Retek® Merchandising System™ 10.2

Release Notes

Corporate Headquarters:

Retek Inc.
Retek on the Mall
950 Nicollet Mall
Minneapolis, MN 55403
USA

888.61.RETEK (toll free US)
Switchboard:
+1 612 587 5000

Fax:
+1 612 587 5100

European Headquarters:

Retek
110 Wigmore Street
London
W1U 3RW
United Kingdom

Switchboard:
+44 (0)20 7563 4600

Sales Enquiries:
+44 (0)20 7563 46 46

Fax:
+44 (0)20 7563 46 10

Customer Support

Customer Support hours

Customer Support is available 7x24x365 via email, phone, and Web access.

Depending on the Support option chosen by a particular client (Standard, Plus, or Premium), the times that certain services are delivered may be restricted. Severity 1 (Critical) issues are addressed on a 7x24 basis and receive continuous attention until resolved, for all clients on active maintenance. Retek customers on active maintenance agreements may contact a global Customer Support representative in accordance with contract terms in one of the following ways.

Contact Method	Contact Information
E-mail	support@retek.com
Internet (ROCS)	rocs.retek.com Retek's secure client Web site to update and view issues
Phone	+1 612 587 5800

Toll free alternatives are also available in various regions of the world:

Australia	+1 800 555 923 (AU-Telstra) or +1 800 000 562 (AU-Optus)
France	0800 90 91 66
United Kingdom	0800 917 2863
United States	+1 800 61 RETEK or 800 617 3835
Mail	Retek Customer Support Retek on the Mall 950 Nicollet Mall Minneapolis, MN 55403

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.

Release Notes

Overview

Please review the enclosed DEFECT documents promptly to establish the impact on your business operations. Retek Customer Support investigates submitted issues with the assumption that all release patches have been applied. While it is ultimately at the client's discretion as to when to apply patches, delays or lags in their application can complicate the support process. To assist in the patch review, Retek Customer Support provides a system-level assessment by assigning a Priority. In addition, a cross-reference spreadsheet is provided to assist with this research (see 'DEFECT Documentation' below).

As listed below, this patch contains both general product fixes and functional enhancements and is considered the RMS 10.2 release. Refer to the 'RMS Fixes' and 'RMS Enhancements' sections of this document for more details.

10.1.7 Fixes – Oracle 9i Forms Upgrade; Sales Audit; Item Maintenance; Organizational Hierarchy; Invoice Matching; Cost Management; Purchase Orders; Transfers; Receiving.

Enhancements – Ability to run batch programs with online users; Integration with Retek Master Data Management (MDM).

Before installing RMS 10.2, confirm that RMS 10.1 and all following patches (RMS 10.1.1, RMS 10.1.2, RMS 10.1.3, RMS 10.1.4, RMS 10.1.5, RMS 10.1.6) have been applied.

The 10.2 patch release contains batch, package, forms program modifications. Refer to the 10.2 patch documentation (located in the doc folder on this CD) for detailed information on each fix.

As with all patches, the following points should be considered before applying RMS 10.2:

- Copy the original files to an archive directory before you overwrite them in case they are later needed for reference.
- Note that the cutoff date for RMS 10.2 was February 20, 2004. Fixes that you've received after this date and applied to your environment may require special consideration when applying this patch.
- Note whether customizations have been made to the module. If so, the customizations will need to be reapplied to the new version of the module (or the fix may need to be applied to the custom version of the code).

DEFECT Documentation

A DEFECT fix is a modification to the base Retek code (e.g. a bug fix, performance enhancement, or functional enhancement). Each DEFECT fix included in this patch has a corresponding DEFECT document in the \doc\defect_doc folder titled <DEFECT#> <module>.doc (e.g. "123456 nxprcno.doc"). DEFECT documents should be fully reviewed before this patch is implemented.

To assist with the patch application process, there is also a DEFECT module cross-reference spreadsheet in the same folder (DEFECT MODULE XREF 1016.xls) which lists and allows sorting by DEFECT, Program Name, Revision #, Functional Area, Priority, and Defect Description. The spreadsheet includes a full list of all the previous patch DEFECT modules, plus tabs showing DEFECTs related to the current patch, and current DEFECTs broken out by module types.

RMS Fixes

As mentioned earlier, RMS 10.2 includes general RMS 10.1.7 product fixes. Refer to the 'RMS 10.2 Patch' tab in the DEFECT module cross-reference spreadsheet (DEFECT MODULE XREF 1016.xls) for DEFECT, module, functional area, priority, and description. This is a complete list of fixes included in the patch. Please read the provided documents for details (see 'Defect Documentation'). Description of several noteworthy fixes follows.

9i Oracle Forms Upgrade

Oracle Forms 9i is now supported with RMS 10.2. An upgrade is not mandatory with RMS 10.1.7 and Forms 6i will continue to be supported until further notice. Retek advises waiting until the release of Oracle's 10G application server (9i) Forms/Reports (mid-April 2004).

New clients, without prior builds on RMS 10.1.x, on Forms 9i can apply all preceding patches on Forms 9i with the expectation that there will be errors with the following forms until RMS 10.1.7 is applied.

Existing clients with builds up to the RMS 10.1.6 patch, that choose to upgrade to Forms 9i, can first upgrade forms and then apply the 10.1.7 patch. Fixes specific to the upgrade are listed below.

- **Defect 360343** – The URL is appended with 'forms90' next to the server and listener port for reports run on 9i and a 'Page cannot be displayed' error occurs.
- **Defect 360307** – The itemfind form has some text fields bottoms cut off when upgrading to Forms 9i.
- **Defect 360086** – The rtkstrt form doesn't populate the right hand side menu when running RMS with Oracle 9i forms builder.
- **Defect 360274** – User receives ORA-06502: PL/SQL: numeric or value error when entering from the Item search window and when reentering the form from another window.
- **Defect 360242** – Compilation error when compiling itemsupp.fmb on the Oracle 9i forms builder: default value of parameter "I_CALLING_FORM" in body must match that of spec.

Advance Fix Notifications

Over the last quarter Advance Fix Notifications were sent for the following defects. Both of these fixes should be applied immediately after the 10.1.6 patch:

Defect 358870 - When creating a new purchase order, an error is displayed stating, "Order Number is invalid.

Defect 359074 - Control script c357602.sql was excluded from Defect 357602.

Sales Audit

Defect 357736 – ReSA did not handle rounding due to the denominations in different markets. This rounding issue can best be explained by an example:

> The customer buys a set of items totaling 2064 KRW. The smallest payable amount is 10 KRW. The customer therefore pays 2060 KRW. ReSA does not accept that the Total Tendered does not match the Total Sales. Modifications for this Defect are to support the above requirement.

Refer to the Sales Audit appendix for an overview of two new forms: Currency Rounding Rules Header Maintenance [sarrulhd] and Currency Rounding Rules Detail Maintenance [sarruldt].

Trade Management

Defect 363973 – Updated RTM Control Scripts for 2003/2004 data installation: hts_headings.sql; oga.sql; tariff_treatment.sql.

Stock Ledger

Defect 361690 – Salweek's c_two_half_store cursor only doing order by store but while doing a comparison it is expecting the half_no to be in ascending order. Under certain conditions the program is fails with the following error message:

> salweek_11~20040207172254~get_location_info~half_data_budget~1403~-1403: non-ORACLE exception ~Invalid dept [2001], store [0000000976], half [20041]

Defect 361818 – Salmth calculates the opening and closing stock cost and retail incorrectly. When no transaction has been taken place for a store, then the program ends with opn_stk_retail/cost and cls_stk_retail/cost as zero instead of its previous value in the MONTH_DATA table.

Transfers

Defect 360196 – When a transfer is manually closed online, the stock on hand is not handled ideally for items shipped but not received. The inventory is place back at the shipping location. This defect is an enhancement to handle those quantities differently.

Defect 360197 – Stock orders with the status of Expired are setting transfers and allocations to Close incorrectly.

Item Maintenance

Defect 353808 – The 10.1.5 patch added a new item form for quick item entry. The new item form, itemadd, has a combination of features from the forms itemmaster, itemsupp, and itemsuppctry. It does not have all of the features of these forms, but has enough to create and approve an item without accessing any other forms. This is designed for a basic item that will only come from one supplier and origin country, and will only have one set of dimensions. See Quick Item Entry appendix for additional details. All supporting documentation for this fix was included with 10.1.5.

Miscellaneous

Defect 353841 - The 10.1.6 patch excluded package invattrb.pls v1.3 from Defect 353841. Patch 10.1.7 Defects 357717 and 358740 include the invattrb.pls code (v1.4 and v1.5). All supporting documentation for this fix was included with 10.1.6.

RMS Enhancements

As mentioned earlier, the RMS 10.2 patch release consists of two primary functional enhancements in addition to the general RMS 10.1.7 product fixes. These functional enhancement areas include:

- Batch with Online Users
- Integration with Master Data Management (MDM).

Batch with Online Users

The RMS 10.2 release enables retailers to run selected batch programs with online users. This functionality is critical for retailers that run in a 24x7 and/or global environment. Prior to RMS 10.1.7, record locking would result in the hang-up or failure of the batch programs for such clients.

Batch with Online Users functionality is provided in addition to the general product fixes and MDM integration included in the RMS 10.1.7 release. Batch programs were evaluated for inclusion in the 10.2 functionality and a selected number are included in the 10.2 release on March 31, 2004. The remaining prioritized programs will be included in a future 10.2.x release.

The selected 10.2 batch programs have been modified to account for record locking issues that arise from both online users and batch processes accessing the same tables and/or data at the same time. Now when a lock occurs a soft error message / error table is written. This allows the batch program to proceed and to later re-process the locked records before continuing on with the batch schedule. In addition, with several programs selected records are now deleted during processing rather than being deleted in the post processing programs. This eliminates the need for certain post programs. This modification ensures records created and added by users online during the batch process will not be removed prior to being processed. This functionality does not address trickle polling / continuous batch processing, nor does it eliminate the need for a minimal-user batch window.

The Batch with Online Users appendix provides a functional overview of the Lock Checking and General Security approaches. The batch programs impacted by this modification are detailed in the Batch Designs (see the RMS 10.2 Operations Guide addendum).

Integration with Master Data Management (MDM)

Retek's Master Data Management (MDM) provides seamless integration from MDM 10.0 to RMS 10.2. All item data created in MDM is published to RMS.

Through MDM Retek now offers an enhanced and more robust user experience for auto item induction and item management in MDM while maintaining the strong integration of items within Retek's solution suite through RMS. For customers implementing MDM, it is assumed that item induction and management will be performed in MDM because of the additional flexibility, usability, and features. The items created and updated in MDM are sent to RMS real-time for use anywhere in RMS and other integrated applications as items are currently.

Interfaces were created for the following functional areas: Suppliers, Locations, Items, Seasons and Phases, and Seed Data. Given assumption that MDM is the single point of entry for item management, RMS has been changed to remove access to the item dialog for most information that is entered in MDM. Access has moved from the RMS item dialog to another folder for two functional areas:

- Replenishment setup (now accessible for the RMS Inventory folder);
- Unavailable Inventory (now accessible for the RMS Inventory folder).

The 'RMS/MDM New Integration Point Summary' in the Integration with Master Data Management appendix provides the specific functional areas that will remain in RMS and where the new access point is from either in MDM or RMS. The appendix provides also an overview of the MDM interfaces with RMS by functional area.

Batch With On-line Users – General Security

Functional Area Overview

Batch with on-line Users – General Security Approach

Security features which are added to RMS are maintained in the batch cycle. With each run, the changes made to the data in RMS will be brought under the security features of RMS through the running of 3 batch programs.

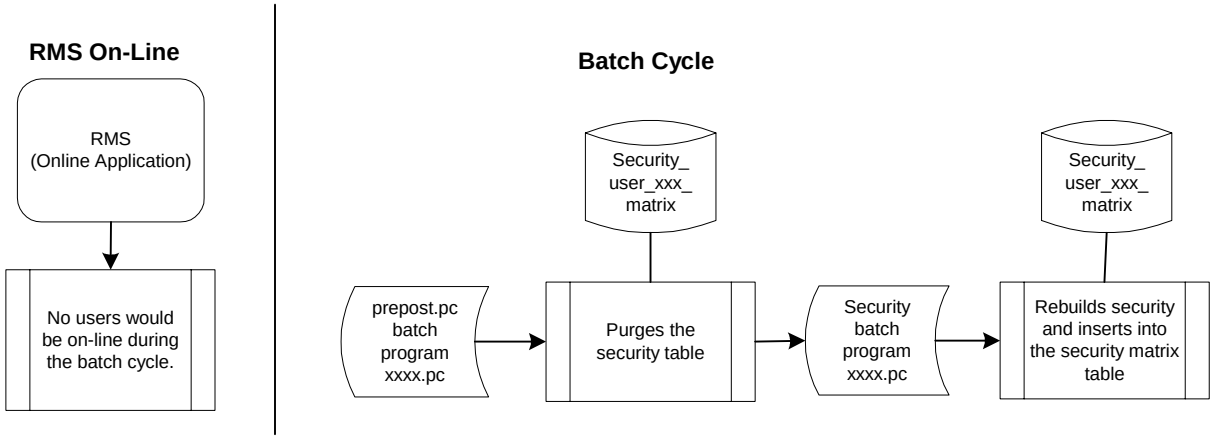
Clients require the ability to run their batch cycle while users are still on the system in order to account for scenarios where there are 24 hours stores or when the client has locations throughout the world within in different time zones.

The general security programs need to be modified to work with the Batch with on-line Users option. The current general security maintenance process is to truncate the security tables then rebuild them. However, with the Batch with on-line Users option it is no longer acceptable to purge the security information from the system, leaving on-line users without security.

