# Oracle® Retail Price Management
Release Notes
Release 12.0.4

June 2007

**ORACLE**

Oracle® Price Management Release Notes, Release 12.0.4

## Value-Added Reseller (VAR) Language

(i) the software component known as **ACUMATE** developed and licensed by Lucent Technologies Inc. of Murray Hill, New Jersey, to Oracle and imbedded in the Oracle Retail Predictive Application Server – Enterprise Engine, Oracle Retail Category Management, Oracle Retail Item Planning, Oracle Retail Merchandise Financial Planning, Oracle Retail Advanced Inventory Planning and Oracle Retail Demand Forecasting applications.

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

(iii) the **SeeBeyond** component developed and licensed by Sun MicroSystems, Inc. (Sun) of Santa Clara, California, to Oracle and imbedded in the Oracle Retail Integration Bus application.

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

(v) the software component known as **Crystal Enterprise Professional and/or Crystal Reports Professional** licensed by Business Objects Software Limited ("Business Objects") and imbedded in Oracle Retail Store Inventory Management.

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

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

(viii) the software component known as **Style Report™** developed and licensed by InetSoft Technology Corp. of Piscataway, New Jersey, to Oracle and imbedded in the Oracle Retail Value Chain Collaboration application.

(ix) the software component known as **i-net Crystal-Clear™** developed and licensed by I-NET Software Inc. of Berlin, Germany, to Oracle and imbedded in the Oracle Retail Central Office and Oracle Retail Back Office applications.

(x) the software component known as **WebLogic™** developed and licensed by BEA Systems, Inc. of San Jose, California, to Oracle and imbedded in the Oracle Retail Value Chain Collaboration application.

(xi) the software component known as **DataBeacon™** developed and licensed by Cognos Incorporated of Ottawa, Ontario, Canada, to Oracle and imbedded in the Oracle Retail Value Chain Collaboration application.

# Preface

A Release Notes document can include some or all of the following sections, depending upon the release:

- Overview of the release
- Functional, technical, integration, and performance enhancements
- Assumptions
- Fixed defects
- Known issues

## Audience

Release Notes are a critical communication link between Oracle Retail and its retailer clients. There are four general audiences in general for whom a Release Notes document is written:

- Retail clients who want to understand the contents of this release
- Staff who have the overall responsibility for implementing Oracle Retail Price Management in their enterprise
- Business analysts who want high-level functional information about this release
- System analysts and system operation personnel who want high-level functional and technical content related to this release

## Related Documents

For more information, see the following documents in the Oracle Retail Price Management Release 12.0.4 documentation set:

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

## Customer Support

https://metalink.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

For a base release (".0" release, such as 12.0), Oracle Retail strongly recommends that you read all patch documentation before you begin installation procedures. Patch documentation can contain critical information related to the base release, based on new information and code changes that have been made since the base release.

## Oracle Retail Documentation on the Oracle Technology Network

In addition to being packaged with each product release (on the base or patch level), all Oracle Retail documentation is available on the following Web site:

http://www.oracle.com/technology/documentation/oracle_retail.html

Documentation should be available on this Web site within a month after a product release. Note that documentation is always available with the packaged code on the release date.

## 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.

# Release Notes

## Overview

This document describes the enhancements and major defect fixes that were made to this release of Oracle Retail Price Management (RPM).

## Trigger RMS_TABLE_RPM_ITL_AIR

In the script s5566354b.sql in the RPM 12.0.3 release, the following statement was commented out:

```
DROP TRIGGER RMS_TABLE_RPM_ITL_AIR;
```

If the trigger RMS_TABLE_RPM_ITL_AIR is still present, it must be dropped.

## Noteworthy Enhancements

### System Option—Future Retail Seed Days Before VDATE

A system option was added to create a seed record in the RPM_FUTURE_RETAIL table for a newly ranged item/location that is inheriting an active promotion. The option fut_ret_seed_days_before specifies the future retail seed days before vdate.

The number of days entered in the field should ensure that the seed record date that is created for this new item/location will be inserted in the RPM_FUTURE_RETAIL table with a date that is prior to the start of the active promotion that the new item/location is inheriting.

