

Retek® 10.1 Integration Bus



Integration Guide

**Retek Distribution Management 10.1
Message Family Manager Designs**



The software described in this documentation is furnished under a license agreement and may be used only in accordance with the terms of the agreement.

No part of this documentation may be reproduced or transmitted in any form or by any means without the express written permission of Retek Inc., Retek on the Mall, 950 Nicollet Mall, Minneapolis, MN 55403.

Information in this documentation is subject to change without notice.

Retek provides product documentation in a read-only-format to ensure content integrity. Retek Customer Support cannot support documentation that has been changed without Retek authorization.

Corporate Headquarters:

Retek Inc.
Retek on the Mall
950 Nicollet Mall
Minneapolis, MN 55403
888.61.RETEK (toll free US)
+1 612 587 5000

Retek[®] Integration Bus[™] and Retek Distribution Management[™] are trademarks of Retek Inc.

Retek and the Retek logo are registered trademarks of Retek Inc.

This unpublished work is protected by confidentiality agreement, and by trade secret, copyright, and other laws. In the event of publication, the following notice shall apply:

©2002 Retek Inc. All rights reserved.

All other product names mentioned are trademarks or registered trademarks of their respective owners and should be treated as such.

Printed in the United States of America.

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 e-mail, 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.

Contact Method	Contact Information
-----------------------	----------------------------

Internet (ROCS)	www.retek.com/support Retek's secure client Web site to update and view issues
------------------------	--

E-mail	support@rettek.com
---------------	--------------------

Phone	US & Canada: 1-800-61-RETEK (1-800-617-3835) World: +1 612-587-5800 EMEA: 011 44 1223 703 444 Asia Pacific: 61 425 792 927
--------------	---

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.

Contents

Chapter 1 – Customer returns.....	1
Functional Area	1
Design Overview	1
State Diagram	1
Description of Activities	1
Create Customer Returns.....	1
Triggers	1
Message Family Manager Procedures.....	2
Public Procedures	2
Referenced Stored Procedures:.....	2
Chapter 2 – Inbound ASN	3
Functional Area	3
Design Overview	3
State Diagram	4
Description of Activities	5
Create Inbound ASN Messages	5
Delete Inbound ASN Messages	5
Triggers	5
Message Family Manager Procedures.....	5
Public Procedures	5
Referenced Stored Procedures	6
Chapter 3 – Inventory adjustments	7
Functional Area	7
Design Overview	7
State Diagram	7
Description of Activities	8
Create Inventory Adjustments	8
Triggers	8
Message Family Manager Procedures.....	8
Public Procedures	8
Referenced Stored Procedures	8

Chapter 4 – Items	9
Functional Area	9
Design Overview	9
State Diagram	10
Description of Activities	10
Triggers	11
Message Family Manager Procedures.....	11
Chapter 5 – Outbound ASN	13
Functional Area	13
Design Overview	13
State Diagram	13
Description of Activities	13
Create RTV Messages	13
Triggers	13
Message Family Manager Procedures.....	14
Public Procedures	14
Referenced Stored Procedures.....	14
Chapter 6 – Receiving.....	15
Appointment Portion	15
Receipt Portion	15
State Diagram	16
Description of Activities	17
Appointment Create.....	17
Appointment Modify	17
Appointment Delete.....	17
Appointment Detail Create.....	17
Appointment Detail Modify	18
Appointment Detail Delete.....	18
Create Receipt	18
Create Receipt Adjustment.....	18
Triggers	18
Message Family Manager Procedures.....	19
Public Procedures	19
Referenced Stored Procedures.....	19

Chapter 7 – Return to vendor	21
Functional Area	21
Design Overview	21
State Diagram	21
Description of Activities	21
Create RTV Messages	21
Triggers	21
Message Family Manager Procedures.....	22
Public Procedures	22
Referenced Stored Procedures	22
Chapter 8 – Space locations	23
Functional Area	23
Design Overview	23
State Diagram	24
Description of Activities	24
Triggers	25
Message Family Manager Procedures.....	25
Public Procedures	25
Referenced Stored Procedures	26
Chapter 9 – Stock order status	27
Functional Area	27
Design Overview	27
State Diagram	27
Description of Activities	28
Create Stock Order Info Messages	28
Triggers	28
Message Family Manager Procedures.....	28
Public Procedures	28
Referenced Stored Procedures	28

