EXCLUDE
 - RPM_AREA_DIFF_PRIM
 - RPM_AREA_DIFF_WH
- RPM_CALENDAR
 - RPM_CALENDAR_PERIOD
- RPM_CANDIDATE_RULE
 - RPM_CONDITION
 - RPM_VARIABLE
 - RPM_VARIABLE_DEPT_LINK
- RPM_PRICE_GUIDE
 - RPM_PRICE_GUIDE_DEPT
 - RPM_PRICE_GUIDE_INTERVAL

Threading

MerchExtractKickOffBatch.java is threaded. The RPM_BATCH_CONTROL table must include a record for MerchExtractKickOffBatch.java for it to run in threaded mode.

MerchExtractKickOffBatch.java is threaded by strategies and the RPM_DEPT_AGGREGATION. WORKSHEET_LEVEL setting.

PL/SQL Interface Point

- Package: RPM_EXT_SQL

NewItemLocationBatch batch design

Overview

The NewItemLocationBatch program replaces the Item/Location Creation RIB message. It ranges item locations by putting them into the future retail table. Item and location are fed to this program via the RPM_STAGE_ITEM_LOC table which is populated by an RMS process.

Usage

The following command runs the LocationMoveBatch job:

```
NewItemLocBatch userid password
```

Where the first argument is the user id and the second argument is the password.

Since this batch is a replacement for the Item/Location Creation RIB message, the following steps should be followed in preparation for using this batch in place of the RIB:

1. Delete existing records from table RPM_STAGE_ITEM_LOC
2. Enable the new trigger RMS_TABLE_RPM_ITL_AIR on table ITEM_LOC.
3. Stop the listener for Item/Location Creation RIB messages:
 - a. Login to Websphere admin console
 - b. Select the RIBforRPM server from *Servers* → *Application Servers*.
 - c. Click the "Message Listener Service" link.
 - d. Click the "Listener Ports" link.
 - e. Click the checkbox next to "ItemLocToRPMPort"
 - f. Click the "Stop" button. The listener is now stopped.
 - g. Click the "ItemLocToRPMPort" link to configure the port.
 - h. Select "Stopped" from the "Initial State" combo-box.
 - i. Click "OK"
 - j. Restart the RIBforRPM app server and verify that "ItemLocToRPMPort" is stopped.
4. Delete JMS subscriber in Egate
 - a. Login to the Egate Schema Manager
 - b. Click the "JMS Administrator" button in the toolbar.
 - c. Expand item "etItmLocFromRMS"
 - d. Right click on the RPM subscriber (it should have "RPM" in its name) and select "delete subscriber"
5. Add a record to table RPM_BATCH_CONTROL with PROGRAM_NAME of 'com.retek.rpm.batch.NewItemLocBatch' to control threading.

To reconfigure the system to process Item/Location Creation messages through the RIB, follow these steps:

6. Start the listener for Item/Location Creation RIB messages:
 - a. Login to Websphere admin console
 - b. Select the RIBforRPM server from *Servers* → *Application Servers*.
 - c. Click the “Message Listener Service” link.
 - d. Click the “Listener Ports” link.
 - e. Click the checkbox next to “ItemLocToRPMPort”
 - f. Click the “Start” button. The listener is now started.
 - g. Click the “ItemLocToRPMPort” link to configure the port.
 - h. Select “Started” from the “Initial State” combo-box.
 - i. Click “OK”
 - j. Restart the RIBforRPM app server and verify that “ItemLocToRPMPort” is stopped.

7. Create JMS subscriber in Egate

Note: No action is needed.

8. Disable the new trigger RMS_TABLE_RPM_ITL_AIR on ITEM_LOC.
9. Run the new batch (NewItemLocBatch.sh) one time to process any records remaining in RPM_STAGE_ITEM_LOC.

Detail

The batch selects rows from the stage table and updates the FUTURE_RETAIL table to reflect the new item/location combination. If any approved price changes/promotions/clearances exist at a parent/zone level that encompasses the new item/location, these are also added to the FUTURE_RETAIL table for the new item/location.

Assumptions and Scheduling Notes

This batch may be run ad-hoc.

Primary Tables Involved

- RPM_STAGE_ITEM_LOC
- RPM_FUTURE_RETAIL

Threading

This program is threaded. If no row exists in the RPM_BATCH_CONTROL table for com.retek.rpm.batch.NewItemLocBatch, then the application is executed with one thread and transactions are committed for each item-loc combination.

ZoneFutureRetailPurgeBatch Batch Design

Overview

The ZoneFutureRetailPurgeBatch program deletes old error message from the price change auto approve batch program.

Usage

The following command runs the PriceChangeAutoApproveResultsPurgeBatch job:

```
ZoneFutureRetailPurgeBatch userid password
```

Where the first argument is the user id and the second argument is the password.

Detail

The ZoneFutureRetailPurgeBatch program deletes all zone/item price change actions which:

10. Have an ACTION_DATE value prior to VDATE, and
11. Have been superseded by at least one other change action whose ACTION_DATE is also prior to VDATE.

The effect is that, for each zone/item with a price change history, the most recent such action, prior to VDATE, will remain after the purge is complete.

Assumptions and Scheduling Notes

ZoneFutureRetailPurgeBatch can be run ad hoc.

Primary Tables Involved

- RPM_ZONE_FUTURE_RETAIL

Threading

This program is not threaded.