When a new item/location from RMS is ranged into the RPM system, a seed record is created with an action_date of vdate minus fut_ret_seed_days_before. The fut_ret_seed_days_before value must be large enough to insure that the seeded action_date precedes all possible active promotion detail start_dates that may be applied to the new item/location.

System default: 365

Valid values: 1 to 999

### Price Changes for Area Differentials

There is a performance fix for creating price changes for the area differential price strategy. The creation of price changes has been removed from the price strategy functional screen and moved to the price change task dialog.

Because the creation of price changes from the area differential strategy has the potential to create large numbers of price changes, real-time generation was causing a performance issue. To improve performance in this area, a staging table and a batch process have been added.

A new batch control script (c5982978_Batch_control_seed.sql) has been added for the preprocessing of the new PriceChangeAreaDifferentialBatch batch. This script must be run for this batch to process succesfully.

**Detailed user experience:**

On the Price Change dialog in the task GUI, there is an option listed in the workflow titled Price Changes for Area Differentials.

When the user selects this task, the top multi-record block displays all primary areas that exist for current area differential strategies. When the user selects the primary area, the bottom section of the screen displays the secondary areas that exist for the primary area strategy.

Selecting a secondary area in the multi-record block enables the Apply block and allows the user to click the Create Price Changes box and select an effective date. The calendar LOV uses price change processing days to default the first available day the user can select. If the user has emergency price change security, the user can select a day that falls between the VDATE and the first available selection date.

When the user clicks Apply, and then Done, price change records are written to a staging table called RPM_AREA_DIFF_PC_CREATE. Records remain in this table until a daily batch named priceChangeAreaDifferentialBatch.sh runs and pick up all records that are in the staging table. Only those records with a state of New (N) are selected. The batch process creates price changes in either Worksheet or Approved state, depending on whether the user has enabled the Auto Approve check box in the dialog. Price changes are generated with the effective date that was selected by the user.

The state of the request in the Area Differential price change dialog remains New until the daily batch creates manual price changes in the price change dialog. If a secondary area has a state of New, the user cannot edit or update this request. This restriction eliminates the potential for duplicate generation of price changes. When the manual price changes are generated, the state of New is updated to Complete, Complete with Errors, or Error.

## PriceEventExecutionBatch Batch Design

The original design of the price event execution batch suffered from performance issues caused by database contention and unbalanced thread workload distribution.

The batch process has been split into three batch programs:

- PriceEventExecutionBatch.java identifies the events that need to be executed and their affected item-locations.
- PriceEventExecutionRMSBatch.java processes the item-locations affected by the price events being executed RMS.
- PriceEventExecutionDealsBatch.java processes the deals affected by the price events being executed.

Threading for PriceEventExecutionBatch.java is still done for each price event to be executed.

Threading for PriceEventExecutionRMSBatch.java is done at a variable number of item-locations affected by the price events to be executed.

## New Permissions

Scripts are included to add the following new permissions:

- rpm.marketBasketCodesCM.createMarketBasketCode
- rpm.linkCodesCM.createLinkCode
- rpm.promotionConstraintsCM.create

- rpm.promotionConstraintsCM.maintain
- rpm.priceChangesCM.areaDiffs

These permissions are added to the RSM database by the following scripts:

- c5935795_rsm_permissions.sql
- c5982978_RSM_seed_data_for_named_permissions.sql

Note that the scripts only add the permissions to the RSM tables. To have these permissions reflected in RPM, they have to be assigned manually in the RSM application to the roles that are set up in the ROLE_NAMED_PERMISSION table.

## Merch Extract Permissions

The Merch Extract batch program was modified to use a pre-process to stage data to make the rest of its execution faster. The pre-process consists of populating pre-tables that are organized in a way optimal for the Merch Extract batch program.

During the pre-process, several operations are performed that are not typically used by our application code, such as table truncations and index alters. The privileges to perform these operations are typically not granted to the application database user. However, since these operations are contained in a PL/SQL package (RPM_EXT_SQL), we rely on the PL/SQL concept of definer's right. The package runs with the privileges of its definer, not with the privileges of the invoker.

