Oracle® Retail Price Management

Release Notes

Release 13.1.3.1

September 2010

Oracle Retail Price Management (RPM) Release 13.1.3.1 is a bundled hot fix release for RPM 13.1. RPM 13.1.3.1 includes selected defect fixes for RPM 13.1 code.

Oracle Customer Support investigates submitted issues assuming that all released updates have been applied. It is the customer's decision when to apply a new release; however, delays in applying updates can complicate the support process.

About Bundled Hot Fix Releases

Oracle Retail bundled hot fix releases are the most frequent releases to update Oracle Retail applications. These releases typically contain only defect fixes; they may also include enhancements that address application performance or other issues. Bundled hot fix releases are of smaller scope than less-frequent patch releases, and they are intended to be easier and faster for the customer to install than major patch updates.

Each bundled hot fix release contains a unique set of defect fixes or enhancements. Note that bundled hot fix releases are not cumulative; that is, defect fixes from a previous bundled hot fix release are not also included in a later bundled hot fix release.

Bundled hot fix releases must be applied in sequential order. Before you can apply a new bundled hot fix release, you must first apply all previous bundled hot fix releases since the last patch release. The Release Notes for each Oracle Retail release identify whether a release is a bundled hot fix release or a patch release.

Periodic patch releases include all defect fixes that have been released through bundled hot fix releases since the last patch release. Patch releases may also include new defect fixes and enhancements that have not previously been included in any bundled hot fix release.

Applying Database Change Scripts

When applying database change scripts, there are some dependencies between scripts. Be sure that scripts named with a pattern such as 12345_rpm_table_name.sql are applied before applying scripts named with a pattern like 12345a_rpm_table_name.sql (note the "a" after the 12345). If the application of these scripts is reversed, the effects of one of the scripts could be lost.

Running Scripts

Back up data before running any script, because the scripts provided *do not* preserve data. See defect reports for details.

Check with your database administrator to determine whether your database should be analyzed after a script is run. In many cases, analysis of the database is necessary to take advantage of new or modified indexes intended to improve performance of the application.

Functional Enhancements

The following functional enhancements are included in this hot fix release.

ItemList Level Price Event

In previous releases, when the user selects an ItemList in the item selection panel to create a price event, RPM explodes the ItemList into all items in that ItemList and creates the price events at the item level. So if the ItemList contains 1,000 items, for example, RPM creates 1,000 price events. For a very large ItemList, this could potentially cause UI performance degradation.

This functional enhancement allows the user to create an ItemList level Simple Promotion, Threshold Promotion, Price Change, and Clearance. From the example above, if the ItemList contains 1,000 items and the user decides to create the price event at the ItemList level, RPM creates only one price event that affects all 1,000 items in the ItemList.

But this is a "point in time" ItemList. RPM captures the items in the ItemList during the creation of the price event (when the price event is saved to the database) and will use the captured list of items throughout the lifetime of the price event. After the price event is created and saved, adding or removing items to/from the ItemList (in RMS) does not impact the price event anymore.

For this enhancement, a new system option, Allow Item List Explosion, has been added. If this system option is unchecked, the user cannot explode the ItemList into items. When the user selects an ItemList when creating Simple Promotions, Threshold Promotions, Price Changes, or Clearances, the price event is created at ItemList level. If this system option is checked, the user has the option to explode the ItemList into its items. If the user chooses to do that, RPM creates item level price events accordingly.

In addition, the threading logic for the Price Event Execution batch has been changed from Price Event ID to Item/Location. Previously, this batch job split into several threads based on the number of Price Events that need to be processed. This number is represented in the THREAD_LUW_COUNT column of the "com.retek.rpm.batch.PriceEventExecutionBatch" entry in the RPM_BATCH_CONTROL table. With this enhancement, the batch job splits into several threads based on the number of Items/Locations. So the user must update the THREAD_LUW_COUNT column of the "com.retek.rpm.batch.PriceEventExecutionBatch" entry in the RPM_BATCH_CONTROL table to represent how many Item/Locations must be processed by each thread for this batch job.