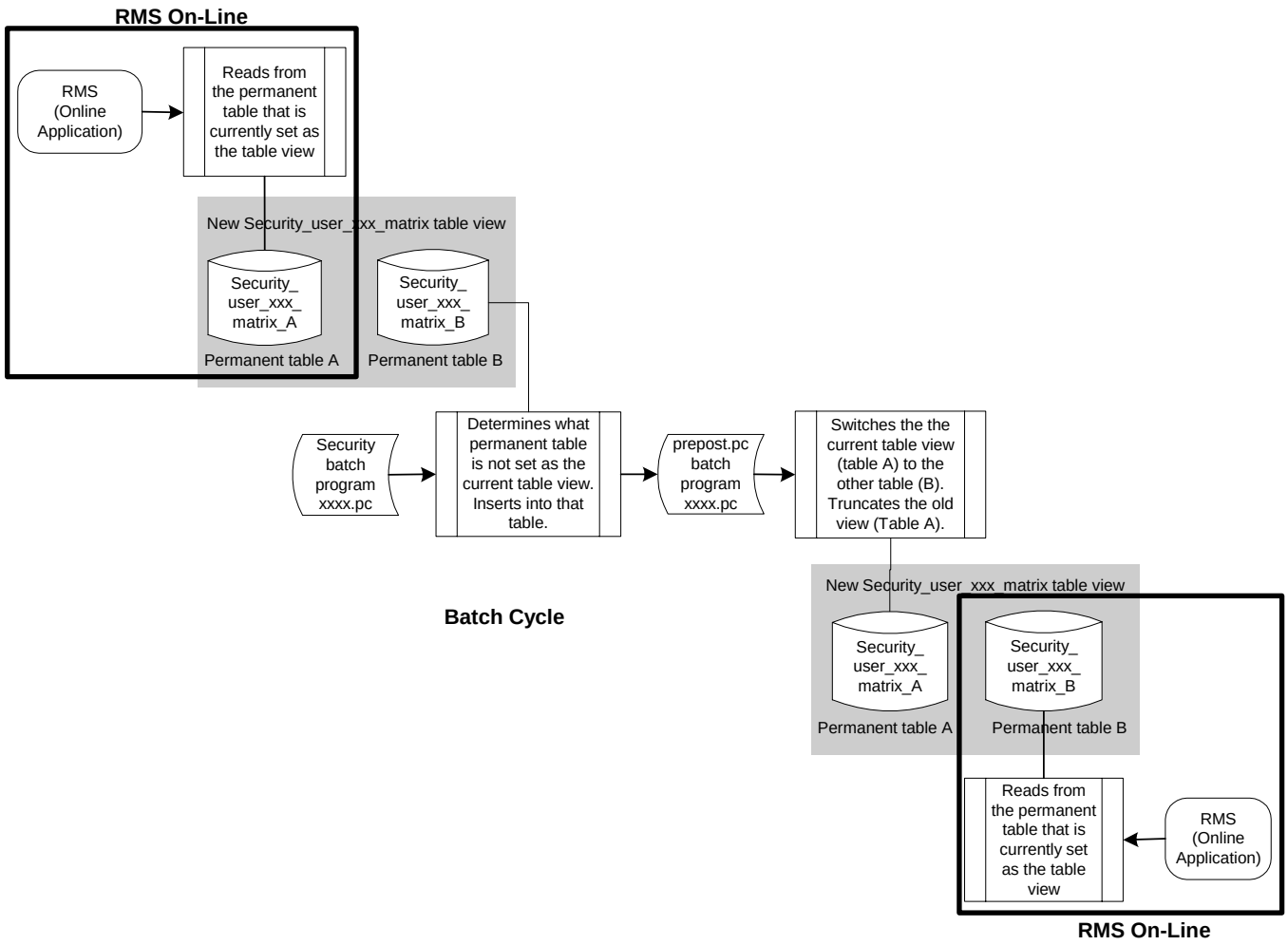
To prevent the above scenario, the system will be modified to create security table views. Within each security view there will be two permanent tables. While the batch is rebuilding the security the on-line application will point to the permanent table with the old security. Once the new security has been built a batch program will switch the on-line application view to the permanent table with the new security.

Functional Area Flow

Current Functional Flow:



system_options.btch_w_user_ind = 'Y':



Detailed Flow

The system_option table will be modified to introduce a new indicator named batch_w_usr_ind. If this indicator is set to 'Y' then system is set for Batch with on-line Users. All the batch programs check if the btch_w_usr_ind is 'Y'es or 'N'o before before rebuilding the security.

The sec_user_xxx_matrix tables will also be changed to be table views and two new permanent tables will be created in place (sec_user_xxx_matrix _a and sec_user_xxx_matrix_b). *A* new function will be called to find what permanent table is set as the current table view (for example Table A). The program will then insert into the table which currently is NOT set as the current view (for example: Table B). Once all the security has been rebuilt, prepost.pc will be modified to create the new table view (switch the view from Table A to Table B) and truncate the old table (Table A).

If btch_w_usr_ind = N

Prepost.pc - pre

- If the btch_w_usr_ind = 'N' call a new function to find what permanent table (A or B) is set as the current table view.
- Truncate the table which is returned as the current view.

Security program

- Add new function call to find what permanent table (A or B) is set as the current table view.
- Modify the insert statements to insert into the table which is set as the current view.

Prepost.pc - post

- Skip the table switch and truncate logic

If btch_w_usr_ind = Y

Prepost.pc – pre

- If the btch_w_usr_ind = 'Y' skip the truncating logic

Security program

- Add new function call to find what permanent table (A or B) is set as the current table view.
- Modify the insert statements to check what permanent table is the current view, if it is table A then the insert statement should insert into sec_user_zone_matrix_b, if the current view is Table B then insert into sec_user_zone_matrix_a (based on what was found as the current view).

Prepost.pc - post

- Call the new function to find what permanent table (A or B) is set as the current table view.
- Call new function to switch the table view from the table which was found as the current view to the other table (A or B).
- Call the existing truncate table function to truncate the table which was returned as the current view.

Screen Layout

No changes

Scheduling Considerations

No changes

Concurrency Considerations

- Error messages could arise when the post security functions run and switch the table views to the new security table if an on-line user is accessing something which they no longer have security for. No data will be corrupted, however error messages may force the user to log out of the system.

Security Considerations

- To keep on-line user security intact, the Batch with on-line Users option will require that the batch cycle runs with all the Oracle policies turned on. Therefore the prepost.pc pre batch function will be modified to only disable Oracle policies if the btch_w_usr_ind = 'N'. There will be performance impacts when the btch_w_usr_ind = 'Y'.

Performance and Volume Considerations

- The Batch with on-line Users option will require that the batch cycle runs with all the Oracle policies turned on. Therefore the prepost.pc pre batch function will be modified to only disable Oracle policies if the btch_w_usr_ind = 'N'. There will be performance impacts when the btch_w_usr_ind = 'Y'.

Design Assumptions

- When the Batch with on-line User Option is on, the batch user which will be used to run the batch cycle will have to have **full privileges** since the Oracle Policies will remain enabled.
- Trickle processing is not in the scope. Batch programs which have been modified for the Batch with on-line option will not run in trickle-processing mode, they will still run in a single night batch run.
- Reclsdly.pc runs before the security processing. When the Batch with on-line User Option is on, during the reclassification process it is possible that an on-line user might lose security due to an item being reclassified.
- Error messages could arise when the post security functions run and switch the table views to the new security table if an on-line user is accessing something which they no longer have security for. No data will be corrupted, however error messages may force the user to log out of the system.

Appendix

- All defect docs for Defect 360642
- Functional Design for RMS 10.2 - Batch With On-line Users – Check Locks

Batch With On-line Users – Lock Checks

Functional Area Overview

Batch with on-line Users – Lock Checking Approach

Clients require the ability to run their batch cycle while users are still on the system in order to account for scenarios where there are 24 hours stores or when the client has locations throughout the world within in different time zones. The current state of RMS doesn't allow not allow the batch cycle to run while on-line application is running. If an on-line process acquires data and at the same time batches are kicked off and trying to process same data, locking contention cause batches to wait for lock to be released.

RMS 10.2 requires the ability to run batch programs at the same time that on-line users will be in the system. There will still be a dedicated batch window to run the programs, but users will no longer be required to be locked out of the on-line system.

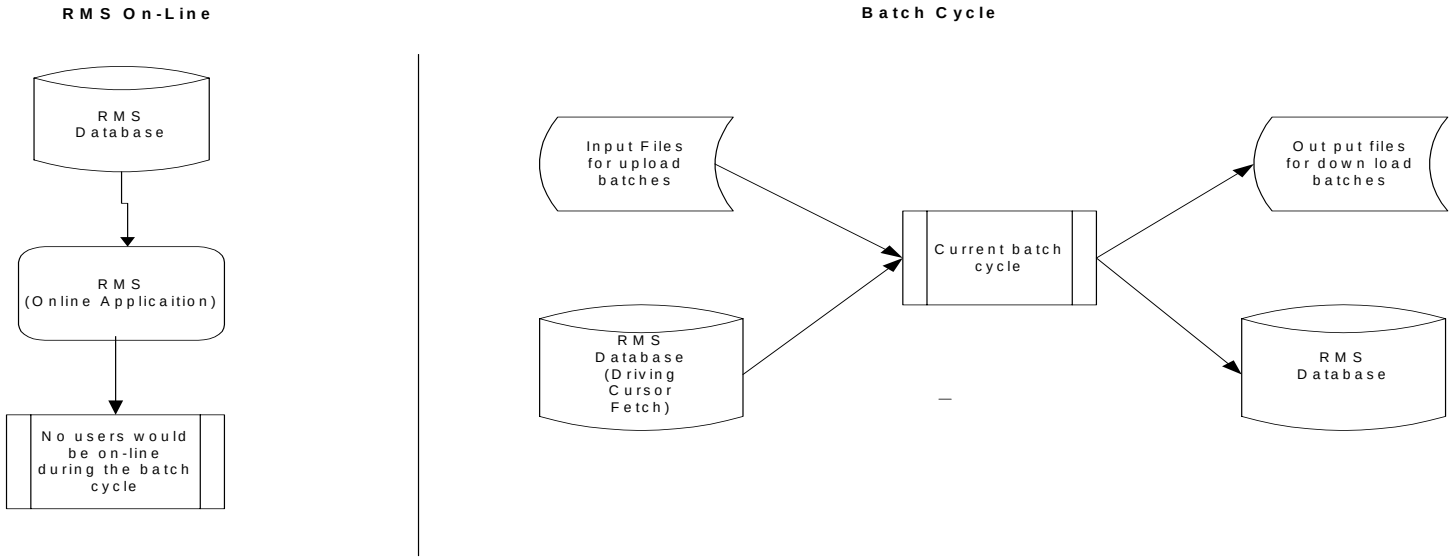
If locked records are encountered the program will terminate "successfully with record locking errors". A fatal error code (255) will be passed out of the program in order to stop the batch schedule since the locked records will have to be reprocessed before the batch schedule can continue. If locked records are encountered during a program it is required that the batch schedule will stop so that the locked records can be reprocessed.

The system_option table will be modified to introduce a new indicator named as batch_w_usr_ind. If this indicator is set to 'Y' then system is set for Batch with on-line Users. All the batch programs check if the btch_w_usr_ind is 'Y'es or 'N'o before running the new check for the lock functionality.

Note, this functionality does not provide clients the ability to run batch programs throughout the day in a trickle process mode, but instead will provide the client the ability to run a single daily batch window while users are using the on-line system.

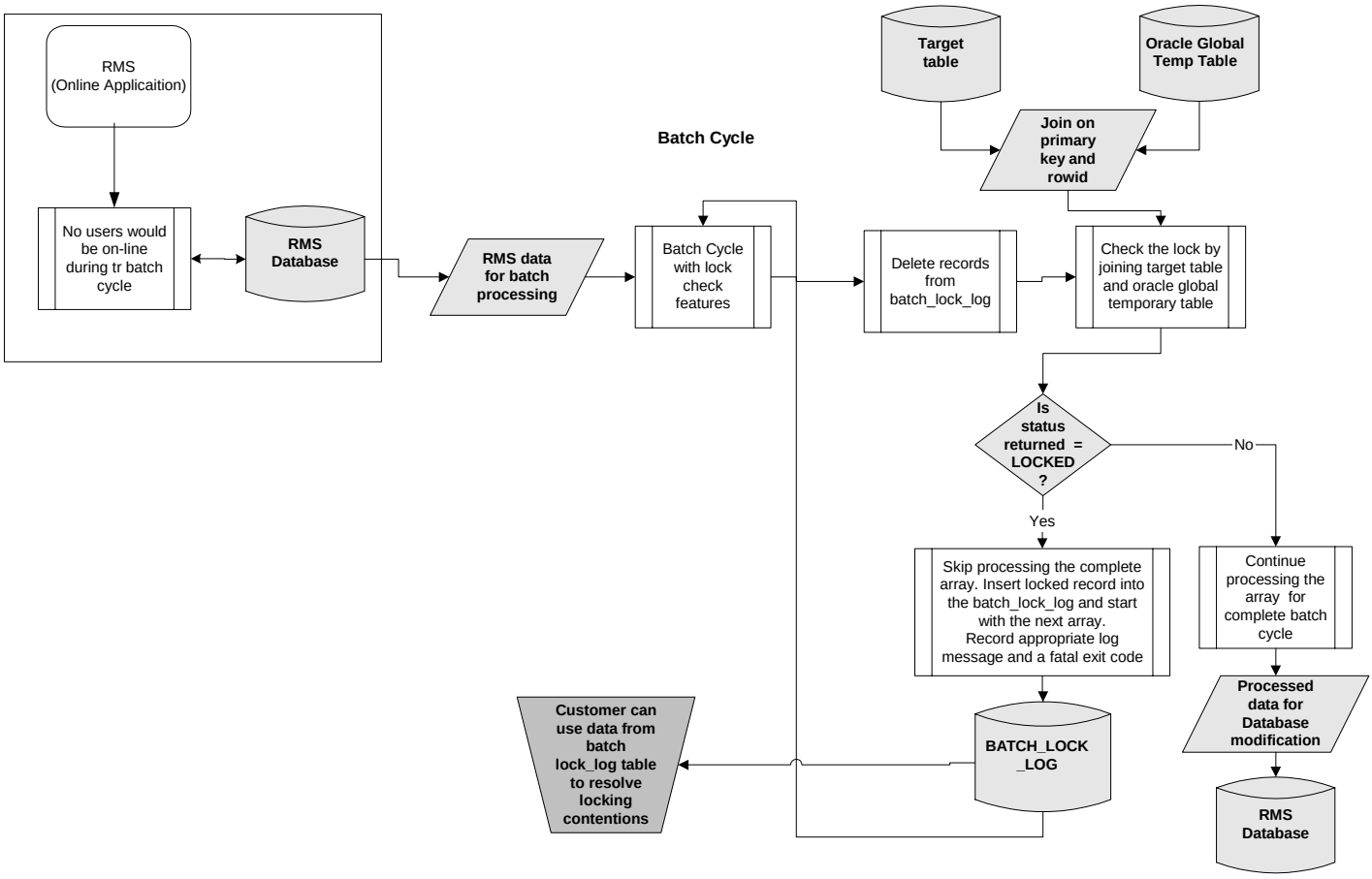
Functional Area Flow

System_options.btch_w_usr_ind = ‘N’:

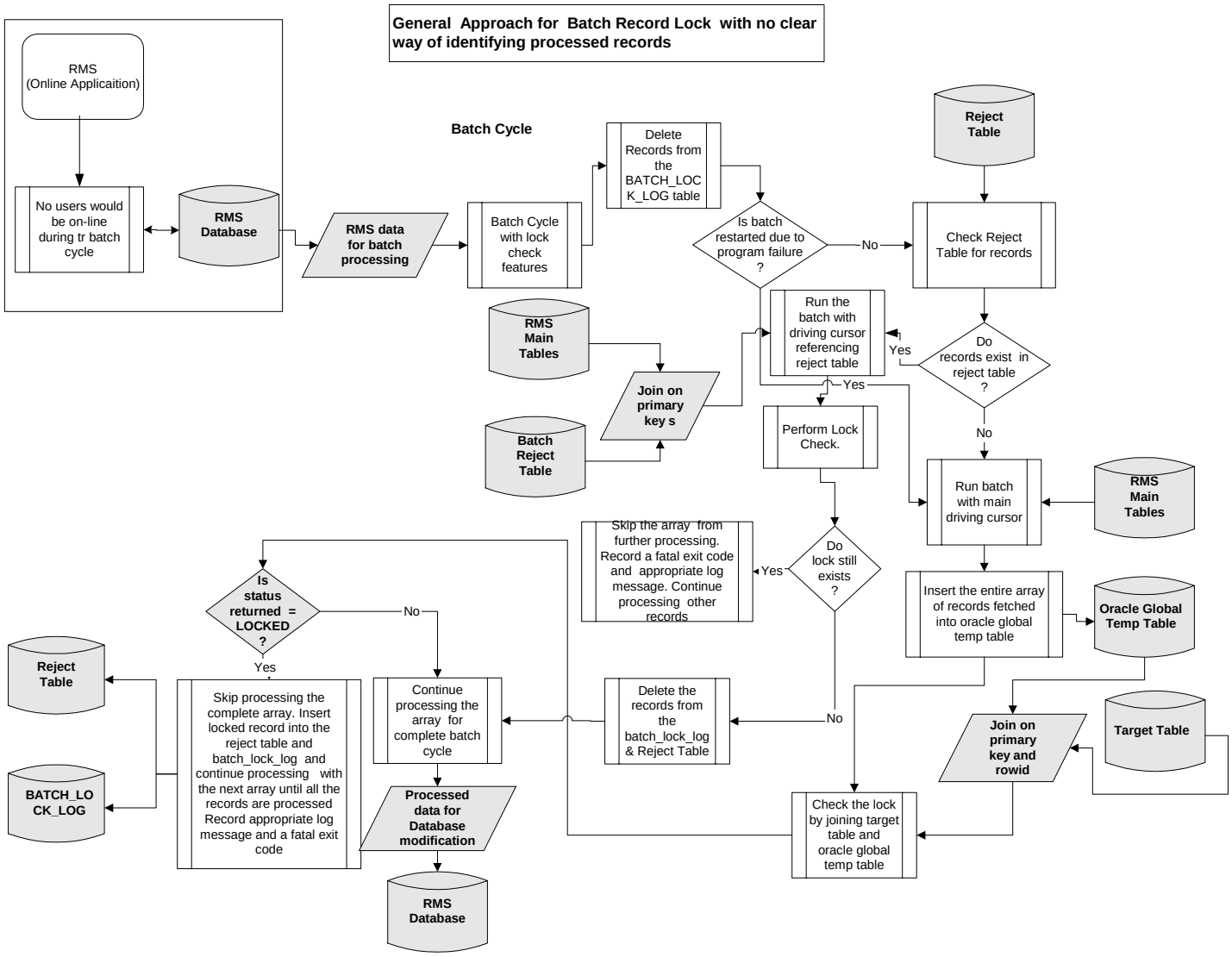


System_options.btch_w_usr_ind = ‘Y’ and programs which have a clear way to determine which records have already been processed (driving cursor does not reselect all information):

General Approach for Batch Record Lock for Programs that rely on application status (Clear way of knowing processed records)



System_options.btch_w_usr_ind = 'Y' and programs which do NOT have a clear way to determine which records have already been processed (driving cursor reselects all information):


General Approach for Batch Record Lock with no clear way of identifying processed records
RMS (Online Applicaition)
No users would be on-line during tr batch cycle
RMS Database
Batch Cycle
RMS data for batch processing
Batch Cycle with lock check features
Delete Records from the BATCH_LOCK_LOG table
Is batch restarted due to program failure ?
No
Reject Table
Check Reject Table for records
Run the batch with driving cursor referencing reject table
Yes
Do records exist in reject table ?
No
Yes
RMS Main Tables
Join on primary key s
Batch Reject Table
Perform Lock Check.
Run batch with main driving cursor
RMS Main Tables
Skip the array from further processing. Record a fatal exit code and appropriate log message. Continue processing other records
Yes
Do lock still exists ?
Insert the entire array of records fetched into oracle global temp table
Oracle Global Temp Table
Is status returned = LOCKED ?
No
Yes
Reject Table
Skip processing the complete array. Insert locked record into the reject table and batch_lock_log and continue processing with the next array until all the records are processed Record appropriate log message and a fatal exit code
Continue processing the array for complete batch cycle
Delete the records from the batch_lock_log & Reject Table
No
Join on primary key and rowid
Target Table
BATCH_LOCK_LOG
Processed data for Database modification
Check the lock by joining target table and oracle global temp table
RMS Database

Detailed Flow

The system_option table will be modified to introduce a new indicator named batch_w_usr_ind. If this indicator is set to 'Y' then system is set for Batch with on-line Users. All the batch programs check if the btch_w_usr_ind is 'Y'es or 'N'o before running the new check for the lock functionality.

The batch modules that are trying to update/delete need to be modified in order to run Batch with on-line Users. In order to achieve Batch with On-line user functionality, bulk record locking strategy will be deployed to help improve performance when the functionality of the batch program allows for it (i.e. typically used when the batch program uses array processing). For tables which are updated or deleted, a corresponding Oracle Global Temporary table will be created in the database. The Oracle Global Temporary table is just like normal table except it is located in database memory. These tables will be created with 'DELETE ON COMMIT' option. The moment a commit is issued these tables get emptied, in cases where a commit will not be reached the temp table will be deleted from In the main processing loop of the batch these tables will be joined with the target table with update for no wait clause. If any of the row in the array is locked the whole array will be rejected and program will proceed to next array available for processing. The rejected rows will be inserted into the BATCH_LOCK_LOG in order to help operations find which records are locked.

Programs which do have a clear way to determine which records have already been processed:

If the btch_w_usr_ind = Y

- If the program is not on a restart, delete all records from the batch_lock_log where program_name = batch name and thread_val = thread value (init() function)
- Insert record block size equals to the commit max counter into global oracle temporary table (if array processing is being used).
- ecord locking will be performed in bulk by joining this oracle global temp table with target table using update for no wait clause (otherwise row by row locking will occur if array processing is not currently being used by the batch program).
- If any of the records in the array are found to be locked
  - The first table found locked, and the associated Primary key values, rowids (if the rowids are already being fetched by the batch program), thread value will be inserted into the batch_lock_log table (see table description under the database section)
  - If an oracle global temp table is used, it should be deleted from since the commit logic will not be reached (cannot use a truncate statement since it will issue a commit which can cause issues with Restart Recovery or program processing no-logical units of work).
  - A fatal exit code should be returned at the end of the program and the whole record block is skipped from further processing.
    - Error Message: "Thread %s - Terminated Successfully with record locking errors."

        - Exit the program at the end of all the processing with a new exit code. A new macro will be created in the common.h library called: LOCKED_RECORDS. The exit value will be equal to 255 (the same as FAILED or REJECTED_RECORDS). This will ensure that the batch schedule will stop.
- If no record found to be locked then continue with further processing.
- If the btch_w_usr_ind = N
- Process normally as it does today.

Programs which do NOT have a clear way to determine which records have already been processed (driving cursor reselects all information):

If the btch_w_usr_ind = Y

- If the program is not on a restart, delete all records from the batch_lock_log where program_name = batch name and thread_val = thread value (init() function)
- Check if the program is on a restart
    - If it is a restart:
        - Need to determine if records locks have already been encountered, if so a non-fatal error should be returned at the end of the program.
            - Error Message: "Thread %s - Terminated Successfully with record locking errors."
            - Exit the program at the end of all the processing with a new exit code. A new macro will be created in the common.h library called: LOCKED_RECORDS. The exit value will be equal to 255 (the same as FAILED or REJECTED_RECORDS). This will ensure that the batch schedule will stop.
        - Continue processing the **main** driving cursor,
    - If it is not a restart:
        - Check if there are any records on the reject table,
            - if there are records on the reject table: Process the reject driving cursor
            - if there are NOT records on the reject table: Process the main driving cursor

Main Driving Cursor processing:

- Select Records from the main driving cursor (as it does today)
- Insert record block size equals to the commit max counter into global oracle temporary table (if array processing is being used).
- Record locking will be performed in bulk by joining this oracle global temp table with target table using update for no wait clause (otherwise row by row locking will occur if array processing is not currently being used by the batch program).
- If any of the records in the array are found to be locked
    - Insert records into a reject table for processing immediately after the batch program completes.

  - The first table found locked, and the associated Primary key values, rowids (if the rowids are already being fetched by the batch program), thread value will be inserted into the batch_lock_log table (see table description under the database section)
  - If an oracle global temp table is used, it should be deleted from since the commit logic will not be reached.
  - A fatal exit code should be returned at the end of the program and the whole record block is skipped from further processing.
    - Error Message: "Thread %s - Terminated Successfully with record locking errors."
    - Exit the program at the end of all the processing with a new exit code. A new macro will be created in the common.h library called: LOCKED_RECORDS. The exit value will be equal to 255 (the same as FAILED or REJECTED_RECORDS). This will ensure that the batch schedule will stop.
- If no record found to be locked then continue with further processing.

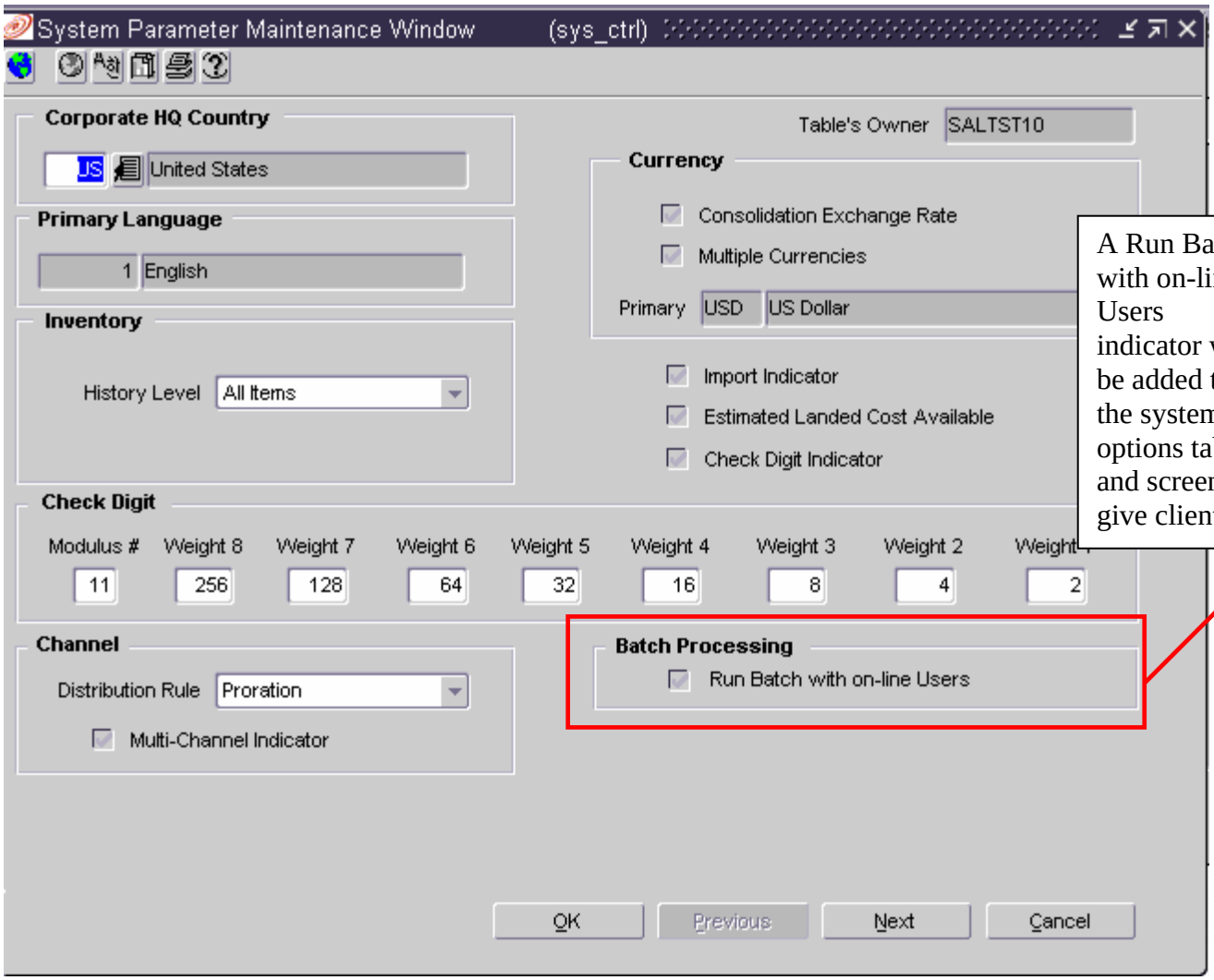
Reject Driving Cursor processing:

- Select records from the reject driving cursor (pulls the driving records only which exist on the reject table)
- Insert record block size equals to the commit max counter into global oracle temporary table (if array processing is being used).
- Record locking will be performed in bulk by joining this oracle global temp table with target table using update for no wait clause (otherwise row by row locking will occur if array processing is not currently being used by the batch program).
- If any of the records in the array are found to be locked
  - The first table found locked, and the associated Primary key values, rowids (if the rowids are already being fetched by the batch program) will be inserted into the batch_lock_log table (see table description under the database section)
  - If an oracle global temp table is used, it should be deleted from since the commit logic will not be reached.
  - A fatal exit code should be returned at the end of the program and the whole record block is skipped from further processing.
    - Error Message: "Thread %s - Terminated Successfully with record locking errors."
    - Exit the program at the end of all the processing with a new exit code. A new macro will be created in the common.h library called: LOCKED_RECORDS. The exit value will be equal to 255 (the same as FAILED or REJECTED_RECORDS). This will ensure that the batch schedule will stop.
- If no record found to be locked then continue with further processing from the reject table, delete all the records (for the current thread value) which were processed successfully before committing the records.