## Status Page Batch Update

The status page batch program (statusPageCommandLineApplication.sh) was enhanced to perform some 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; each check should return 0 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 the RPM_FUTURE_RETAIL table per item, location, and action date.

## statusPageCommandLineApplication.sh

The new command usage is as follows:

```
statusPageCommandLineApplication.sh username password [phase-choice]
 [max-rows-choice]
```

Valid values for *phase-choice* are as follows:

| | |
|---|---|
| S | System check only |
| D | Data integrity check only |
| B (default) | Both |

The value specified for *max-rows-choice* is the maximum row count for the query. By default, the query is run for the full count.

For example:

```
./statusPageCommandLineApplication.sh alain.frecon retek S
```

Here is sample output of the batch program.

```
Performing System Check
The following RpmRibMessageStatusException is normal.
We need to throw an exception to ensure that the test messages are rolled back.
10:30:04,599 ERROR [ServiceAccessor] InvocationTargetException received on a
service call...
java.lang.reflect.InvocationTargetException
at sun.reflect.NativeMethodAccessorImpl.invoke0(Native Method)
at sun.reflect.NativeMethodAccessorImpl.invoke(NativeMethodAccessorImpl.java:79)
at
sun.reflect.DelegatingMethodAccessorImpl.invoke(DelegatingMethodAccessorImpl.java(
Compiled Code))
at java.lang.reflect.Method.invoke(Method.java(Compiled Code))
at org.apache.commons.beanutils.MethodUtils.invokeMethod(MethodUtils.java:216)
at
com.retek.platform.service.ServiceAccessor.callRemoteMethod(ServiceAccessor.java:3
00)
at
com.retek.platform.service.ServiceAccessor.remoteTransaction(ServiceAccessor.java:
485)
at
com.retek.platform.service.ServiceAccessorProxy.invoke(ServiceAccessorProxy.java:5
1)
at $Proxy4.performRibMessageCheck(Unknown Source)
at com.retek.rpm.statuspage.RpmRibMessageCheck.execute(RpmRibMessageCheck.java:25)
at com.retek.rpm.statuspage.StatusPageCheck.runTest(StatusPageCheck.java:15)
at
com.retek.rpm.statuspage.StatusPageProcessor.execute(StatusPageProcessor.java:19)
at
com.retek.rpm.statuspage.StatusPageCommandLineApplication.performAction(StatusPage
CommandLineApplication.java:80)
at
com.retek.rpm.statuspage.StatusPageCommandLineApplication.main(StatusPageCommandLi
neApplication.java:65)
Caused by:
<com.retek.rpm.app.statuspage.service.RpmRibMessageStatusException>
<message>
  No cause associated
</message>
</com.retek.rpm.app.statuspage.service.RpmRibMessageStatusException>

at
com.retek.rpm.app.statuspage.service.StatusPageAppServiceImpl.performRibMessageChe
ck(StatusPageAppServiceImpl.java:71)
at sun.reflect.NativeMethodAccessorImpl.invoke0(Native Method)
at
sun.reflect.NativeMethodAccessorImpl.invoke(NativeMethodAccessorImpl.java(Compiled
Code))
at
sun.reflect.NativeMethodAccessorImpl.invoke(NativeMethodAccessorImpl.java(Compiled
Code))
at
sun.reflect.DelegatingMethodAccessorImpl.invoke(DelegatingMethodAccessorImpl.java(
Compiled Code))
at java.lang.reflect.Method.invoke(Method.java(Compiled Code))
at org.apache.commons.beanutils.MethodUtils.invokeMethod(MethodUtils.java(Compiled
Code))
```

```
at
com.retek.platform.service.ServiceCommandImpl.execute(ServiceCommandImpl.java(Comp
iled Code))
at
com.retek.platform.service.impl.CommandExecutionServiceEjb.executeCommand(CommandE
xecutionServiceEjb.java(Compiled Code))
at
com.retek.platform.service.impl.EJSRemoteStatelessCommandExecutionService_76208b17
.executeCommand(Unknown Source)
at
com.retek.platform.service.impl._EJSRemoteStatelessCommandExecutionService_76208b1
7_Tie.executeCommand__com_retek_platform_service_ServiceCommand(_EJSRemoteStateles
sCommandExecutionService_76208b17_Tie.java(Compiled Code))
at
