Retek® Merchandising System™ 9.0.15

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

Printed in the United States of America.

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

You may choose to apply only a portion of the enclosed patch, based on the Retek Merchandising System (RMS) functionality you are currently running. Retek strongly advises that you review each enclosed DEFECT document immediately in order to judge the impact of not applying a given fix. When Retek Customer Support investigates a potential software bug, they assume that you have applied all patches for a given module before the issue was discovered.

The RMS 9.0.15 patch contains both general 9.0.15 patch release fixes and Master Data Management (MDM) functional enhancement. (The 'Base RMS functionality fixes' and 'Base RMS functionality enhancements' section of this document provides more details.) For detailed information on what is included regarding software fixes in this patch release, refer to the 9.0.15 patch documentation located in the doc folder. Before you apply the RMS 9.0.15 patch:

- Check that RMS 9.0 is installed.
- Check that the following patches have been applied:
  - RMS 9.0.1.0 patch
  - RMS 9.0.2.0 patch
  - RMS 9.0.3.0 patch
  - RMS 9.0.4 patch
  - RMS 9.0.5 patch
  - RMS 9.0.6 patch
  - RMS 9.0.7 patch
  - RMS 9.0.8 patch
  - RMS 9.0.9 patch
  - RMS 9.0.10 patch
  - RMS 9.0.11 patch
  - RMS 9.0.12 patch
  - RMS 9.0.13 patch
  - RMS 9.0.14 patch

Note that an upgrade to Oracle 9.2.0.x is required before applying the full RMS 9.0.15 patch. Clients utilizing Master Data Management (MDM) will also need to upgrade to Oracle 9.2.0.x. Modules impacted by the Oracle 9i upgrade are listed in the appendix. Please refer to the RMS 9.0.14 Release Notes on the 9i upgrade approach. Note that there were several Oracle 8i/9i related batch compilation issues in RMS 9.0.14. See DEFECT 362868 for the resolution details. This fix may be needed to complete the upgrade.

Before applying the patch source files over your code:

- Note whether customizations have been made to the module. If so, then the customizations must be reapplied to the new version of the module (or the fix may need to be applied to the custom version of the code).
- Note the cutoff date for RMS 9.0.15 code was March 15, 2004. Fixes that you've received after this date and already applied to your environment will require special consideration when applying this patch.
- Copy the original files to a different directory before copying over them in case they need to be referred to at a later date.

DEFECT documentation (DEFECT MODULE XREF 90.xls)

A DEFECT fix is a modification to base Retek code (a bug fix, a performance enhancement, or a functional enhancement). Each DEFECT fix that is included in this patch has a corresponding DEFECT document in the \doc\defect_doc folder titled <DEFECT#> <module>.doc, such as "1557 nxprcno.doc". There is also a document in that same folder entitled "DEFECT MODULE XREF 90.xls", that lists every DEFECT # and the accompanying modules/scripts that are included in the patch.

In order to provide better customer service, we have decided to create a more comprehensive, user-friendly version of our DEFECT module cross-reference Excel document. This updated document includes a full list of all the previous patch DEFECT modules, plus new tabs showing DEFECTs related to the current patch, and current DEFECTs broken out by module types. We hope this will assist in installing and maintaining patch-related projects. We will follow this format for all upcoming patches.

Enclosed is an updated version of the cross-reference document. Each DEFECT document should be reviewed fully before this patch is implemented. What follows is a list of the more significant fixes in the 9.0.15 patch.

Base RMS functionality enhancements

Retek's Master Data Management (MDM) provides seamless integration from MDM 10.0 to RMS 9.0.15. All item data created in MDM is published to RMS. Integration with MDM functionality is provided in addition to the general product fixes included in the RMS 9.0.15 release.

With the introduction of MDM, Retek now offers a more robust, enhanced 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 Integration with Master Data Management appendix provides an overview of the MDM interfaces with RMS by functional area.

Base RMS functionality fixes

The RMS 9.0.15 patch contains general product fixes related to the following functional areas: Batch; Clearance; Cost Zone Group; Expenses; Foundation Data; General APIs; HTS; Invoice Matching; Item List; Item Location; Item Maintenance; Location List; Organizational Hierarchy; Partner, Point of Sales; Price Management; Promotion; Purchase Order; Sales Audit; Seasons; Stock Ledger; Store; Supplier; Ticket; Trade Management; Transfers; Transportation; User Defined Attributes; Warehouse. Below is a listing of defects by functional. We strongly recommend that you become familiar with the documents related to all of the following issues for this patch.