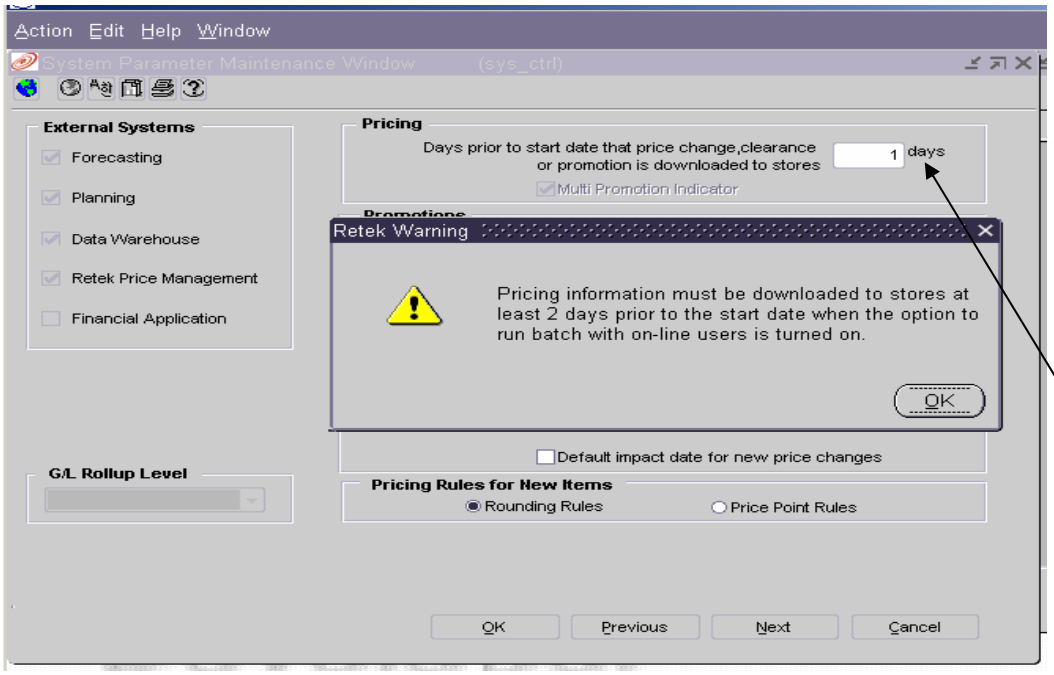
If the btch_w_usr_ind = N

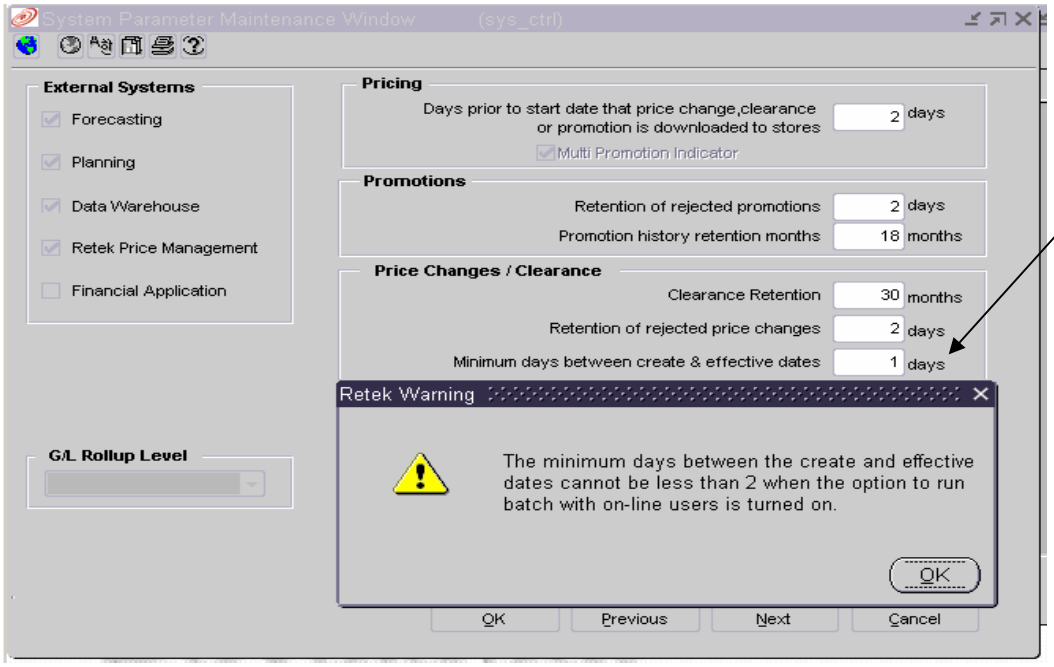
- Process normally as it does today.

Screen Layout

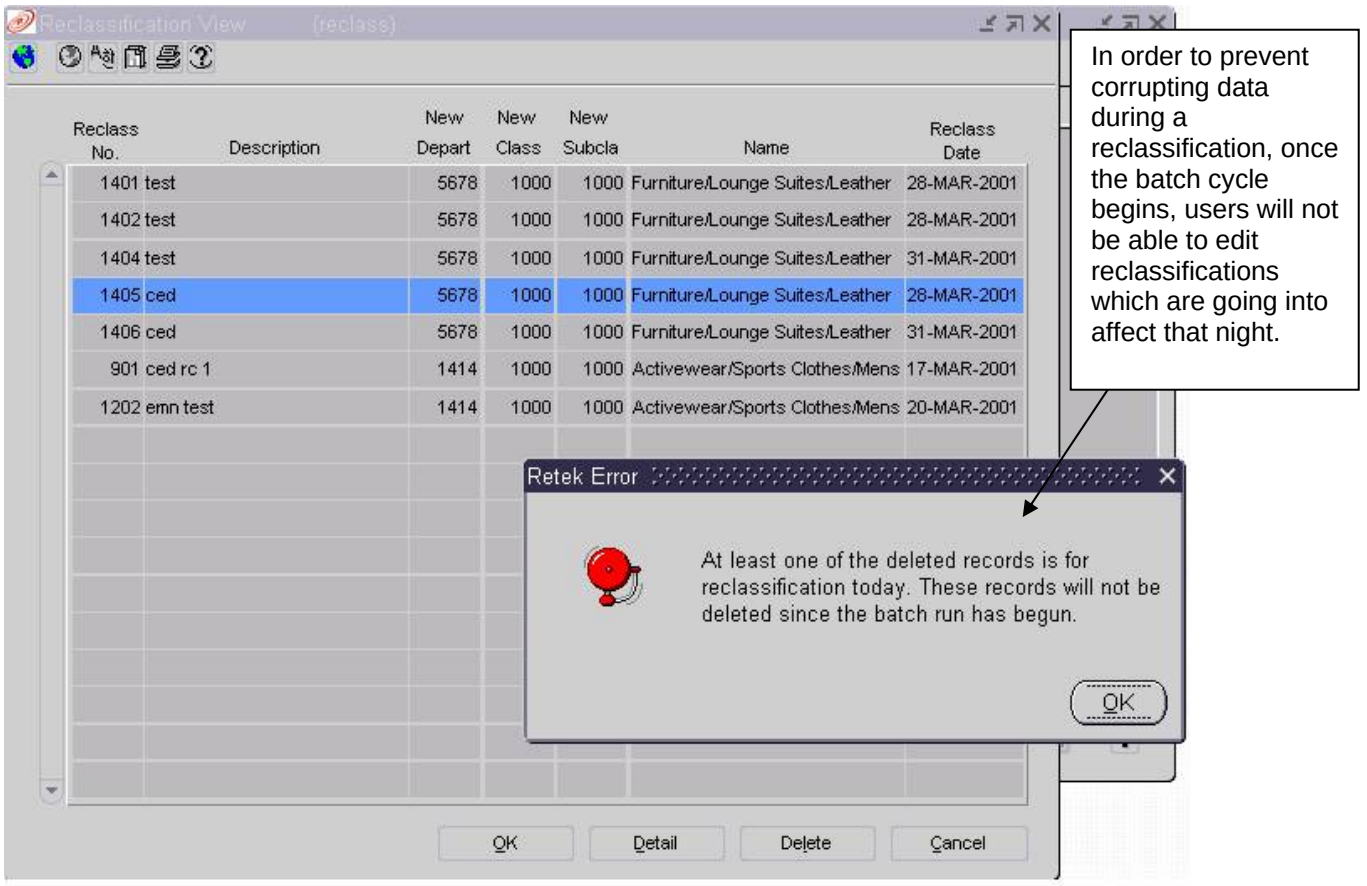


A Run Batch with on-line Users indicator will be added to the system options table and screen to give clients





In order prevent the risk of on-line users creating price changes, clearances or promotions on-line during the batch cycle which need to be processed the same night, the user will not be able to set the POS extract days less


Reclassification View (reclass)
Reclass No.
Description
New Depart
New Class
New Subcla
Name
Reclass Date
1401 test 5678 1000 1000 Furniture/Lounge Suites/Leather 28-MAR-2001
1402 test 5678 1000 1000 Furniture/Lounge Suites/Leather 28-MAR-2001
1404 test 5678 1000 1000 Furniture/Lounge Suites/Leather 31-MAR-2001
1405 ced 5678 1000 1000 Furniture/Lounge Suites/Leather 28-MAR-2001
1406 ced 5678 1000 1000 Furniture/Lounge Suites/Leather 31-MAR-2001
901 ced rc 1 1414 1000 1000 Activewear/Sports Clothes/Mens 17-MAR-2001
1202 emn test 1414 1000 1000 Activewear/Sports Clothes/Mens 20-MAR-2001
OK
Detail
Delete
Cancel
Retek Error
At least one of the deleted records is for reclassification today. These records will not be deleted since the batch run has begun.
OK
In order to prevent corrupting data during a reclassification, once the batch cycle begins, users will not be able to edit reclassifications which are going into affect that night.

Scheduling Considerations

The following schedule changes have been made:

New Schedule:

- pctrandn
- pctranex
- **postpctranex**
- pcext
- pccrext

New Schedule:

- pre precostcalc1
- pre precostcalc2
- costcalc
- post

New Schedule:

- prmxpld
- prmext
- **postprmext**
- pre prmpcupd post

New Schedule:

- supmth
- **postsupmth**

Concurrency Considerations

- If a batch program continues to encounter record locking errors due to an on-line user having the record locked, it may be necessary to have an operational person end their session in order to release the lock. This solution does not provide a programmatic process for doing ending locked sessions, some sort of user intervention will be needed.

Security Considerations

- To keep on-line user security intact, the Batch with on-line Users option will require that the batch cycle runs with all the Oracle policies turned on. Therefore the prepost.pc pre batch function will be modified to only disable Oracle policies if the btch_w_usr_ind = 'N'. There will be performance impacts when the btch_w_usr_ind = 'Y'.
- Please see the following document for more information on security: Functional Design for RMS 10.2 - Batch With On-line Users – General Security

Performance and Volume Considerations

- The Batch with on-line Users option will require that the batch cycle runs with all the Oracle policies turned on. There will be performance impacts with this approach.
- All the upload/download program currently performing row by row processing should be checked with row level locking (Please refer current version of posupld.pc) while all other programs with array processing should perform the lock check by bulk record locking via oracle global temporary table.
- Table locking, which occurs during the nightly batch cycle when the Batch with on-line Users option is turned on, will have performance impacts. For clients that have no need for the batch with on-line user option, there will be a system option setting to turn it off so that performance will not be impacted.

Design Assumptions

- In case of bulk record lock check, the entire logical unit of array that fails to acquire lock will be processed in a re-run of the batch program. It is **required** that all programs that completes successfully with record locks must be rerun until all the locked records are reprocessed before continuing the rest of the batch cycle.
- When the Batch with on-line User Option is on, the batch user which will be used to run the batch cycle will have to have full privileges since the Oracle Policies will remain enabled.
- In cases where pre/post functions are using truncate statements, the functions may need to be modified not to truncate the table to prevent record loss. In these cases, the truncate functionality will be replaced by a delete statement (for the records processed) inside the main program.
    - It was concluded no changes were needed to pcovrlp.pc or pre prcovl, since the price_conflict table which is being truncated during the pre process will be completely rebuilt during pcovrlp.pc. Therefore no records will be lost.
- Trickle processing is not in the scope. Batch programs which have been modified for the Batch with on-line option will not run in trickle-processing mode, they will still run in a single night batch run.
- For each batch program which does not have a clear way to determine which records have already been processed (i.e. driving cursor will reselect all the records), there will be a reject table specifically for that batch program.
    - No restart recovery will be needed when processing records from the reject table for those programs which do not have a clear way to determine which records have already been processed (i.e. driving cursor will reselect all the records).
- For all batch programs there will be one master batch_lock_log which the locked records will be written to. This will be used as one central place for researching and reporting record locks. In the case where a program continues to fail to acquire a lock on a specific table, this information could be used by a operational person to determine what session might need to be killed in order to release a lock.
    - Only the first table which encounters a lock will be written to the batch_lock_log.

