

**Oracle® Retail Price Management**  
Operations Guide Addendum  
Release 11.0.10

May 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

<b>Preface .....</b>	<b>v</b>
Audience .....	v
Related Documents .....	v
Customer Support .....	v
Conventions .....	vi
<b>1 Introduction .....</b>	<b>1</b>
<b>2 Batch Processes .....</b>	<b>3</b>
Java Batch Processes.....	3
Java Batch Process Architectural Overview .....	3
Running a Java-Based Batch Process .....	3
Script Catalog .....	4
Scheduler and the Command Line .....	5
Functional Descriptions and Dependencies .....	5
Batch Process Scheduling.....	6
Threading and the RPM_BATCH_CONTROL Table.....	6
Return Value Batch Standards .....	6
Return Values .....	6
Batch Logging .....	6
Price Event Execution Batch Processes .....	7
PriceChangeAreaDifferentialBatch Batch Design .....	8



---

---

# 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 11.0.10 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
- Oracle Retail Price Management Batch Schedule

## 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* and any subsequent RPM 11.0.x Operations Guide Addendums. (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 the *Oracle Retail Price Management 11.0.4 Operations Guide* provides retailers with a complete overview of the application.

For the Oracle Retail Price Management 11.0.10 release, there is one new batch design (PriceChangeAreaDifferentialBatch) and one updated batch design (PriceEventExecutionBatch). Modifications to existing content 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 11.0.10 Release Notes.





---

# 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” or any subsequent RPM 11.0.x Operation Guide Addendums.

## 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
<code>itemLocDeleteBatch.sh</code>	<code>ItemLocDeleteBatch</code>
<code>locationMoveBatch.sh</code>	<code>LocationMoveBatch</code>
<code>merchExtractKickOffBatch.sh</code>	<code>MerchExtractKickOffBatch</code>
<code>newItemLocBatch.sh</code>	<code>NewItemLocBatch</code>
<code>PriceChangeAreaDifferentialBatch.sh</code>	<code>PriceChangeAreaDifferentialBatch</code>
<code>priceChangeAutoApproveResultsPurgeBatch.sh</code>	<code>PriceChangeAutoApproveResultsPurgeBatch</code>
<code>priceChangePurgeBatch.sh</code>	<code>PriceChangePurgeBatch</code>
<code>priceChangePurgeWorkspaceBatch.sh</code>	<code>PriceChangePurgeWorkspaceBatch</code>
<code>priceEventExecutionBatch.sh</code>	<code>PriceEventExecutionBatch</code>
<code>priceEventExecutionDealsBatch.sh</code>	<code>PriceEventExecutionDealsBatch</code>
<code>priceEventExecutionRMSBatch.sh</code>	<code>PriceEventExecutionRMSBatch</code>
<code>priceStrategyCalendarBatch.sh</code>	<code>PriceStrategyCalendarBatch</code>
<code>promotionPurgeBatchbatch.sh</code>	<code>PromotionPurgeBatchbatch</code>
<code>purgeExpiredExecutedOrApprovedClearancesBatch.sh</code>	<code>PurgeExpiredExecutedOrApprovedClearancesBatch</code>
<code>purgeLocationMovesBatch.sh</code>	<code>PurgeLocationMovesBatch</code>
<code>purgeUnusedAndAbandonedClearancesBatch.sh</code>	<code>PurgeUnusedAndAbandonedClearancesBatch</code>
<code>worksheetAutoApproveBatch.sh</code>	<code>WorksheetAutoApproveBatch</code>
<code>launchRpmBatch.sh</code>	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.

Script	Batch program executed
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.
PriceChangeAreaDifferentialBatch	This batch program creates price changes for Area Differential price strategies.
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.
PriceEventExecutionDealsBatch	This batch process processes the deals affected by the price events being executed.
PriceEventExecutionRMSBatch	This batch process processes the item-locations affected by the price events being executed RMS.
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.

Batch processes	Details
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.

## Price Event Execution Batch Processes

### Overview

The price event execution batch processes perform the necessary work to start (regular price change, clearance price change, promotions) and end (price change, promotions) pricing events.