Batch

DEFECT 362868

Batch compilation issues with RMS 9.0.14 patch release on Oracle 9i.

Clearance

DEFECT 359510

The items in the Clearance Item List form are not listed in order. This makes it hard for the end-user to identify a specific SKU.

DEFECT 359064

User receives the following error while trying to add a new style to a clearance, "Item is not attached to any store or it is not Active at any store", even though the item is active in that zone.

DEFECT 359968

The "Apply All" button raises the error "The clearance price is greater than the unit retail for item %s1 in zone group %s2, zone %s3. A clearance markdown cannot increase the price of an item." even if the clearance price is lesser than the unit retail for the item in the specified zone group / zone.

Cost Zone Group

DEFECT 361592

Cost Zone Group data will be published out of RMS for integration to MDM.

Expenses

DEFECT 361139

Cost components do not default onto order expenses if currency exchange rate types P (Purchase Order) or U (Custom Entry) exist on the currency_rates table for that currency.

Foundation Data

DEFECT 361171

Country and Currency data will be published out of RMS for integration to MDM.

General APIs

DEFECT 360487

Added various functions that support the new MDM API's.

HTS

DEFECT 360161

Htsupld.pc rejects records on the HTS 2004 file when it should not.

DEFECT 360244

HTSUPLD.PC program aborts with core when records in ITEM_HTS table for a HTS are more than RESTART_CONTROL.COMMIT_MAX_CTR.

Invoice Matching

DEFECT 360200

When creating a stand-alone Credit Memo, the amount keyed into Total Merchandise VAT Included Discount gets converted to zero when this form is closed and opened again.

While creating a new stand-alone Credit Memo the VAT Code field and Merchandise VAT Details button are disabled.

DEFECT 359278

In the Invoice Matching form, for a Debit Memo created automatically by RTV, when the Merchandise VAT Details button is clicked and no details are entered and OK is clicked to navigate back on the Invoice Matching form, the Total Merchandise VAT Including Discount is deleted but the total includes the now deleted amount.

DEFECT 359125

The Payment Terms and Payment Method fields are not automatically populated after entering the Supplier Number while creating Stand-Alone Credit Memo, Stand-Alone Debit Memo or Stand-Alone Credit Note Request.

Item List

DEFECT 361594

Item List will be subscribed into RMS for integration to MDM.

Item Location

DEFECT 361597

Item Location will be subscribed into RMS for integration to MDM.

Item Maintenance

DEFECT 358651

Required documents attached to a style are not being inherited by its fashion SKU's.

DEFECT 358570

When receiving a shipment for a purchase order for several fashion SKUs that are under a single fashion style, the quantity received for that fashion style is updated incorrectly.

DEFECT 361391

Create API for subscription to Items from MDM. Anyone implementing MDM will use MDM as the master for item information, with RMS accepting item information from MDM.

MDM will call the API via a service layer, instead of the normal publishing/subscribing architecture. The service call to RMS will call a new service wrapper, RMSSVC_XITEM. RMSSVC_XITEM will call the necessary functions in RMSSUB_XITEM. For more information about the Retek Service layer, please refer to Retek Integration Bus documentation.

Location List

DEFECT 361589

Location List data will be published out of RMS for integration to MDM.

Organizational Hierarchy

DEFECT 361588

Organizational hierarchy data will be published out of RMS for integration to MDM.

Partner

DEFECT 361651

Partner data will be published out of RMS for integration to MDM.

Point of Sales

DEFECT 361867

Posupld.pc does not write any records to the lock file and Terminated OK eventhough the locking error exists while processing.

Price Management

DEFECT 360202

The item LOV button in pchist always returns the result of the first search criteria.

DEFECT 358401

The field for the number of days (after the out of stock date) when the clearance will be reset is being required by the pcclclss form.

Promotions

DEFECT 360666

When a promotion for a sku is extracted, POS_MODS.UPC is not populated.

DEFECT 359977

It is not possible to rebuild an item list in case of mix and match promotion.

DEFECT 360417

When attempting to select an item list for a multi-unit promotion via the list of values (LOV) on the (threshld) form, not all of the existing item lists are available in the LOV. It is still possible to add item lists manually, even if they are not available in this LOV.

Purchase Order

DEFECT 360276

Unique constraint error appears on stage_purged_shipskus table while running ordprg.pc program.

Sales Audit

DEFECT 359570

Performance issue with saexprdw.pc. A large amount of time is taken for exporting the transactions of one day.