- Programs which encounter record locks will exit with 'LOCKED_RECORDS' which will pass out an exit code 255. This value is the same as the program failure exit code and will **stop** the batch schedule so that the locked records can be reprocessed.
- In order prevent the risk of on-line users creating price changes, clearances or promotions on-line during the batch cycle which need to be processed the same night, the user will not be able to set the POS extract days less than 2 or the 'Min. days between create and effective dates' less than 2.
    - There are risks with editing the following two columns through the database rather than through RMS since there are no database constraint preventing a user to set the value less than 2 when the btch_w_usr_ind = 'Y': UNIT_OPTIONS. POS_EXTRACT_DAYS and UNIT_OPTIONS. PRICE_PRIOR_CREATE_DAYS
- In order to prevent corrupting data during a reclassification, once the batch cycle begins, users will not be able to edit reclassifications which are going into affect that night.
- Truncate statements cannot be used to remove records from the oracle temp tables since truncate statements performs a commit, which could cause bad data when processing non-logic units work or create issues with Restart Recovery.
- The Batch with on-line Users can be changed when the batch cycle is not running, however it is required that it is not changed once the batch cycle has started for the day.
- Programs which have non-logical units of work will reject the entire logical unit of work if one detail record is locked. For example, price changes. When processing price changes, if the batch program cannot lock all the detail records within the price change, no records will be processed for the specific price change.

EXAMPLE:
commit max counter = 2.

price chg	zone_group_id	zone_id	store	item
1	1	1	1	A
2	2	2	10	B
2	2	2	11	B
3	3	3	20	C
3	3	3	9	C
4	4	4	8	D
4	4	4	6	D
4	4	4	5	D
4	4	4	7	D
5	5	5	5	E

- If the first array is not locked, it will process the two records.
- If in the second array price chg 3 is locked, we will finish processing price chg 2, then reject price chg 3.
- In the 3rd array, we will reject the entire array since price chg 3 also exists there
- The 4th array will also be rejected since price chg 4 existed in the 3rd array (which was rejected).
- The 5th array should be rejected since price chg 4 also exists there and was rejected in the previous array.

Appendix

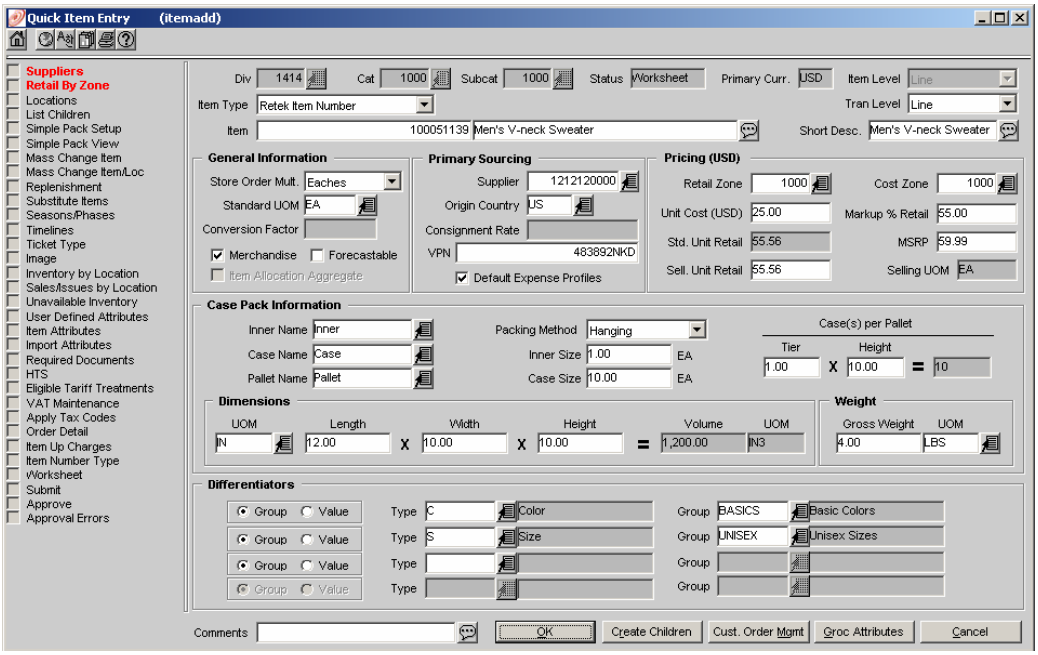
- All defect docs for Defect 360642
- Functional Design for RMS 10.2 - Batch With On-line Users – General Security

Quick Item Entry

Quick Item Entry (itemadd)

A new form was created to simplify the item creation process. This new Quick Item Entry form allows users to create a new, single level item, with one supplier and one origin country, in Approved status, using just one screen. Fashion Items with parents and children will require up to four screens at a minimum. The layout and functionality of the form is very similar to the Item Master form with a few additional fields from the Item Supplier and Item Supplier Country forms.

When creating a staple type of item with no reference items the user will only need to enter in one screen to create an approved item.



Difference between Quick Item Entry and Item Master

The user is allowed to create an item without being forced to go to the Item Supplier, Item Supplier Country, and Retail By Zone forms. The required fields from the Item Supplier and Item Supplier Origin Country forms have been added to the Quick Item Entry screen. With regards to the Retail by Zone form, the Quick Item Entry form assumes the values defaulted are acceptable. The processing is the same as if you went to the Retail by Zone form from Item Master and without making any changes pressed the OK button.

The creation of children items is performed exactly the same as it is via the standard Item dialog. The RCOM Attributes button and window were renamed to 'Cust. Order Mgmt.'. The 'Groc Attributes' button will only be displayed if the new System Option – Grocery Items Indicator is set to 'Yes'. The only other difference between Quick Item Entry and Item Master is that an Item can go directly from Worksheet to Approved Status in Quick Item Entry.

Primary Sourcing Information

- Primary Supplier (Item Supplier)
- Primary Origin Country (Item Supplier Country)
- Consignment Rate (Item Supplier)
- VPN (Item Supplier)
- Default Expenses Indicator (Item Supplier Country prompts user with a message on whether or not to default expenses)
- Unit Cost (Item Supplier Country)

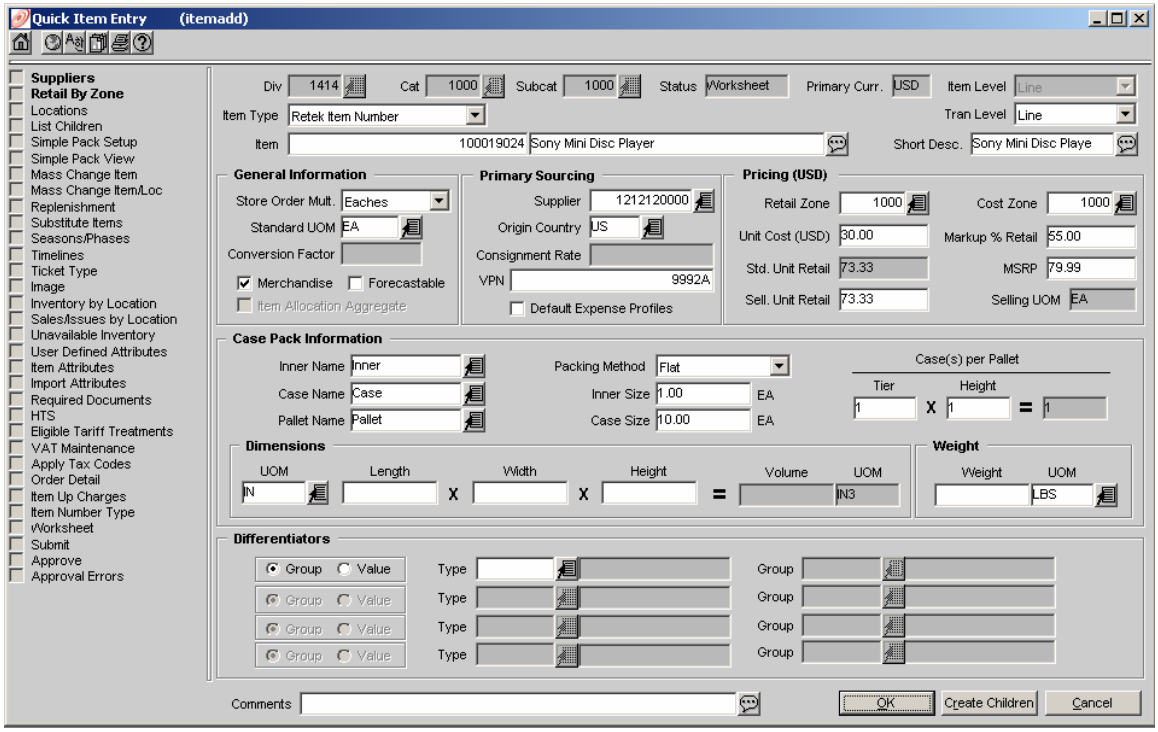
Case Pack Information

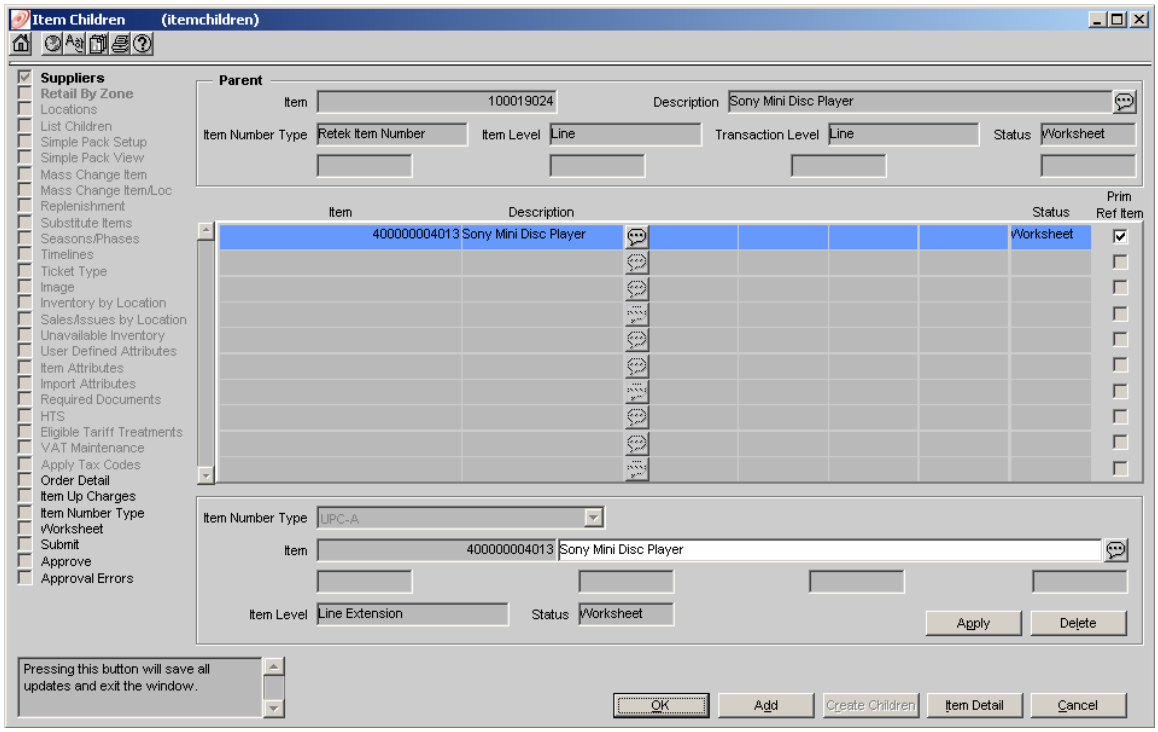
- Inner Name (Item Supplier Country)
- Case Name (Item Supplier Country)
- Pallet Name (Item Supplier Country)
- Packing Method (Item Supplier Country)
- Inner Size (Item Supplier Country)
- Case Size (Item Supplier Country)
- Tier (TI-HI) (Item Supplier Country)
- Height (TI-HI)

Dimension Information

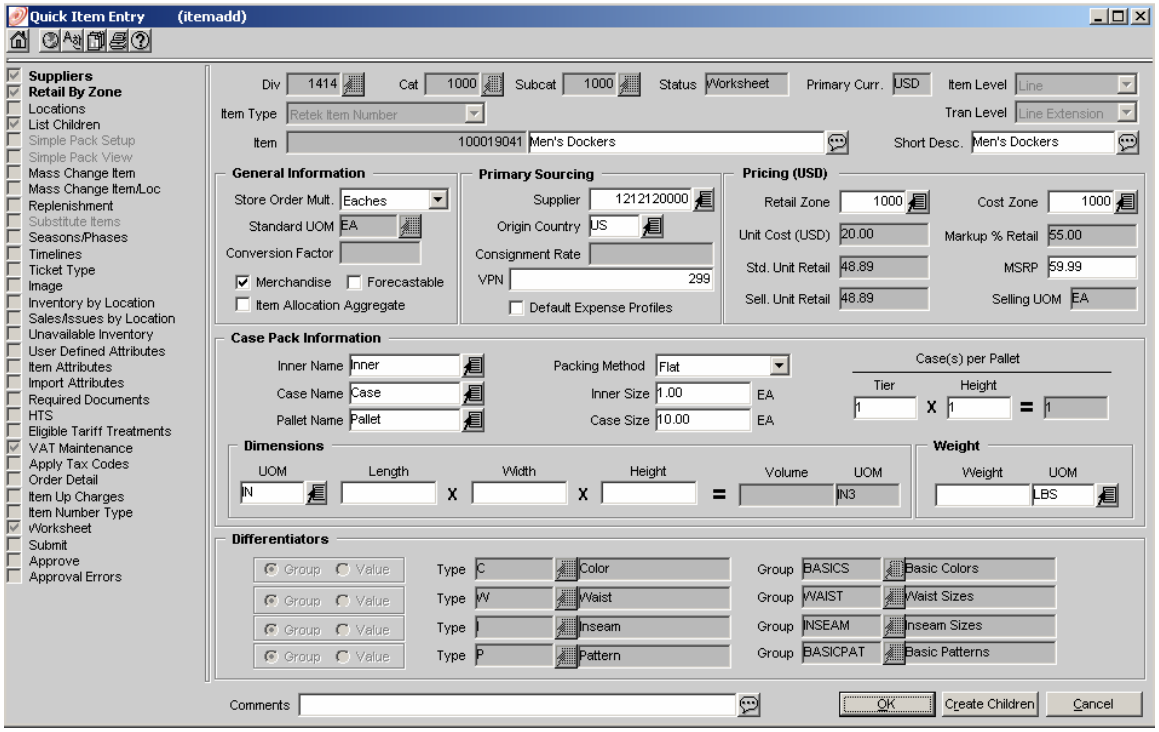
- Length (Item Supplier Country)
- Width (Item Supplier Country)
- Height (Item Supplier Country)
- Dimension UOM (Item Supplier Country)
- Weight (Item Supplier Country)
- Weight UOM (Item Supplier Country)