com.retek.platform.service.impl._EJSRemoteStatelessCommandExecutionService_76208b1
7_Tie._invoke(_EJSRemoteStatelessCommandExecutionService_76208b17_Tie.java(Compile
d Code))
at
com.ibm.CORBA.iiop.ServerDelegate.dispatchInvokeHandler(ServerDelegate.java(Compil
ed Code))
at com.ibm.CORBA.iiop.ServerDelegate.dispatch(ServerDelegate.java(Compiled Code))
at com.ibm.rmi.iiop.ORB.process(ORB.java(Compiled Code))
at com.ibm.CORBA.iiop.ORB.process(ORB.java(Compiled Code))
at com.ibm.rmi.iiop.Connection.doWork(Connection.java(Compiled Code))
at com.ibm.rmi.iiop.WorkUnitImpl.doWork(WorkUnitImpl.java(Compiled Code))
at com.ibm.ejs.oa.pool.PooledThread.run(ThreadPool.java(Compiled Code))
at com.ibm.ws.util.ThreadPool$Worker.run(ThreadPool.java(Compiled Code))
*****************************************
Starting Report
com.retek.rpm.statuspage.RsmServerCheck Passed
*****************************************
Starting Report
com.retek.rpm.statuspage.RpmLoginCheck Passed
*****************************************
Starting Report
com.retek.rpm.statuspage.RpmDataAccessCheck Passed
*****************************************
Starting Report
com.retek.rpm.statuspage.RpmRibMessageCheck Passed REGPRCCHG.REGPRCCHGCRE is ON
************************The above exception indicates that we have passed
**************************
*****************************************
Starting Report
com.retek.rpm.statuspage.RpmRibMessageCheck Passed REGPRCCHG.REGPRCCHGMOD is ON
************************The above exception indicates that we have passed
**************************
*****************************************
Starting Report
com.retek.rpm.statuspage.RpmRibMessageCheck Passed REGPRCCHG.REGPRCCHGDEL is ON
************************The above exception indicates that we have passed
**************************
*****************************************
Starting Report
com.retek.rpm.statuspage.RpmRibMessageCheck Passed CLRPRCCHG.CLRPRCCHGCRE is ON
************************The above exception indicates that we have passed
**************************
*****************************************
Starting Report
com.retek.rpm.statuspage.RpmRibMessageCheck Passed CLRPRCCHG.CLRPRCCHGMOD is ON
************************The above exception indicates that we have passed
**************************
*****************************************
Starting Report
```

```
com.retek.rpm.statuspage.RpmRibMessageCheck Passed CLRPRCCHG.CLRPRCCHGDEL is ON
**********************The above exception indicates that we have passed
**************************
*****************************************
Starting Report
com.retek.rpm.statuspage.RpmRibMessageCheck Passed PRMPRCCHG.PRMPRCCHGCRE is ON
***********************The above exception indicates that we have passed
**************************
*****************************************
Starting Report
com.retek.rpm.statuspage.RpmRibMessageCheck Passed PRMPRCCHG.PRMPRCCHGMOD is ON
***********************The above exception indicates that we have passed
**************************
*****************************************
Starting Report
com.retek.rpm.statuspage.RpmRibMessageCheck Passed PRMPRCCHG.PRMPRCCHGDEL is ON
***********************The above exception indicates that we have passed
**************************
*****************************************
Starting Report
RpmJmsServerCheck Passed

Done.
```

# Noteworthy Fixes

## Bugs 5944561 and 6008606—Pricing Strategy Overlap

Pricing strategies allow users to define how prices are proposed when pricing worksheets are generated.

A functional change in this release allows multiple pricing strategies to overlap; however they cannot have calendars that overlap. The multiple strategies functionality validates current or future calendars to ensure that an item/location combination for a specific zone exists in only one strategy for a given calendar day.