DEFECT 360263

The negative tender amount is not reflected in the export file to Data Warehouse(RDWF).

Season

DEFECT 361587

Season data will be published out of RMS for integration to MDM.

Stock Ledger

DEFECT 350640

MAX_ORACLE_ARRAY_SIZE is not properly applied in salweek.pc.

DEFECT 359650

Peformance enhancement for salmth.pc batch program.

DEFECT 362085

Salmth.pc inserts incorrect data in the MONTH_DATA table.

Store

DEFECT 360139

MDM will have integration points with RMS 9. To complete Item creation, MDM requires all Store data. An RMS 9 publishing API will populate the store data.

Supplier

DEFECT 360142

MDM will have integration points with RMS 9. To complete Item creation, MDM requires all Supplier data. An RMS 9 publishing API will populate the supplier data.

Tickets

DEFECT 358572

When creating ticket request for an item with location type of 'All Stores', all stores are displayed in the multi-view block including the stores that are already closed.

DEFECT 359063

When the ticket type of a fashion style is modified, the change is not reflected to its fashion SKUs.

Trade Management

DEFECT 361577

When finalizing the transportation for an order that has more than one transportations created with the same VVE and the order contains two pack item that contains the same component items, the form display the following error message "ORA-00001: Unique contraint (CSRMS90B.PK_CE_CHARGES) violated returned by program unit CE_CHARGES_SQL.INSERT_COMPS with error code %s".

Transfers

DEFECT 358614

When an existing transfer for an item and location is opened in edit mode and no change has been made to the transfer details, a new transfer for the same item and location cannot be created, and the message "Table %s1 with key values %s2, %s3 is locked by another user." is displayed.

DEFECT 358900

When a transfer is manually closed online, the associated shipment status is set to Received even though the shipment has not been received.

DEFECT 358481

When closing a transfer where the received quantity is greater than the ship quantity, the stock on hand of the source location is decreased by the difference between the received quantity and the ship quantity.

DEFECT 348192

The rdmuitsf.pc logic considers only the scenario where the transfer has been partly shipped i.e. shipsku records are created. In the scenario where a transfer has only been extracted (i.e. tsfdnld.pc creates only the shipment record) and never shipped, the rdmuitsf.pc batch will not process the error line from rdm_if_errors. The tranfer's quantities and status is incorrect.

DEFECT 358866

When receiving the second shipment for a partially received purchase order containing two items, the auto- generated transfers are not created and updated to reflect the receipt of the second shipment.

Transportation

DEFECT 361145

Unable to re-finalize a Vessel/Voyage/ETD even though there are transportation records for the same VVE combination that have not been finalized for different import countries.

User Defined Attributes

DEFECT 361593

User Defined Attribute will be subscribed into RMS for integration to MDM.

Warehouse

DEFECT 360140

MDM will have integration points with RMS 9. To complete Item creation, MDM requires all Warehouse data. An RMS 9 publishing API will populate the warehouse data

RMS 9.0.15 Oracle 9i – Defects and Dependent Modules

Most objects dependent on Oracle 9i features are associated with the Master Data Management (MDM) functional enhancement. See the following Defect documents for more information.

- Defect 360139: To complete Item creation, MDM requires all Store data. An RMS 9 publishing API will populate the store data.
- Defect 360140: To complete Item creation, MDM requires all Warehouse data. An RMS 9 publishing API will populate the warehouse data.
- Defect 360142: To complete Item creation, MDM requires all Supplier data. An RMS 9 publishing API will populate the supplier data.
- Defect 360487: Added various functions that support the new API's in RMS 9 for integration with MDM. The rib_* modules are taken from RIB 10.3.4 code.
- Defect 361171: Country and Currency data will be published out of RMS for integration to MDM.
- Defect 361391: Create API for subscription to Items from MDM. Anyone implementing MDM will use MDM as the master for item information, with RMS accepting item information from MDM.
- Defect 361588: Organizational hierarchy data will be published out of RMS for integration to MDM.
- Defect 361589: Location List data will be published out of RMS for integration to MDM.
- Defect 361592: Cost Zone Group data will be published out of RMS for integration to MDM.
- Defect 361593: User Defined Attribute will be subscribed into RMS for integration to MDM.
- Defect 361594: Item List will be subscribed into RMS for integration to MDM.
- Defect 361597: Item Location will be subscribed into RMS for integration to MDM.
- Defect 361651: Partner data will be published out of RMS for integration to MDM.
- Defect 360487: General Changes needed for MDM API's
- Defect 362728: Forms need modifying in order to be called from MDM.