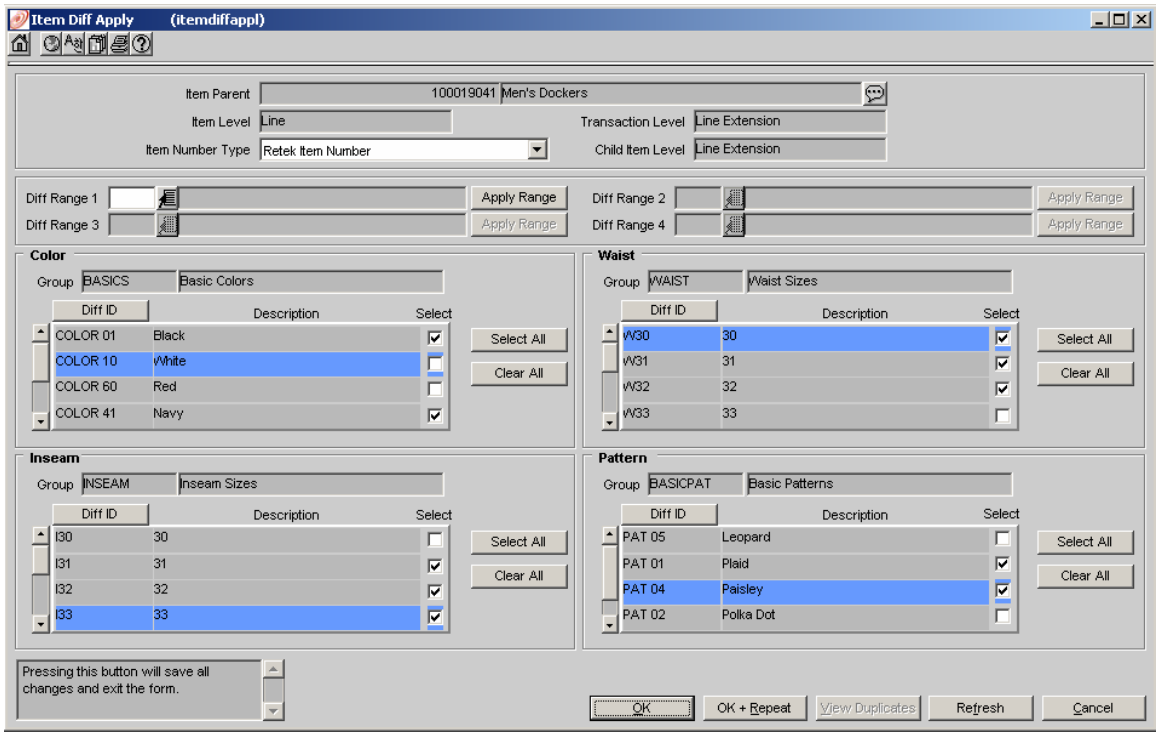
If the user is creating an item that has reference items there are 2 entry screens needed.

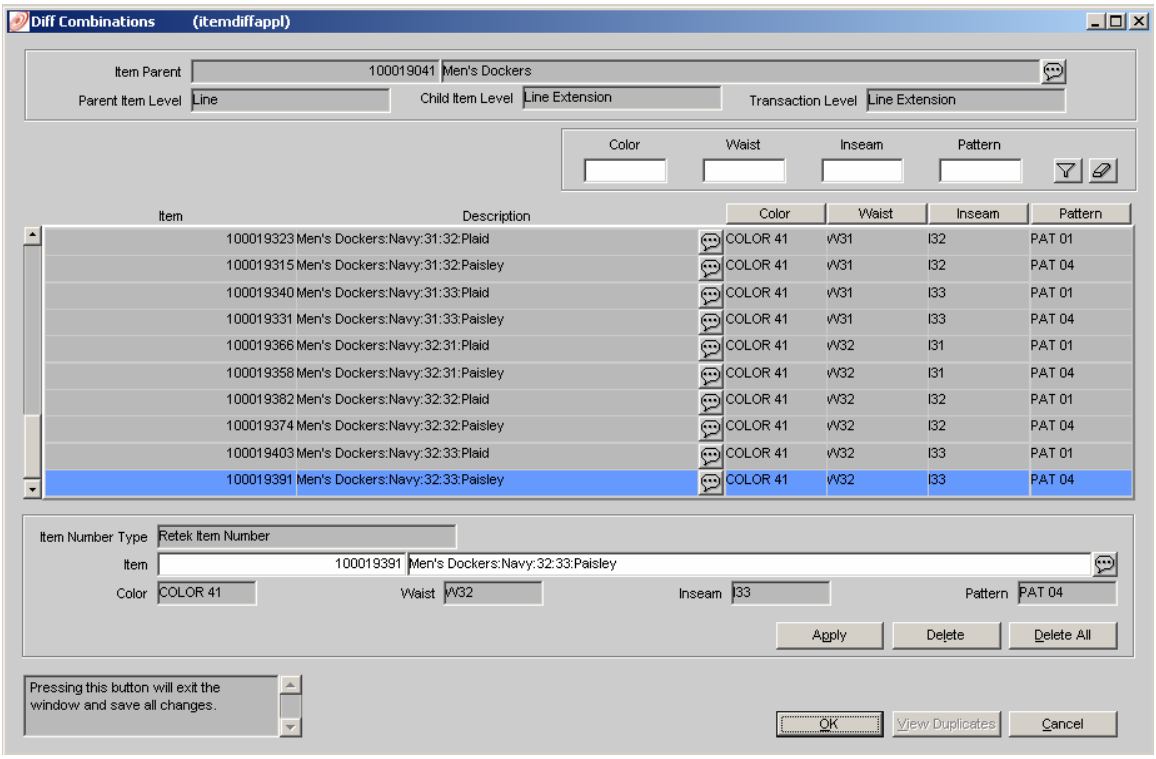


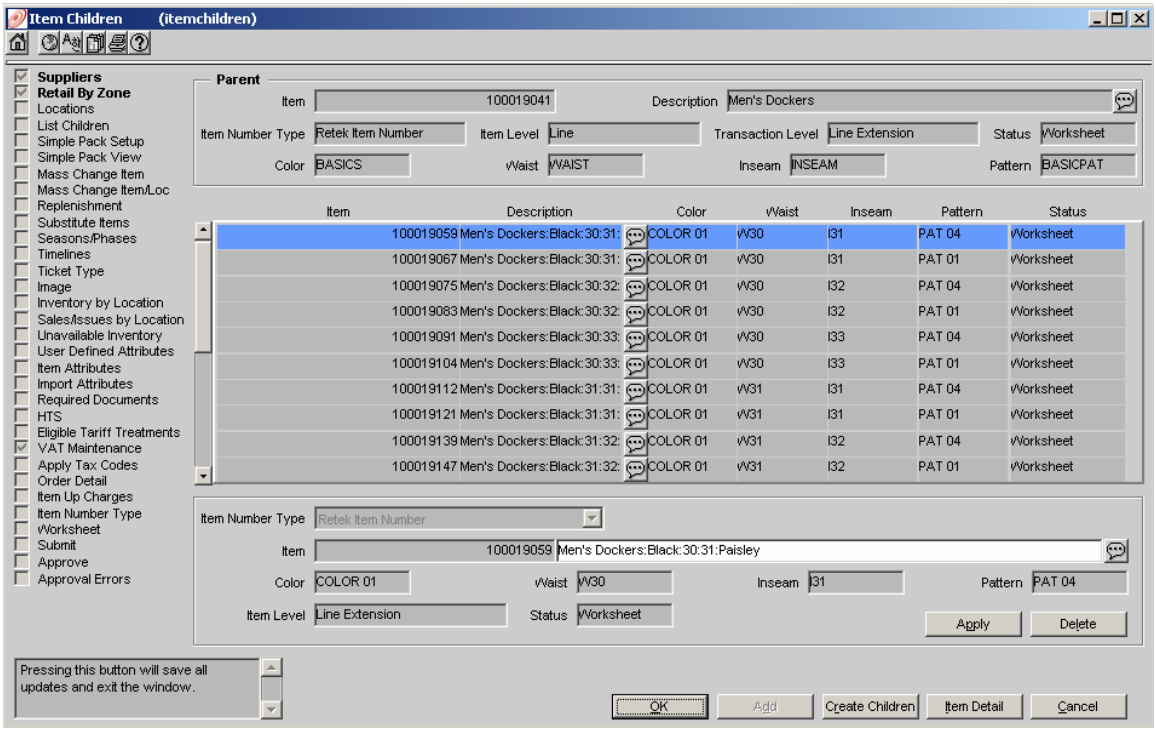


If the user is creating a fashion item that has children items there are 4 entry screens needed.



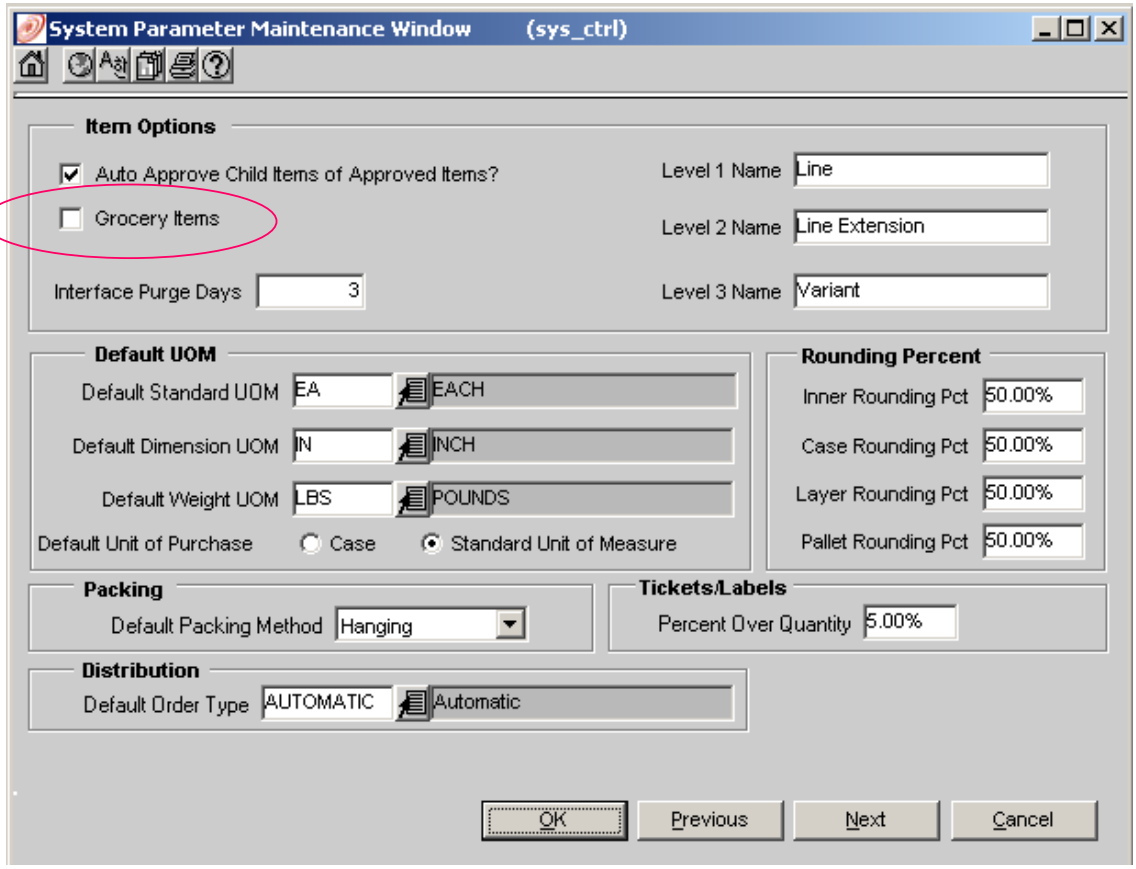



Diff Combinations (itemdiffappl)
Item Parent 100019041 Men's Dockers
Parent Item Level Line
Child Item Level Line Extension
Transaction Level Line Extension
Color Waist Inseam Pattern
Item Description Color Waist Inseam Pattern
100019323 Men's Dockers:Navy:31:32:Plaid COLOR 41 W31 I32 PAT 01
100019315 Men's Dockers:Navy:31:32:Paisley COLOR 41 W31 I32 PAT 04
100019340 Men's Dockers:Navy:31:33:Plaid COLOR 41 W31 I33 PAT 01
100019331 Men's Dockers:Navy:31:33:Paisley COLOR 41 W31 I33 PAT 04
100019366 Men's Dockers:Navy:32:31:Plaid COLOR 41 W32 I31 PAT 01
100019358 Men's Dockers:Navy:32:31:Paisley COLOR 41 W32 I31 PAT 04
100019382 Men's Dockers:Navy:32:32:Plaid COLOR 41 W32 I32 PAT 01
100019374 Men's Dockers:Navy:32:32:Paisley COLOR 41 W32 I32 PAT 04
100019403 Men's Dockers:Navy:32:33:Plaid COLOR 41 W32 I33 PAT 01
100019391 Men's Dockers:Navy:32:33:Paisley COLOR 41 W32 I33 PAT 04
Item Number Type Retek Item Number
Item 100019391 Men's Dockers:Navy:32:33:Paisley
Color COLOR 41 Waist W32 Inseam I33 Pattern PAT 04
Apply Delete Delete All
Pressing this button will exit the window and save all changes.
OK View Duplicates Cancel