Skip Conflict Checking during Submit

This enhancement provides an option for the user to skip conflict checking process during the price event Submit action. A new system option, "Do not run Conflict Check for Submit," has been added. 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 uses existing functionality to execute the conflict checking process (and show conflicts found) when the user performs the price event Submit action.

Skip Conflict Checking during Multi-Buy Approval

This enhancement provides an option for the user to skipthe conflict checking process during Multi-Buy Promotion approval (and un-approval). When the user approves (or un-approves) a Multi-Buy promotion, the system does not update the Future Retail table and does not perform conflict validation. But it populates the Payload tables so that RPM can communicate the Multi-Buy promotion to other systems (through RIB or the RPOS extract, for example).

A new system option "Do not run Conflict Check for Multi-Buy Promotion Approval" has been added for this. If this system options is checked then RPM would not perform conflict checking process (hence not showing any conflict if any) when the user approves (or un-approves) a Multi-Buy promotion. If this system option is un-checked, RPM uses existing functionality to execute a complete conflict checking process (and show conflicts found) when the user approves (or un-approves) a Multi-Buy promotion.

Choosing this option can improve the very large Multi-Buy promotion (for example, Department/Zone Level) approval dramatically. But retailers should consider this very carefully, because some functionality is lost with this option. For example:

- Overlapping promotions: Item/Location in the Multi-Buy promotion is not be considered when validating the overlapping promotions.
- Price Change Promotion Overlap/Clearance Promotion Overlap: Item/Location in the Multi-Buy promotion is considered when validating these types of overlapping.
- The Multi-Buy promotion information is not available in the Future Retail tables.

Although the new system option can be changed from one status to the other in the System Options Edit screen, flipping the system option while having active or pending Multi-Buy promotion in the system could produce unpredictable results.

Unique Unit of Measure

This enhancement provides the ability to inform the system that there is only one unit of measure (UOM) to be used throughout the system. This is particularly useful in improving the performance when creating a "fixed price" price event. Previously, when the user created a "fixed price" price event without entering a new UOM, the system validated that all item/locations affected by the price event had a unique UOM. When the price event is very large (for example, a Department/Zone level promotion), this validation process could cause some performance degradation.

For this enhancement, two system options are added. The "Unique UOM is used for all items" system option informs the system that there will is only one UOM in the system. When this system option is checked, the system does not perform UOM validation when the user creates a "fixed price" price event. If this system options is checked, the user must provide the valid unique UOM to be used throughout the system in the "Unique UOM to be used for all items" system option. These two system options are to be set during the RPM installation. For an existing RPM environment, these system options should be updated from the backend by the DBA.

Chunking to Bulk Conflict Checking Ratio

When running a conflict checking process, the system decides whether it should run a Chunking Conflict Checking or Bulk Conflict Checking , based on the following formula:

Number of potential Item/Location in the price event > X * Bulk Conflict Check LUW.

(The factor, X, is currently set at 2.5, and the user must change the codes, as required.)

If the above condition is True, the system triggers the Chunking Conflict Checking process; otherwise, it continues the Bulk Conflict Checking process. "Bulk Conflict Check LUW" is represented by the THREAD_LUW_COUNT column of the "com.retek.rpm.app.bulkcc.service.BulkConflictCheckAppService" entry in the RPM_BATCH_CONTROL table. When it is performing the Chunking Conflict Checking, the system uses Bulk Conflict Check LUW as the size of the chunk.

For this enhancement, the user can change the factor X above without having to go to the code. A new column, CONFLICT_CHECK_CHUNK_FACTOR, has been added to the RPM_SYSTEM_OPTIONS table. If this column is NOT NULL, the system uses the value in this column as the factor X. Otherwise, uses the default (2.5) for the factor X. This column must be updated from the backend by the DBA; there is no UI screen for this new column.