The following RMS 9.0.15 objects require Oracle 9i features.

- Packages:
  - api_libraryb.pls
  - api_librarys.pls
  - apicodes.pls
  - orgsqls.pls
  - rmsmfm_costzngrpb.pls
  - rmsmfm_costzngrps.pls
  - rmsmfm_loclistb.pls
  - rmsmfm_loclists.pls
  - rmsmfm_orghierb.pls
  - rmsmfm_orghiers.pls
  - rmsmfm_partnerb.pls
  - rmsmfm_partners.pls
  - rmsmfm_seasonb.pls
  - rmsmfm_seasons.pls
  - rmsmfm_seedobjb.pls
  - rmsmfm_seedobjs.pls
  - rmsmfm_storeb.pls
  - rmsmfm_stores.pls
  - rmsmfm_supplierb.pls
  - rmsmfm_suppliers.pls
  - rmsmfm_whb.pls
  - rmsmfm_whs.pls
  - rmssub_itemlistb.pls
  - rmssub_itemlists.pls
  - rmssub_itemlistsqlb.pls
  - rmssub_itemlistsqls.pls
  - rmssub_itemlistvalb.pls
  - rmssub_itemlistvals.pls
  - rmssub_xitemb.pls
  - rmssub_xitemlocb.pls
  - rmssub_xitemlocs.pls
  - rmssub_xitemlocsqlb.pls

- rmssub_xitemlocsqls.pls
- rmssub_xitemlocvalb.pls
- rmssub_xitemlocvals.pls
- rmssub_xitems.pls
- rmssub_xitemsqlb.pls
- rmssub_xitemsqls.pls
- rmssub_xitemvalb.pls
- rmssub_xitemvals.pls
- rmssub_xudab.pls
- rmssub_xudas.pls
- rmssub_xudasqlb.pls
- rmssub_xudasqls.pls
- rmssub_xudavalb.pls
- rmssub_xudavals.pls
- rmssvc_xitems.pls
- rmssvc_xitemb.pls
- store_xmlb.pls
- store_xmls.pls
- supplier_xmlb.pls
- supplier_xmls.pls
- wh_xmlb.pls
- wh_xmls.pls
- xitemsubb.pls
- xitemsubs.pls

- XML Oracle_Objects:
  - RIB_OBJECT.SQL
  - RIB_OBJ_UTIL.PKB
  - CreateClientRibObjects.sql
  - CreateCostZnGrpRibObjects.sql
  - CreateSeedobjRibObjects.sql
  - CreateItemListRibObjects.sql
  - CreateLocListRibObjects.sql
  - CreateMrchHrRibObjects.sql
  - CreateOrgHierRibObjects.sql
  - CreateSeasonRibObjects.sql
  - CreateStatusRibObjects.sql
  - CreateXDiffGrpRibObjects.sql
  - CreateXDiffIDRibObjects.sql
  - CreateXItemLocRibObjects.sql
  - CreateXItemRibObjects.sql
  - CreateXUDARibObjects.sql

- XML Utilities:
  - rib_codes.pls
  - rib_ddl.sql
  - rib_doctypes_rms.ctl
  - rib_doctypes_rms.dat
  - rib_message_seq.sql
  - rib_settings_tables.sql
  - rib_settings_values.sql
  - rib_sxw.pkb
  - rib_sxw.pks
  - rib_type_settings_values.sql
  - rib_xmlb.pls
  - rib_xmls.pls

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 9

Supplier: The current Supplier publisher for RMS 10 will be reused with slight modifications for RMS 9. 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 9

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 9
- 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

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

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 9

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 9

Item: Data for this interface will be inserted into multiple Item tables, win_sku, rag_style, and packhead. The RMS 9 Diff equivalents must be inserted

Set during Item Hierarchy Assignment and Product Level Attributes

- Win_SKUS/Rag_Style/Packhead:Dept
- Win_SKUS/Rag_Style/Packhead:Class
- Win_SKUS/Rag_Style/Packhead:Subclass
- Rag_Style/Rag_SKU:Style
- Win_SKUS:SKU_desc / Rag_Style:Style_Desc / PackHead:Pack_Desc
- Win_SKUS/Rag_Style/Packhead:Short_Desc
- PackHead:Orderable – only on Packs in RMS 9
- PackHead:Sellable - only on Packs in RMS 9 (set to Merchandise Ind for Rag Style and Win_SKUS)
- Item_Master:Pack_Ind
- Item_Master:Gift_Wrap_Ind
- Item_Master:Ship_Alone
- Packhead:Simple_Pack
- Item_Master:Const_Dimension_Ind (default to No)
- Win_SKUS/Rag_Style/PackHead:Forecast Ind (default to No)
- Rag_Skus:Color
- Rag_Skus:Size 1
- Rag_Skus:Size 2 (optional)