Item Children (itemchildren)
Suppliers
Retail By Zone
Locations
List Children
Simple Pack Setup
Simple Pack View
Mass Change Item
Mass Change Item/Loc
Replenishment
Substitute Items
Seasons/Phases
Timelines
Ticket Type
Image
Inventory by Location
Sales/Issues by Location
Unavailable Inventory
User Defined Attributes
Item Attributes
Import Attributes
Required Documents
HTS
Eligible Tariff Treatments
VAT Maintenance
Apply Tax Codes
Order Detail
Item Up Charges
Item Number Type
Worksheet
Submit
Approve
Approval Errors
Parent
Item 100019041 Description Men's Dockers
Item Number Type Retek Item Number Item Level Line Transaction Level Line Extension Status Worksheet
Color BASICS Waist WAIST Inseam INSEAM Pattern BASICPAT
Item Description Color Waist Inseam Pattern Status
100019059 Men's Dockers:Black:30:31: COLOR 01 W30 I31 PAT 04 Worksheet
100019067 Men's Dockers:Black:30:31: COLOR 01 W30 I31 PAT 01 Worksheet
100019075 Men's Dockers:Black:30:32: COLOR 01 W30 I32 PAT 04 Worksheet
100019083 Men's Dockers:Black:30:32: COLOR 01 W30 I32 PAT 01 Worksheet
100019091 Men's Dockers:Black:30:33: COLOR 01 W30 I33 PAT 04 Worksheet
100019104 Men's Dockers:Black:30:33: COLOR 01 W30 I33 PAT 01 Worksheet
100019112 Men's Dockers:Black:31:31: COLOR 01 W31 I31 PAT 04 Worksheet
100019121 Men's Dockers:Black:31:31: COLOR 01 W31 I31 PAT 01 Worksheet
100019139 Men's Dockers:Black:31:32: COLOR 01 W31 I32 PAT 04 Worksheet
100019147 Men's Dockers:Black:31:32: COLOR 01 W31 I32 PAT 01 Worksheet
Item Number Type Retek Item Number
Item 100019059 Men's Dockers:Black:30:31:Paisley
Color COLOR 01 Waist W30 Inseam I31 Pattern PAT 04
Item Level Line Extension Status Worksheet
Apply Delete
Pressing this button will save all updates and exit the window.
OK Add Create Children Item Detail Cancel

System Variables (sys_ctrl)

A new indicator was added to the System Variables form. This checkbox indicates whether or not the retailer handles grocery merchandise items. This indicator will drive whether or not the user will have access to the Grocery Attributes window in the Quick Item Entry form by only displaying the Grocery Attributes button when the indicator is set to 'Y'es.



Sales Audit Appendix

Defect 357736 introduced two new forms to RMS 10.2. An overview is provided below. Please see the defect documentation for additional details.

Currency Rounding Rules Header Maintenance [sarrulhd]

This form allows you to maintain a list of rounding rules based on currency and country. Only one rule can be defined per currency/country combination. If a rule is applicable for all countries using a specific currency, set the country field to null. If a currency has more than one rule defined, (i) rule1 for currency = X and country = X (ii) rule2 for currency X and country = null; Country X will use rule1 whereas all other countries having currency X will use rule2. Both the Start Date and Status fields determine the effectivity of the rule. Start Date will only be applicable if the Status is Active. An Inactive rule will be inactive regardless if there is a Start Date.

You can access the Currency Rounding Rules Detail Maintenance window in order to edit the rounding rules range and round amount.

Currency Rounding Rules Detail Maintenance [sarruldt]

This form lets you set up the rounding rules range and the corresponding amount it rounds down/up to. A rounding rule range is defined by the lowest(inclusive) and highest(inclusive) ending amount. Ending amount can be before or after the decimal point for the currency. Lowest and highest ending amount are always defined with 4-decimals. Currency Decimal Places Retail in the Currency Maintenance window determines the number of decimals for the round amount. A set of rounding rules cannot have an overlapping range.

Ending amount before the decimal point – given an ending amount range of 0 to 9.9999 with round amount of 0; any ending amount values within that range will be rounded down to 0.

Ending amount after the decimal point - given an ending amount range of 0 to 0.2455 with round amount of 0; 0.2456 to 0.7455 with round amount of 0.75; 0.7456 to 1.0000 with round amount of 1. An ending amount 0.50 will be rounded up to 0.75.

Functional Specification for MDM Interfaces with RMS

Overview

As the enterprise store for Item information, MDM must interface with RMS to provide Item access in support of the customer's business processes. Currently, MDM will interface with RMS 10.2, RMS 10.1.7, and RMS 9.0.15. Integration with RMS 9 requires that the customer run the Oracle 9i database. Customers running Oracle 8i will need to upgrade to run these interfaces. Using the RIB architecture, these interfaces will be loosely coupled with RMS, meaning that MDM will be able to publish and subscribe to messages involving other applications. For RMS 9, a batch component will be written so that retailer's are not required to purchase the RIB. For each interface, a technical procedure will be developed to map MDM defined attributes to the receiving applications fields.

Functional Areas

Interfaces are required for the following functional areas: Suppliers, Locations, Items, Seasons and Phases, and Seed Data. These are individually detailed below.

Suppliers Publication from RMS

RMS owns all Supplier data, with the exception of certain UCCnet specific information (GLN) only needed in MDM for Item creation, MDM requires all Supplier data from RMS. MDM will own the Item-Supplier intersections. For MDM details, see FS71 Supplier and Locations Structures.

Ship Points will be setup in MDM. These are only setup per Item in RMS. MDM will have a Ship Point creation process. Interface will need to revisited based on Item-Supplier-ShipPt intersection analysis to satisfy publication back to RMS.

Publication out of RMS 10

Supplier: The current Supplier publisher for RMS 10 will be reused. MDM requires the following information:

- Supplier number
- Supplier name
- Supplier status
  - indicates whether the supplier is active or not
- currency code
  - indicates the primary currency for the supplier
- Default item lead time
  - default setting for the lead-time attribute at item-supplier-ship pt (country) level.

- Duns number
  - Duns and Bradstreet supplier identifying number. This will assist GLN mapping from UCCnet.
- Cost Change percent variance.
- Cost Change amount variance

Subscription into MDM

Supplier: All attributes passed from RMS will be added as attributes to the Supplier in MDM.

Locations Publication from RMS

For MDM Phase 1, RMS will own most Location data and MDM will maintain a mirror of this data, including Locations, Location Groups, and the Location Hierarchy. MDM will own the Item-Loc intersection. For MDM details, see FS71 Supplier and Locations Structures.

Publication out of RMS 10

Locations: The following information is required by MDM. Information for both Warehouses and Stores will be mapped into a single structure in MDM, with both Warehouse ID and Store ID mapping to the location identifier.

- Location name
- Location identifier
- Channel
- Stock holding indicator
- Selling location indicator
- Virtual/Physical location flag – default to physical when does not exist.
-

In general, Locations have the following grouping structures or capability:

- Hierarchy – A multi-level categorization of stores.
- Groups – A set of stores for various uses. This was previously Location Lists in RMS 10
- Subgroups – A limited hierarchical system of Groups.

Location Hierarchy

MDM will support hierarchies of locations with all the features of hierarchies of Items. The general hierarchy requirements of MDM will satisfy Organization Hierarchy as it exists in RMS, with enhancements including multiple hierarchies, inheritance and defaulting attribute values, optional parents and level skipping.

Location hierarchy maintenance is out of scope for Phase 1. During the initial release, MDM will contain a mirror of RMS via the locations interface.

Publication out of RMS 10

Organization Hierarchy: The following information is required for District, Area, Region, and Chain.

- Node identifiers
- Node names
- Node parent

Location Groups

MDM should support groups of locations with all the features of groups of items, including static, dynamic, and scheduled types; and criteria including attribute values, hierarchy membership, nested group membership, and logical combinations of these. Groups of items have been built with this in mind, and this should be relatively straightforward to achieve. The group construct in MDM easily satisfies the existing capabilities of Location Lists in RMS today.

Location group maintenance is out of scope for MDM Phase 1. User requirements for Location Groups will be satisfied via mirroring the structure of RMS.

Publication out of RMS 10

Location Groups: The following information is required to create the Location Groups structure in MDM.

- Location List identifier
- Location identifier
- Location type identifier

Location Sub-Groups

Location Sub-Groups are known as Zones in RMS. A Subgroups is a set of groups of locations which guarantee that every location belongs to one and only one group per subgroup. Subgroups are used today in RMS to assign defaults for pricing behavior, and to define sets of stores which can transfer goods to one another. When the user picks “Price Subgroup 1” for an item, the system knows that each location belongs to a single group a to n within subgroup 1, and that a given item-location should use the rule assigned to its group within price subgroup 1.

Table: An Example of Subgroups

	Group 1			Group 2			Group 3		
Sub Group	A	B	C	A	B	C	A	B	C
Stores	1-5	6-10	11-15	1-3	4-12	13-15	1,8-11, 15	2-7	12-14

Notes on Table:

- User choices are usually limited to Groups 1, 2, and 3.
- System understands sub-grouping under each group.
- Each subgroup guarantees a group assignment for each of the 15 stores.

Location Subgroup maintenance is out of scope for MDM Phase 1. User requirements for Subgroups will be satisfied via mirroring RMS data within one of the structures above..

Publication out of RMS 10

Cost Zones: All fields from the RMS Cost Subgroup, Cost Subgroup Group, and Cost Subgroup Group Loc tables are required for Cost Subgroups in MDM.

Price Zones: Price zone population in MDM is out of scope due to the ability to link to the Retail by Zone form in RMS

Item Publications from MDM

MDM is the data store of record for all Item information. Because of this, MDM will own and publish the majority of Item attributes and intersections. The goal for MDM is to publish a comprehensive set of Item data, which can then be manipulated as needed by the receiving interfaces. However, since external systems, including RMS, will currently make updates to Items that need to be propogated to MDM, all of the Item interfaces will be two-way interfaces.

Item creation in MDM can be performed in multiple steps and a publication can occur following combinations of certain steps. The following steps make up the Item create workflow in MDM:

- Hierarchy Selection and Generation of Item keys (Item)
- Item Dimension attributes (Item Supplier Country)
- Item Product Level attributes (Item)
- Item Supplier attributes (Item Supplier)
- Item Supplier Ship Point attributes (Item Supp Country)
- Item Location attributes (Item Locs and Item Loc Traits)
- Item Receiving Locations (Item Supplier Country Loc)
-

Publication to RMS will occur at these points:

- Item Level information complete. This allows for the population of the highest level RMS Item tables, allowing a reduced set of Item maintenance in RMS. Publication will occur for in worksheet status for a new Item.
- Supplier Ship Point information complete, includes Supplier and Dimensions. This is dependant on the Item Level information.
- Item Locations information populated. This is dependant on the Item Level information.
- Item Receiving Locations information populated. This is dependant on the Item Supplier Ship Point and Item Locations information.

The following RMS states must be mapped during integration. In RMS there are currently two levels at which Item's have status, Item and Item Location. These may just be mapped values to a calculated status during integration. Here are the Item Master Status's:

- 'W' = Worksheet: item setup in progress, cannot be used in system
- 'S' = Submitted: item setup complete and awaiting approval, cannot be used in system. MDM will not publish in Submitted status.
- 'A' = Approved: item is approved and can now be used throughout the system

Item

MDM owns Item setup and maintenance. For customers with an existing RMS implementation, Items already in the system will be moved to MDM during conversion. No online interface will be necessary. MDM will then map other needed attributes, such as the GTIN. The base Item interface publishes information from MDM to RMS, however due to Item create abilities during Purchase Order creation and complementary interface will return data to MDM.

Subscription for RMS 10

Item: Data for this interface will be pulled from the Item Master table. Fields TBD.

Set during Item Hierarchy Assignment and Product Level Attributes

- Item_Master:Dept
- Item_Master:Class
- Item_Master:Subclass
- Item_Master:Tran_Level
- Item_Master:Item
- Item_Master:Item_Number_Type
- Item_Master:Item_desc
- Item_Master:Short_Desc
- Item Master:Diff_1, Diff_2, Diff_3, Diff_4
- Item_Master:Orderable
- Item_Master:Sellable
- Item_Master:Pack_Ind

- Item_Master:Gift_Wrap_Ind
- Item_Master:Ship_Alone
- Item_Master:Simple_Pack
- Item_Master:Const_Dimension_Ind (default to No)
- Item_Master:Forecast_Ind (default to No)

Assigned while entering Dimensions:

- Item_Master:Standard UOM
- Item_Master:UOM Conversion Factor
- Item_Master:Package Size

Entering Components

- Pack_Item:Pack_No
- Pack_Item:Seq_No
- Pack_Item:Pack_Qty
- Pack_Item_Breakout:Item
- Pack_Item_Breakout:Comp_Pack_Qty

Entered as Product Level attributes

- Backorderable
  - Boolean
  - Maps to RMS
- Item_Master:Catch weight ind
- Item_Master:Comments
- Item_UDA:Convey Method
- Item_UDA:Employee Discount %
- Flavor - gets dropped or maps to a diff - (Item_Master:Diff)
- Item_Master:Handling sensitivity
  - initial values on code detail table as type ‘HSEN’
- Item_Master:Handling temp
  - values on code detail table as type ‘HTMP’
- Item_Image:Item
- Item_Image:Name
- Item_Image:Desc
- Item_Master:Mfg_rec_retail
- Item_Master:cost price zone

- Item_Master:Store order multiple
- Item_Master:Waste pct
- Item_Master:Waste type
- Item_Seasons:Season
- Item_Seasons:Phase

Item Locations

MDM will own the Item-Location intersection and publish this data to RMS. Item Location will be created during the Create Item workflows. Locations will be published separately from other Item data, but are dependant on having Item information in the RMS Item tables. Item Locations will be published at the Store location level and not using higher levels in the hierarchy.

Currently at the Item Location level here are the definitions of the current status's which must be passed to RMS.

- 'A' = Active, item is valid and can be ordered and sold
- 'I' = Inactive, item is valid but cannot be ordered or sold
- 'C' = Discontinued, item is valid and sellable but no longer orderable
- 'D' = Delete, item is invalid and cannot be ordered or sold

Subscription into RMS 10

ItemLoc: The following information will be entered in MDM for an item location and must be received by RMS 10.