Chapter 10 – SKU OPTIMIZATION Subscription 29

SKU OPTMIZATION message structure	29
Message subscription process.....	29
Message summary	30
PL/SQL procedures	30
Public procedures	30
RDM Generic Processing	30
Primary SKU OPTMIZATION tables	30

Chapter 1 – Customer returns

Functional Area

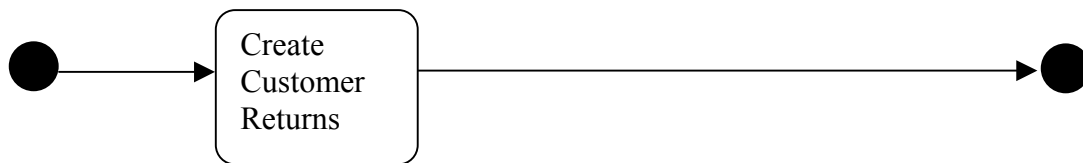
Customer Returns Publication

Design Overview

RDM is responsible for communicating Customer Returns Information to the Host System.

RDM provides the capability to process item level return information. Information to the host upon completion of the process will include: item information, unit quantity information, the RMA number, zero or more reason codes, zero or more action codes, and possibly replacement items and replacement quantities.

State Diagram



Description of Activities

Create Customer Returns

- 1 **Prerequisites:** There are no prerequisites for Customer Returns.
- 2 **Activity Detail:** There are no activity details, once the message has been processed there are no modifications.
- 3 **Messages:** When a Customer Return is created, the “Customer Returns Create” data is inserted into the Returns_Upload table. The Customer Returns Create message is a flat message containing a full snapshot of the Customer Returns at the time the Customer Returns is created.

Triggers

None

Message Family Manager Procedures

Public Procedures

GETNXT (O_status_code, O_error_msg, O_message_type, O_message, I_facility_type) – This publicly exposed procedure is typically called by a RIB publication adaptor. Its parameters are well defined and arranged in a specific order. The message type is the RIB defined short message name, the message is the XML message, and the family keys are the key for the message as pertains to the family, not all of which will necessarily be populated for all message types. Status code is one of five values; these codes come from an EAI team defined RIB_CODES package. For more discussion of the status codes, refer to the Error Handling Guidelines document.

The error text parameter contains application-generated information, such as the application's sequence number of the message that failed, and the Oracle or other error that occurred when the retrieval failed.

Referenced Stored Procedures:

CUSTOMER_RETURN_BUILD_XML (I_FACILITY_TYPE) – This procedure is responsible for retrieving the Customer Returns Information from the Returns_Upload table, creating the appropriate XML, inserting the XML into the Customer_Returns_Queue table, and marking the Returns_Upload->Pub_Status to 'S' and the Returns_Upload->Transaction_TS to the SYSDATE.

See beyond can only handle 30 character procedure names. So, the entire name of the call, package.procedure has to be within 30 characters. This only applies to public procedures, not internal, private functions and procedures.