A new entry, "com.retek.rpm.app.bulkcc.service.ChunkConflictCheckAppService," has been added in the RPM_BATCH_CONTROL table. The THREAD_LUW_COUNT column of this entry is used as the size of the chunk. If it is not available in the table, the system uses the "Bulk Conflict Check LUW" as the size of the chunk. The client must set the THREAD_LUW_COUNT of this new entry accordingly.

With this new setup, the formula used to determine if chunk conflict checking is used or not has been updated to the following:

Number of potential Item/Location in the price event > X * Chunk Conflict Check LUW.

Documentation Correction

In addition to updates to content to address the release, the following corrections were made to the *Oracle Retail Price Management Operations Guide*.

RIB Subscribe Messages

The Oracle Retail Merchandising Sytem (RMS) does not notify RPM through Oracle Retail Integration Bus (RIB) messages. In other words, RPM does not subscribe to RIB messages; rather, RPM only publishes RIB messages.

Until the 13.1.3.1 release, the RPM Operations Guide contained text and tables explaining how RPM utilized RIB subscribe messages. This information is not valid and has since been removed from Chapter 5, "Integration Methods and Communication Flow."

Further, the "XML Format Message Format" diagram in the section called "RPM and the Oracle Retail Integration Bus (RIB)" has been updated to indicate only a flow from RPM through RIB to another Oracle Retail application (not in the reverse order).

Defect Fixes and Documentation

A defect fix is a modification to the base Oracle Retail code (for example, a bug fix, a performance enhancement, or a functional enhancement). Each defect fix that is included in this patch has a corresponding defect report titled <*defect-number*>.PDF (for example, 1234567.PDF).

In the same folder, the file named DEFECT MODULE XREF RPM 13.1.3.1.XLS lists every defect number and the modules and scripts that are included in the patch. Review each defect report carefully before implementing the defect fixes. Please note that scripts do not preserve data. Make sure that all data is backed up before you run any script.

Noteworthy Defect Fixes

The following defect fixes and enhancements are included in RPM 13.1.3.1. This is not a complete list; see the cross-reference spreadsheet and defect reports for a complete list and full details.

Defect Number	Summary
9690281	Enable the user to create Price Changes, Clearances, Simple Promotions, and Threshold Promotions at the ItemList level.
9880447	When an item on a promotion is deleted from RMS, the Item Location Delete batch removes data only from the RPM_FUTURE_ RETAIL table and not from other RPM tables (such as clearance and promotions). This causes a fatal exception when attempting to maintain or copy the promotion.

Defect Number	Summary
10071525	When applying a price event at the department/zone level with a fixed price discount, the application stalls at unique unit of measure (UOM) validation.
10083377	Add a new system option (Do not run Conflict Check for Submit) to enable users to skip the Conflict Checking process during the Submit process. If this system option is checked, the system skips the conflict checking process when Submit is clicked.
10083409	Add a new system option to dictate whether conflict checking is executed for complex promotions (such as Buy/Get promotions).

Related Documentation

For more information, see the following documents in the Oracle Retail Price Management Release 13.1.3.1 documentation set:

- *Oracle Retail Merchandising Batch Schedule*
- *Oracle Retail Price Management Data Model*
- *Oracle Retail Price Management Operations Guide*
- *Oracle Retail Price Management User Guide*

Note: The *Oracle Retail Price Management Online Help* for this bundled hot fix release is in progress.

Installing the Bundled Hot Fix Release

1. Shut down the server and connected clients.
2. Download the .ZIP file from My Oracle Support.
3. Unzip the .ZIP file into a folder, such as RPM_HOTFIX.
4. There will be another .ZIP file called RPMxxApplication.zip (where xx is the version number, such as 11, 12, and 13).
5. Unzip RPMxxApplication.zip. Inside the file is RPM.ear.
6. Copy the RPM.ear file to the destination directory in your application server.
7. Go to the DBSource folder inside RPM_HOTFIX to find the changed .SQL scripts and .PLS files.
8. Open a SQL*Plus session as an RPM database user and compile all the .PLS files and execute .SQL files.
9. Restart the application server.