This change impacts the following RPM pricing strategies, which can all be active strategies for an item/location as long because their strategy calendars do not overlap:

- Clearance
- Competitive
- Margin

## Bug 5935795—RPM Permissions Not Executing as Designed in RSM

The two workflows of Promotion Constraints have been separated to be controlled by two different permissions. The maintenance screen now honors the IS_VIEW and IS_EDIT permissions. This means that if a user only has the IS_VIEW permission, the screen is accessible but read-only. If a user has IS_EDIT, the user can modify promotion constraints. New permissions and code to use them were added to workflows for Create Market Basket Codes and Create Link Codes to serve the same purposes.

## Bug 5971471—Enable Price Event Execution Batch to Process Past-Date Events

The Price Execution batch can now pick up failed item/locations from the previous day's runs, once the user has fixed whatever issues caused the original failures. Changes to the priceExecutionBatch.sh batch pick up approved price changes, promotions, and clearances with start and effective dates before the active date, as well as promotions with end dates before the active date. Changes to the priceExecutionRMSBatch.sh batch ensure that item/locations are processed according to active date when processing same item locations. The revised batch process respects the active dates of their associated price events, in case the batch process picks up two events, one VDATE+1, and another in the past but affecting the same set of item/locations.

Examples:

- If priceEventExecutionBatch.sh fails to process a particular price event, that event remains in "approved" status. The next-day batch run will pick up this failed price event for reprocessing.

- If priceEventExecutionRMSBatch.sh fails to process a particular item/location for one or more price events, the affected events are in "executed" status and the item/locations that failed to process remain staged in RPM_EVENT_ITEMLOC. These item/locations are picked up again by the next-day batch run.

- If priceEventExecutionDealsBatch.sh fails to process a particular item/location deal for one or more price events, the affected events are in "executed" status, and their associated item/locations are posted in RMS ITEM_LOC and PRICE_HIST tables. However, the failed item/location deals that failed to process remain in RPM_EVENT_ITEMLOC_DEALS. The next-day batch run picks these up again.

## Bug 5918935—New Item/Loc Batch Includes Active Promotions When Ranging Item/Locs

Changes have been made so that the date written into the future retail seed record is configurable. A new system option (FUT_RET_SEED_DAYS_BEFORE) controls the date that is used. Seed records are created on the date (VDATE minus FUT_RET_SEED_DAYS_BEFORE), that is, FUT_RET_SEED_DAYS_BEFORE days before vdate. It is expected that FUT_RET_SEED_DAYS_BEFORE will be used by clients to ensure that the seed record dates fall before the start date of any active promotions that might exist in their systems.

## Bug 5995935—On Linux, No Data Returned for a Promotion Search by Promotion ID

## Bug 5995935—On AIX, Display Error for a Promotion Search by Promotion ID

RPM bug 5995935 was filed because a search for a promotion by Promotion ID did not return the expected results. To correct this defect, apply Oracle database patch 5921386.

# Known Issues

### Defect 329 (Bug 5995871) (Priority 3): Browser Cache—Load Balancing Issue

**Description:** The Web browser can cache the jnlp file. In the event of a failover, the load balancer would generate the correct jnlp file, but the browser cache might prevent this version from being picked up.

**Resolution:** Development is actively working on a fix that should be available in a forthcoming hot fix.

**Workaround:** There is an additional file called rpm-jnlp-nocache.html in the same directory as rpm.jnlp. This file can be used to launch RPM, as an alternative to the rpm.jnlp file. This file performs a redirect to the JNLP to launch the client, using a GET parameter that makes the request URL unique every time. This should prevent the browser from using a cached version of the rpm.jnlp file. Another workaround is to delete cookies and clear the browser cache before launching rpm.jnlp.

### Bug 6008527 (Priority 3): Intermittent IllegalStateException upon Login

**Description:** An intermittent error occurs immediately after login: IllegalStateException

This error is caused by a bug in the polling logic for pulling alerts into the RPM client. The error only occurs immediately after login.

**Resolution:** Development is actively working on a fix that should be available in a forthcoming hot fix.

**Workaround:** Click the Continue button. The application should function without any problems after that.