Chapter 2 – Inbound ASN

Functional Area

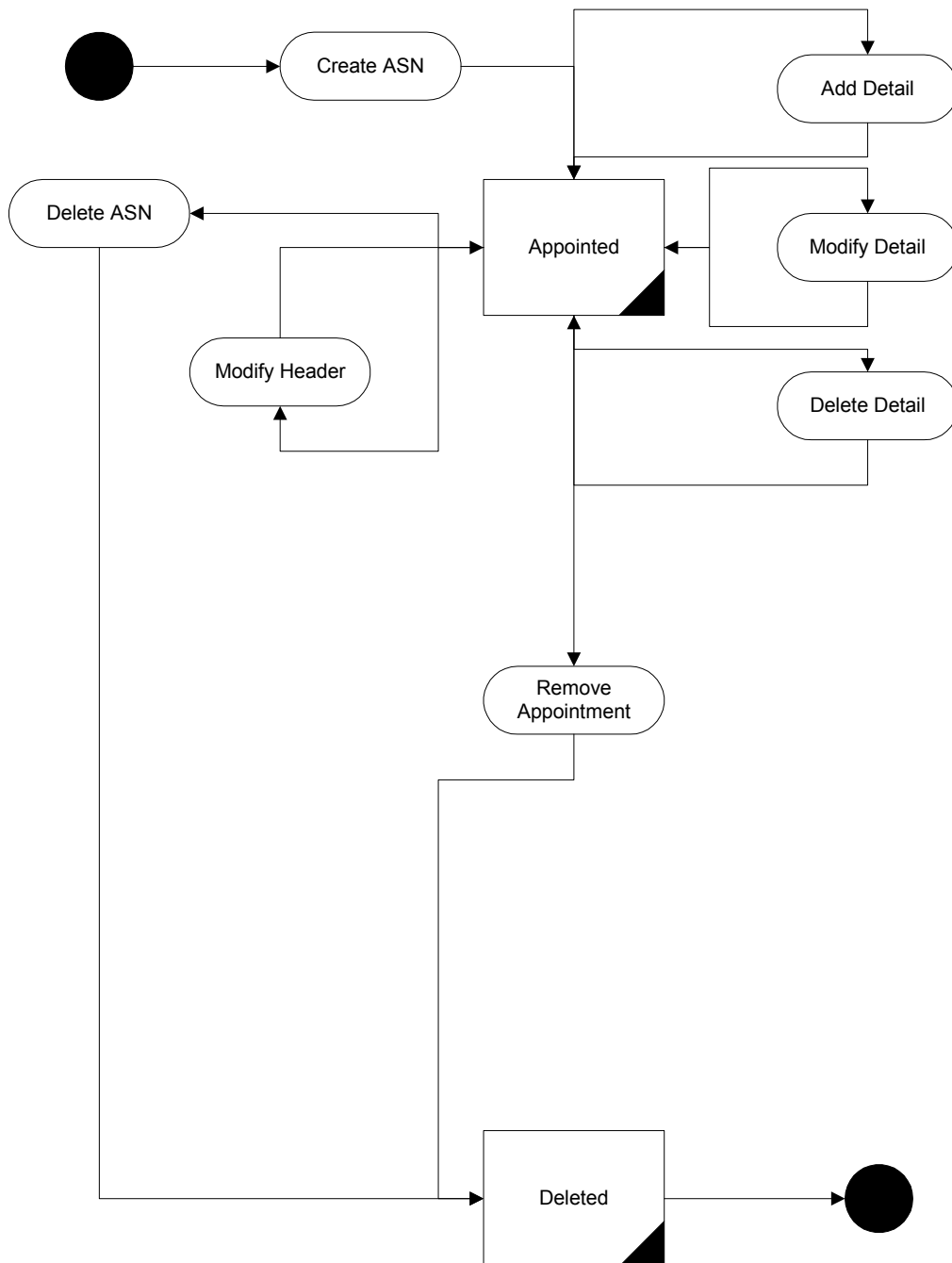
Inbound ASN Publication

Design Overview

RDM is responsible for communicating Inbound ASN Information to the Host System. Inbound ASN is defined as ASN Information originating in the RDM System. Inbound ASNs can be Container or PO Type ASNs. PO Type ASNs detail item information to be received at a unit level, not container level information. Container Type Inbound ASNs detail item information to be received at a container level. Container information includes Container ID, Destinations, Distro Number, Unit Quantity, PO and Item.

Inbound ASN messages are communicated to the Host once it has been appointed. The entire hierarchical message is sent. To modify an ASN, the ASN must not be associated to an Appointment. Once modified, the entire hierarchical message is resent.

State Diagram



Description of Activities

Create Inbound ASN Messages

- 1 **Prerequisites:** Must be ASN appointment and a valid ASN.
- 2 **Activity Detail:** Assign the ASN to an Appointment.
- 3 **Messages:** When Inbound ASN Messages are created, the “Inbound ASN Create” data is inserted into the ASN_Upload table. The Inbound ASN Create message is a hierarchical message containing a full snapshot of the Inbound ASN Message at the time the ASN was appointed.

Delete Inbound ASN Messages

- 1 **Prerequisites:** Must be ASN appointment and a valid ASN.
- 2 **Activity Detail:** Remove the ASN from the Appointment
- 3 **Messages:** When Inbound ASN Delete Messages are created, the “Inbound ASN Delete” data is inserted into the ASN_Upload table. The Inbound ASN Create message is a hierarchical message containing a full snapshot of the Inbound ASN Message at the time the ASN was appointed.

Triggers

None

Message Family Manager Procedures

Public Procedures

GETNXT (O_status_code, O_error_msg, O_message_type, O_message, I_facility_type, O_from_location, O_ASN) – This publicly exposed procedure is typically called by a RIB publication adaptor. Its parameters are well defined and arranged in a specific order. The message type is the RIB defined short message name, the message is the XML message, and the family keys are the key for the message as pertains to the family, not all of which will necessarily be populated for all message types. Status code is one of five values; these codes come from an EAI team defined RIB_CODES package. For more discussion of the status codes, refer to the Error Handling Guidelines document.