- Rag_Style:Size_Group_1
- Rag_Style_Size_Group_2 (optional)

Assigned while entering Dimensions:

- Win_SKUS/Rag_Style/Pack_Head:Standard UOM
- Win_SKUS/Rag_Style/Pack_Head:UOM Conversion Factor

Entering Components

- Pack_Item:Pack_No
- Pack_Item:Seq_No
- Pack_Item:Pack_Qty
- Pack_SKU:SKU
- Pack_SKU:Pack_SKU_Qty

Entered as Product Level attributes

- Item_Master:Catch weight ind
- Item_Master:Comments
- UDA_Item_FF:Convey Method
- UDA_Item:FF:Employee Discount %
- Item_Master:Handling sensitivity
- Item_Master:Handling temp
- Item_Image:Item
- Item_Image:Name
- Item_Image:Desc
- Item_Image:Image_Type
- Item_Master:Mfg_rec_retail
- Win_SKU/Rag_Style/PackHead:cost zone group
- Win_SKU/Rag_Style/PackHead:Store order multiple
- Desc_Look:Waste pct
- Desc_Look:Waste type
- Desc_Look:Desc_Up
- Desc_Look:System_Ind – derived based on the Item type
- PackHead:Primary_Repl_Ind
- PackHead:Contains_Inner_Ind

- 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 for RMS 9

ItemLoc: This interface must insert data into the Win_Store, Win_WH, Rag_Skus_Store, Rag_Skus_WH, Rag_Style_Store, Rag_Style_WH, PackStore, PackWH.

Bullets in Red are required in RMS but not in MDM.

Bullets in blue exist in MDM but not in RMS.

- ***_WH/***_Store:SKU/Pack
- *_Store:Store, *_WH:WH
- *_WH/*_Store:Unit_Cost – need to add to MDM
- *_WH/*_Store:Average Cost
- *_WH/*_Store:Stock On Hand – (not MDM info, default to zero on create)
- *_WH/*_Store:In Transit Qty – (not MDM info, default to zero on create)
- *_WH/*_Store:Tsf_Reserved_Qty – (not MDM info, default to zero on create)
- *_WH/*_Store:Tsf_Expected_Qty – (not MDM info, default to zero on create)
- *_WH:Pack_Comp_SOH – (not MDM info, default to zero on create)
- *_WH:Pack_Comp_Intran – (not MDM info, default to zero on create)
- *_WH:Pack_Comp_Resv – (not MDM info, default to zero on create)
- *_WH/*_Store:Rtv_Qty – (not MDM info, default to zero on create)
- *_WH/*_Store:Non_Sellable_Qty – (not MDM info, default to zero on create)
- *_Store:Clear_Ind – (not MDM info, default to zero on create)
- *_Store:taxable ind
- *_Store:Status
- Item_Loc:local desc
- Item_Loc:local short desc
- Item_Loc:ti
- Item_Loc:hi
- Item_Loc:store_ord_mult
- *_Store: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
- Rag_Style_St/WH:Style
- Rag_Style_St/WH:Store
- Rag_Style_St/WH:Primary_Supplier – does not exist in MDM at this level (can be null in RMS)
- PackStore:Primary_Supplier – does not exist in MDM at this level (can be null RMS)

Item Supplier

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

Subscription into RMS 9

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 (does not Integrate to RMS 9
- Item_Supplier:Direct Ship Indicator (does not Integrate to RMS 9)
- 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 9

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:Dimension_UOM
- Item_Supp_Country:Ship_Carton_Len
- Item_Supp_Country:Ship_Carton_Wid
- Item_Supp_Country:Ship_Carton_Hgt
- Item_Supp_Country:Weight_UOM
- Item_Supp_County:Ship_Carton_Wt

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

- Item_Supplier_Country:Unit Cost
- Item_Supp_Country_Dim:Presentation Method

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 9

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:System Ind
- Skulist_Detail:Insert Id
- Skulist_Detail:Insert date
- 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 behavior 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 9

UDA Definitions: RMS 9 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 9

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 9

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 9

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 9

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 9

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 9

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	Added to the Inventory folder
	Item Location Inventory				
	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**	
Replenishment Set-up	Replenishment Attribute Maintenance	N	None	None	Added 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	