- Item_Loc:Item Id
- Item_Loc:Location Id
- Item_Loc:Location Type (S or W)
- Item_Loc:taxable ind
- Item_Loc:local desc
- Item_Loc:local short desc
- Item_Loc:ti
- Item_Loc:hi
- Item_Loc:store_ord_mult
- Item_Loc:status
- Item_Loc:daily waste pct
- Item_Loc_Traits:launch date
- Item_Loc_Traits:qty key options
- Item_Loc_Traits:manual price entry
- Item_Loc_Traits:deposit code

- Item_Loc_Traits:food stamp ind
- Item_Loc_Traits:wic ind
- Item_Loc_Traits:proportional tare pct
- Item_Loc_Traits:fixed tare value
- Item_Loc_Traits:fixed tare uom
- Item_Loc_Traits:reward eligible ind
- Item_Loc_Traits:natl brand comp product
- Item_Loc_Traits:return policy
- Item_Loc_Traits:stop sale ind
- Item_Loc_Traits:elect mtk clubs
- Item_Loc_Traits:report code
- Item_Loc_Traits:req shelf life on selection
- Item_Loc_Traits:req shelf life on receipt
- Item_Loc_Traits:ib shelf life
- Item_Loc_Traits:store reoderable ind
- Item_Loc_Traits:rack size
- Item_Loc_Traits:full pallet product
- Item_Loc_Traits:in store market basket
- Item_Loc_Traits:storage location
- Item_Loc_Traits:alt storage loc
- Item_Loc_Traits:returnable ind
- Item_Loc_Traits:refundable ind
- Item_Loc_Traits:backorder ind

Item Supplier

MDM will own the Item-Supplier intersection and publish this data to RMS.

Subscription into RMS 10

ItemSupplier:

- Item_Supplier:Supplier
- Item_Supplier:Item
- Item_Supplier:Primary_Supp_Ind
- Item_Supplier:VPN
- Item_Supplier:Supplier label
  - Supplier's Short Description (30)
    - Maps to RMS 'Supplier label'
- Item_Supplier:Consignment rate
- Item_Supplier:Discontinue date
- Item_Supplier:Direct Ship Indicator
- Item_Supplier:Pallet name – default to "Pallet"
- Item_Supplier:Case name – default to "Case"
- Item_Supplier:Inner name – default to "Inner"

Item Supplier Country

MDM will own the Item-Supplier intersection and publish this data to RMS. At this level we can publish the majority of the dimensions attributes as well as the intersection.

Subscription into RMS 10

ItemSupplierCountry: The following attributes will be passed to RMS during the Supplier Ship Pt publication.

Entered during Supplier Ship Pt setup in MDM

- Item_Supp_Country;Supplier
- Item_Supp_Country:Origin Country
- Item_Supp_Country:Item
- Item_Supp_Country:Unit Cost
- Item_Supp_Country:Lead Time
- Item_Supp_Country:Pick up Lead Time
- Item_Supp_Country:Min Qty
- Item_Supp_Country:Max Qty
- Item_Supp_Country:Supp_Hier_Lvl1 (Manufacturer)

- Item_Supp_Country:Supp_Hier_Lvl2 (Distributor)
- Item_Supp_Country:Supp_Hier_Lvl3 (Wholesaler)
- Item_Supp_Country:Default_UOP
- Item_Supp_Country:Supp_Pack_Size
- Item_Supp_Country:Inner_Pack_Size
- Item_Supp_Country:Primary Supp Ind
- Item_Supp_Country:Primary Country Ind

Entered during Dimensions setup in MDM

- Item_Supp_Country_Dim:Tare Weight
- Item_Supp_Country_Dim:Tare Type
- Item_Supp_Country_Dim:LWH UOM / Item_Supp_Country:Dimension_UOM
- Item_Supp_Country_Dim:Length / Item_Supp_Country:Ship_Carton_Len
- Item_Supp_Country_Dim:Width / Item_Supp_Country:Ship_Carton_Wid
- Item_Supp_Country_Dim:Height / Item_Supp_Country:Ship_Carton_Hgt
- Item_Supp_Country_Dim:Liquid Volume
- Item_Supp_Country_Dim:Liquid Volume UOM
- Item_Supp_Country_Dim:Stat Cube
- Item_Supp_Country_Dim:weight UOM / Item_Supp_Country:Weight_UOM
- Item_Supp_Country_Dim:weight / Item_Supp_County:Ship_Carton_Wt
- Item_Supp_Country_Dim:net weight

Entered during Supplier setup in MDM and defaulted to the Ship Pt:

- Item_Supp_Country:TI
- Item_Supp_Country:HI
- Item_Supplier_Country:Unit Cost
- Item_Supp_Country_Dim:Presentation Method

Item Supp Country Loc

MDM will own the Item-Supp-Country-Loc intersection, called Item Receiving Locations in MDM, and publish this data to RMS. However, like Item this interface will be defined in both directions to support intersection information populated in RM

Subscription into RMS 10

ItemSupplier: The user supplies values for the following list of attributes which may be published after the creation of Item-Supplier-ShipPt and an Item-Location relationships.

- Item_Supp_Country_Loc:Unit_cost
- Item_Supp_Country_Loc:Primary_loc_ind
- Item_Supp_Country_Loc:Pick up Lead Time
- Item_Supp_Country_Loc:round level
- Item_Supp_Country_Loc:round case percent
- Item_Supp_Country_Loc:round layer percent
- Item_Supp_country_Loc:round pallet percent
- Item_Supp_Country_Loc:Origin_Country_ID (taken from the Ship Pt)

Item Groups

MDM provides the ability to place Items into groups, both statically and dynamically via a selection criteria. MDM owns this grouping structure and must publish groups to RMS. In RMS all Item groups published from MDM will be statically defined, meaning MDM will not publish to the SkuList_Criteria table. Dynamic groups are generated via query criteria only executed on the MDM database. All dynamic groups will be published to external applications upon creation, maintenance, and during a periodic batch cycle to capture items created since the last publication.

RMS users will have the ability to generate dynamic groups within RMS, but these will not be published to MDM and will only include items existing in RMS.

Publication out of MDM

Groups:

- Item Group identifier
- Item identifier

Subscription into RMS 10

Groups: Item groups are placed into the Item List tables: SkuList_Head and Skulist_Detail

- Skulist_Detail:Skulist Id
- Skulist_Detail:Item Id
- Skulist_Detail:Item level
- Skulist_Detail:Transaction level
- Skulist_Detail:Pack ind

- Skulist_Detail:Insert Id
- Skulist_Detail:Insert date
- Skulist_Detail:Create datetime
- Skulist_Detail:Last update datetime
- Skulist_Detail:Last update id
- Skulist_Head:Skulist
- Skulist_Head:Skulist description
- Skulist_Head:Create date
- Skulist_Head:Create id
- Skulist_Head:Static_Ind
- Skulist_Head:Last Rebuild Date
- Skulist_Head:User Security Ind
- Skulist_Head:Tax Prod Group Ind (Nullable)
- Skulist_Head:Comment Desc (Nullable)

Item User Defined Attributes and UDA Definitions

MDM does not distinguish between standard Item attributes and UDAs. MDM will publish Item data and it is the responsibility of the RMS Item subscriber to determine the proper location for Item attributes. However, MDM must publish any new UDA definitions to RMS prior to publishing data for that UDA. This data must be integrated into the RMS UDA, UDA_Values, and UDA_Item_Defaults tables. RMS can accept UDAs that are free-form text, dates, or a list of values, so all MDM UDAs will need to map to one of those types.

User must specify whether or not the new Attribute will be passed to RMS. The default behaviour will be to pass to RMS as a UDA.

Publication out of MDM

UDA Definitions: MDM must send the following data for integration with RMS. Much of this data can be derived from the MDM attribute definitions.

- UDA identifier
- UDA description
- Data Type (Number, Alphanumeric, Date, optional))
- Display type (Free form, List of Values, Date)
- Data length
- Single value Ind
- UDA Values

Subscription into RMS 10

UDA Definitions: RMS 10 must populate the following fields with data passed from MDM.

- UDA identifier
- UDA description
- Module (only valid value is “ITEM”)
- Display Type
- Data Type
- Data Length
- Single Value Ind
- UDA_Values:UDA_Value
- UDA_Value:UDA_Value_Desc
- UDA_Item_Defaults:UDA_ID
- UDA_Item_Defaults:Seq_No
- UDA_Item_Defaults:Dept
- UDA_Item_Defaults:Class (Nullable)
- UDA_Item_Defaults:Subclass (Nullable)
- UDA_Item_Defaults:UDA_Value (Nullable)
- UDA_Item_Defaults:Required Ind
- UDA_Item_Defaults:Hierarchy Value

Item Diffs and Diff Maintenance

MDM will not implement the current RMS Diff structure, but will provide the same functionality through hierarchical attributes. Because of this, MDM will publish Item data and it is the responsibility of the RMS subscriber to map the specific attributes defined as Diffs in RMS appropriately. Diffs and Diff Groups will be maintained in MDM using the attributes structures and changes integrated to RMS. Currently, RMS uses Diffs to create certain Item attributes, such as SKU, which will now be passed from MDM. To support RMS, MDM will limit the number of diffs set on an Item to 4. MDM will maintain this using a business rule during Item create/maintain workflows.

Item Publications from RMS

Because MDM is the central Item data store, any applications creating Items during process not performed in MDM must send that data to MDM.

Item

RMS will publish Item data to MDM for Items created during Purchase Order creation. In this instance RMS creates SKU level items from an existing parent Style, defaulting most of the attributes to achieve a fully formed Item.

Publication out of RMS 10

Item: Item Master attributes and Item intersections will be published from RMS, including Item-Supplier, Item-Supplier-Country, Item-Supplier-Country-Loc, Item-Seasons, and Item-Locations. RMS will publish all data specified in the corresponding subscription interfaces, above

Merchandise Hierarchy

MDM will publish Merchandise Hierarchy information to RMS as part of the Item publication. However, modifications to the Merchandise Hierarchy must take place in RMS due to RMS dependancies and will not be allowed in MDM. This includes reclassification, or moving nodes within the Merchandise Hierarchy, which must be interfaced from RMS to MDM.

Publication out of RMS 10

MerchHier: All Merchandise Hierarchy data will be published by RMS, including Division, Group, Dept, Class, and Subclass. This is an existing interface from RMS.

Seasons and Phases Publication from RMS

Seasons information set as Item attributes will be passed to RMS as part of the Item interface. However, base Seasons data will be maintained in RMS and sent to MDM as a list. MDM will not modify this data, but will simply be using the list as valid values for the Seasons and Phases Item attributes.

Publication out of RMS 10

Seasons: All fields from the RMS Seasons table are required in MDM.

Phases: All fields from the RMS Phases table are required in MDM.

Seed Data Publications from RMS

RMS data is needed in MDM during initial population as valid values for Item Attributes. MDM will simply maintain the needed attribute values for these fields.

Unit of Measure

RMS will own Unit of Measure data and this will be mirrored in MDM. UoM is stored on the RMS Codes tables. To support possible other UoM's in MDM due to inducted Items, MDM will need to provide mapping functionality to the RMS units of measure. This codes interface can be used for other Codes tables information. MDM will also need to retain Unit of Measure conversion data which will be inserted via one time script.

Publication out of RMS 10

Codes: MDM will utilize the current RMS Code Head and Code Detail publisher. MDM requires the following data:

- Code Head: Code Type
- Code Head: Code Type Description
- Code Detail: Code Type
- Code Detail: Code
- Code Detail: Code Description
- Code Detail: Code Seq

Currency and Exchange Rates

MDM will mirror the RMS list of currencies from the Currency table in RMS. This interface must also populate exchange rates periodically in MDM. Typically RMS is populated via a feed from an external source. It is during implementation to accept that feed into MDM and not utilize this interface for Exchange Rate data if appropriate.

Publication out of RMS 10

Currency: Information taken from the RMS Currency and Currency_Rates tables must be populated in MDM. This includes:

- Currency code
- Currency description
- Exchange rate
- Effective date
- Exchange type

Countries

MDM will mirror the RMS list of countries from the Country table in RMS.

Publication out of RMS 10

Country: Information taken from the RMS Country table must be populated in MDM. This includes:

- Country ID
- Country description

Considerations

Localization

All Localized MDM data must be passed on all outbound interfaces. MDM will not receive localized information on inbound interfaces due to limitations in external systems. Prior to the integration message entering an external system, the language accepted by that system will be determined and only that language will be populated.

RMS/MDM New Integration Point Summary Chart

Function	Form Name	Integration Needed	Integration Direction	MDM Access Point (container)	New RMS access? (RMS Patch)
Sales and Inventory lookup	Unavailable Inventory	N	None	None	Add to the Inventory folder
	Item Location Inventory				Currently available from the Inventory folder
	Sales Info/Issues by Location				
Item ELC & Importing	Item Expense Maintenance	Y	MDM to RMS	Supplier and Supplier Ship Point	
	Item /Supplier Origin Country List (ELC view)			Supplier Ship Point	
	Item Import Attributes			Assign Product **Attributes**	
	Required Documents				
	Item HTS Maintenance				
	Item Eligible Tariff Treatment				
Retail Pricing	Item Retail Price by Zone	Y	MDM to RMS	Assign Product **Attributes**	
Bracket Costing	Item Supplier Country Location Bracket Cost	Y	MDM to RMS	Supplier *and* Supplier Ship Point *and* Receiving Locations to Ship Points	
Replenishment Set-up	Replenishment Attribute Maintenance	N	None	None	Add to the Inventory folder
	Substitute Item Maintenance				
Ticketing	Item Ticket Detail	Y	MDM to RMS	Assign Product **Attributes**	
Item Timeline set-up	Timeline	Y	MDM to RMS	Assign Product **Attributes**	
Tax/VAT	Tax Rate View	Y	MDM to RMS	Assign Product **Locations**	
	Item Tax Codes			Assign Product **Attributes**	
	VAT Item Maintenance				
Item Upcharges	Item Upcharges	Y	MDM to RMS	Assign Product **Attributes**	
Item List Use	Many of the Item List screens in RMS	Y	RMS to MDM	Item Group	