The error text parameter contains application-generated information, such as the application’s sequence number of the message that failed, and the Oracle or other error that occurred when the retrieval failed.

Referenced Stored Procedures

INBOUND_ASN_BUILD_XML(I_FACILITY_TYPE) – This procedure is responsible for retrieving the Inbound ASN Message Data from the ASN_Upload / ASN_Item_Upload / ASN_Cont_Upload / ASN_Item_Upload tables, creating the appropriate XML, inserting the XML into the Inbound_ASN_Queue table, and marking the ASN_Upload->Pub_Status to 'S' and the ASN_Upload->Transaction_TS to the SYSDATE..

Chapter 3 – Inventory adjustments

Functional Area

Inventory Adjustments Publication

Design Overview

RDM is responsible for communicating Inventory Adjustments Information to the Host System.

Inventory Adjustments can be categorized as true inventory adjustments or inventory disposition changes.

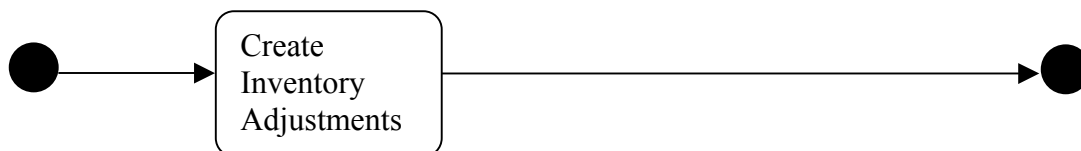
True inventory adjustments are adjusting the actual quantity of the inventory available. Inventory disposition is changing the status of the inventory (i.e. from unavailable to sell to available to sell). True inventory adjustments must always have a disposition change, however, you may have an inventory disposition without a true inventory adjustment.

Inventory Disposition statuses include:

- Receipt in Process
- Available to Sell
- Pending WIP on Inventory (WIP code will be included)
- Trouble (Trouble code will be included)
- Distributed

The user can define alternate statuses to be uploaded to the host through an RDM defined editor.

State Diagram



Description of Activities

Create Inventory Adjustments

- 1 **Prerequisites:** None.
- 2 **Activity Detail:** Inventory adjustments are created throughout the entire system as a result of normal processing.
- 3 **Messages:** When an Inventory Adjustments is created, the “Inventory Adjustments Create” data is inserted into the Inv_Adjustment_To_Upload table. The Inventory Adjustments Create message is a flat message containing a full snapshot of the Inventory Adjustments at the time the Inventory Adjustments is created.

Triggers

None

Message Family Manager Procedures

Public Procedures

GETNXT (O_status_code, O_error_msg, O_message_type, O_message, I_facility_type) – This publicly exposed procedure is typically called by a RIB publication adaptor. Its parameters are well defined and arranged in a specific order. The message type is the RIB defined short message name, the message is the XML message, and the family keys are the key for the message as pertains to the family, not all of which will necessarily be populated for all message types. Status code is one of five values; these codes come from an EAI team defined RIB_CODES package. For more discussion of the status codes, refer to the Error Handling Guidelines document.

The error text parameter contains application-generated information, such as the application’s sequence number of the message that failed, and the Oracle or other error that occurred when the retrieval failed.

Referenced Stored Procedures

INVADJ_XML_BUILDER(I_FACILITY_TYPE) – This procedure is responsible for retrieving the Inventory Adjustments Information from the InvAdj_To_Upload table, creating the appropriate XML, inserting the XML into the Inventory_Adj_Info_Queue table, and marking the Inv_Adjustment_To_Upload->Pub_Status to ‘S’ and the Inv_Adjustment_To_Upload->Transaction_TS to the SYSDATE.