Executing price events require running three batch programs. These are:

- PriceEventExecutionBatch.java identifies the events that need to be executed and stages the affected item-locations for the next batch to process. **If this batch fails to process a particular price event, that event will remain in “approved” status and the next-day batch run is guaranteed to pick up this failed price event for re-processing.**
- PriceEventExecutionRMSBatch.java processes the item-locations affected by the price events being executed RMS. **If this batch fails to process a particular item-location for one or more price events, the affected events will be in “executed” status and the item-locations that failed to process will remain staged in RPM\_EVENT\_ITEMLOC. These item-locations will be picked up again by the next-day batch run.**
- PriceEventExecutionDealsBatch.java processes the deals affected by the price events being executed. **If this batch fails to process a particular item-location deal for one or more price events, the affected events will be in “executed” status and their associated item-locations are posted in RMS ITEM\_LOC and PRICE\_HIST tables. However, the item-location deals that failed to process will remain in RPM\_EVENT\_ITEMLOC\_DEALS and the next-day batch run is guaranteed to pick these up.**

### Usage

The following commands will need to be executed in order:

```
PriceEventExecutionBatch userid password
PriceEventExecutionRMSBatch userid password
PriceEventExecutionDealsBatch userid password
```

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

### Detail

The batch programs process regular price changes, clearance price changes, and promotions events that are scheduled for the run date. **Restartability features allow events missed in past runs of the batch to be picked up in later runs. When posting information in the ITEM\_LOC and PRICE\_HIST table, the batch process will respect the active dates of their associated price events.**

- Promotions:
  - Promotions that are scheduled to start are activated. **These include all approved promotions whose start dates are <= VDATE+1.**
  - Promotions that are scheduled to end are completed. **These include all active promotions whose end dates are <= VDATE.**
- Clearances:
  - Clearance markdowns that are scheduled to take place are executed. **These include all clearances whose effective dates are <= VDATE+1.**
  - Clearances that are scheduled to be completed (reset) are completed.

- Regular price changes:
  - Regular price changes that are scheduled to take place are executed. **These include all price changes whose effective dates are  $\leq$  VDATE+1.**

### Assumptions and Scheduling Notes

The batch processes must run in the following order:

- PriceEventExecutionBatch
- PriceEventExecutionRMSBatch
- PriceEventExecutionDealsBatch

The **previous three** processes must run before the following programs:

- Storeadd (RMS)
- MerchExtractKickOffBatch

The following programs must run before **the PriceEventExecution** batch processes:

- Salstage (RMS)
- LocationMoveBatch

### Primary tables involved

- RPM\_PRICE\_CHANGE
- RPM\_CLEARANCE
- RPM\_PROMO\_COMP\_DETAIL

### RMS interface point

The PriceEventExecutionRMSBatch interfaces with the RMS price change subscription package RMSSUB\_PRICE\_CHANGE. All price change, clearance, and promotion prices will be passed along to this RMS package at the item location level and will be applied in RMS.

### Threading

Two of the three batch programs involved in price event execution utilize concurrent processing. These are PriceEventExecutionBatch and PriceEventExecutionRMSBatch. The threading strategies for these two batch programs are different from each other.

PriceEventExecutionBatch is threaded by a variable number of pricing events to be executed (i.e., price changes, clearances, and promotions).

PriceEventExecutionRMSBatch is threaded by a variable number of item-locations affected by the pricing events to be executed.

## PriceChangeAreaDifferentialBatch Batch Design

### Overview

The PriceChangeAreaDifferentialBatch program creates price changes for Area Differential price strategies.

### Usage

The following command runs the PriceChangeAreaDifferentialBatch job:

```
PriceChangeAreaDifferentialBatch userid password
```

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

**Detail**

The PriceChangeAreaDifferentialBatch creates price changes for all secondary areas in the RPM\_AREA\_DIFF\_PC\_CREATE table. Only items that have a STATE of 'N' will have price changes created.

**Assumptions and Scheduling**

PriceChangeAreaDifferentialBatch can be run ad hoc.

**Primary Tables Involved.**

RPM\_AREA\_DIFF\_PC\_CREATE

**Threading**

This program does thread the price change approval process if the strategy has the Auto Approve option selected.