Chapter 4 – Items

Functional Area

Item Publication

Design Overview

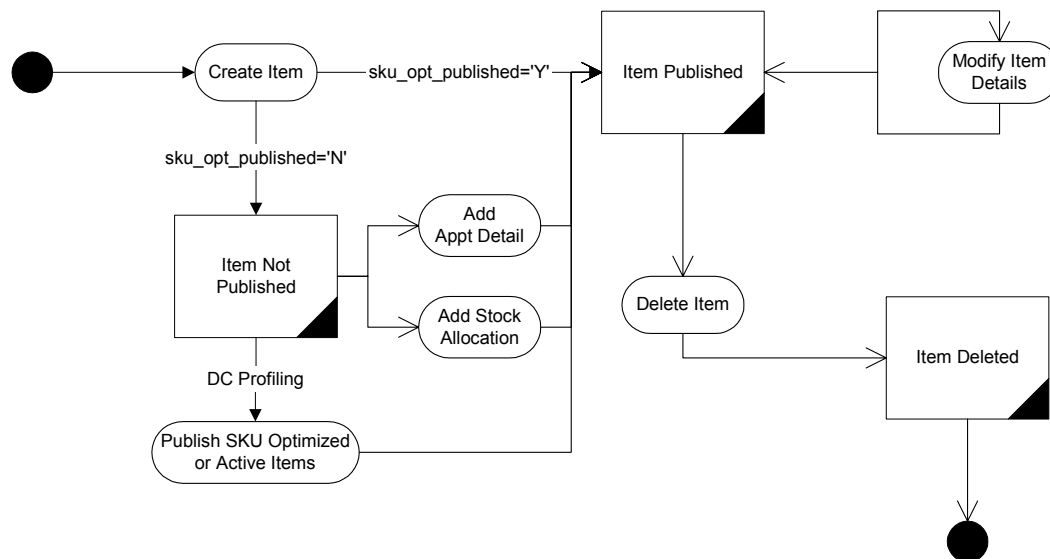
RDM is responsible for communicating items that reside in a forward picking / forward case picking locations to a third-party SKU profiling system for the purposes of warehouse optimization.

RDM item information can be published in one of two ways. The first is through a Distribution Center (DC) Profiling support function provided within RDM. The second is through a series of item related event triggers that result in the item data being sent. These event triggers include:

- Modification or Deletion of an optimized and published item/SKU (item has sku_optimized and sku_opt_published flags set to 'Y')
- Modification or Deletion of an optimized and published SKU's supplier information
- Modification or Deletion of an optimized and published SKU's supplier country information
- Modification or Deletion of an optimized and published SKU's supplier country DIM information
- Creation, Modification, or Deletion of an optimized and published SKU's association to a forward picking/case picking location
- First association of an optimized SKU to an appointment
- First association of an optimized SKU to an allocation

The information sent to the third-party system for add or modify requests would include: item header information, item supplier information, item supplier country information, and item supplier country DIM information. Deletes of item information will include only the DC Destination ID (from location) and the item id for the item being deleted.

State Diagram



Description of Activities

Create Items

- 1 **Prerequisites:** The item must be marked for SKU optimization with the sku_optimization flag in the item_master table set to 'Y'.
- 2 **Activity Detail:** Run DC Profiling from the RDM application support functions menu selecting optimized items or active items or both as the option where items exist in forward picking locations and are not yet published; Modify the item details (item, item supplier, item supplier country, item supplier country DIM, picking location) for an item that is marked for SKU optimization publication with sku_opt_published flag in the item_master table set to 'Y'; Add an item with sku_opt_published flag in the item_master table set to 'N' to an appointment detail or allocation.
- 3 **Messages:** When a SKU optimized item is published, the "Item Create" data is inserted into the ITEM_MASTER_UPLOAD table. The Item Create message is a message containing details pertaining to the item at the time the upload request was processed into an XML message.

Delete Items

- 1 **Prerequisites:** The item must be marked for SKU optimization with the sku_optimization and sku_opt_published flags in the item_master table set to 'Y'.
- 2 **Activity Detail:** Delete a published item.
- 3 **Messages:** When a published item is deleted, the “Item Delete” data is inserted into the ITEM_MASTER_UPLOAD table. The Item Master Delete message is a flat message containing the DC Destination ID for the facility where the item exists (from_location) and the item id for the item that was deleted.

Triggers

- item_master_sku_opt_trig (after update or delete on item_master)
- item_supplier_opt_trig (after update or delete on item_supplier)
- item_supp_country_opt_trig (after update or delete on item_supplier_country)
- item_supp_country_dim_opt_trig (after update or delete on item_supplier_country_dim)
- appt_detail_opt_trig (after insert on appt_detail)
- stock_allocation_opt_trig (after insert on stock_allocation)
- pick_from_loc_opt_trig (after update, insert, delete on pick_from_loc)

Message Family Manager Procedures

Public Procedures:

GETNXT (O_STATUS_CODE, O_ERROR_MSG, O_MESSAGE_TYPE, O_ITEM_ID, O_MESSAGE, I_NUM_THREADS, I_THREAD_VAL, I_FACILITY_TYPE) – This publicly exposed procedure is typically called by a RIB publication adaptor. Its parameters are well defined and arranged in a specific order. It should be noted that only the RIB uses the I_NUM_THREADS and I_THREAD_VAL parameters.

The message type is the RIB defined short message name, the item id is the item associated with the message, the message is the XML message, and the family keys are the key for the message as pertains to the family, not all of which will necessarily be populated for all message types. Status code is one of five values; these codes come from an EAI team defined RIB_CODES package. For more discussion of the status codes, refer to the Error Handling Guidelines document.

The error text parameter contains application-generated information, such as the application's sequence number of the message that failed, and the Oracle or other error that occurred when the retrieval failed.

Referenced Stored Procedures:

ITEM_BUILD_XML (I_FACILITY_TYPE) – This procedure is responsible for retrieving the Item information from the ITEM_MASTER_UPLOAD table, creating the appropriate XML, inserting the XML into the ITEM_MASTER_QUEUE table, and marking the ITEM_MASTER_UPLOAD records as published (by setting pub_status to 'S' and transaction_ts to the SYSDATE).

- Seebeyond can only handle 30 character procedure names. So, the entire name of the call, package.procedure has to be within 30 characters. This only applies to public procedures, not internal, private functions and procedures.

Chapter 5 – Outbound ASN

Functional Area

Outbound ASN Publication

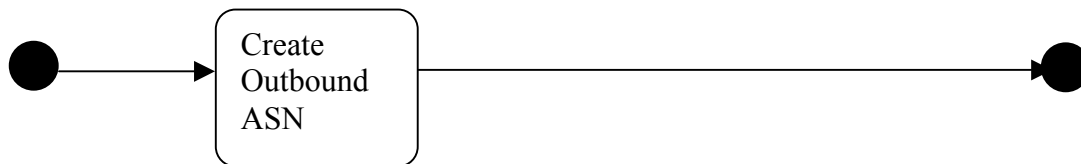
Design Overview

RDM is responsible for communicating Outbound ASN Information to the Host System.

Outbound ASN Information consists of ASN Information, BOL Number, Manifest Information including Trailer and Carrier, Container Information including Items, Unit Quantities, Container ID, Destination and Distro Information.

An outbound ASN is generated for a distinct Shipping Trailer/Destination.

State Diagram



Description of Activities

Create RTV Messages

- 1 **Prerequisites:** Trailer must be in a Shipped Status.
- 2 **Activity Detail:** None
- 3 **Messages:** When Outbound ASN Messages are created, the “Outbound ASN Create” data is inserted into the BOL_To_Upload table. The Outbound ASN Create message is a hierarchical message containing a full snapshot of the Outbound ASN Message at the time the shipment was created.

Triggers

None

Message Family Manager Procedures

Public Procedures

GETNXT (O_status_code, O_error_msg, O_message_type, O_message, I_facility_type) – This publicly exposed procedure is typically called by a RIB publication adaptor. Its parameters are well defined and arranged in a specific order. The message type is the RIB defined short message name, the message is the XML message, and the family keys are the key for the message as pertains to the family, not all of which will necessarily be populated for all message types. Status code is one of five values; these codes come from an EAI team defined RIB_CODES package. For more discussion of the status codes, refer to the Error Handling Guidelines document.

The error text parameter contains application-generated information, such as the application's sequence number of the message that failed, and the Oracle or other error that occurred when the retrieval failed.

Referenced Stored Procedures

OUTBOUND_ASN_BUILD_XML(I_FACILITY_TYPE) – This procedure is responsible for retrieving the Outbound ASN Message Data from the BOL_To_Upload table and selecting from the Stock_Order, Container, Container_Item, and Manifest tables, creating the appropriate XML, inserting the XML into the Outbound_ASN_Queue table, and marking the BOL_To_Upload->Pub_Status to 'S' and the BOL_To_Upload->Transaction_TS to the SYSDATE.

Chapter 6 – Receiving

Appointment Portion

RDM is responsible for communicating Appointment Information to the Host System. Appointment information consists of the Appointment Number, PO Information, Item Details, Scheduled Units and as well as ASN Information when related to an ASN.

Appointment messages are transmitted to the Host once the Appointment has been scheduled. Once scheduled, Appointment messages will be communicated at the addition, modification, or deletion of a detail, a modification of the header information such as arrival time, or at the Open, Close, and Deletion of the appointment.

Receipt Portion

RDM is responsible for communicating Receipt Information to the Host System.

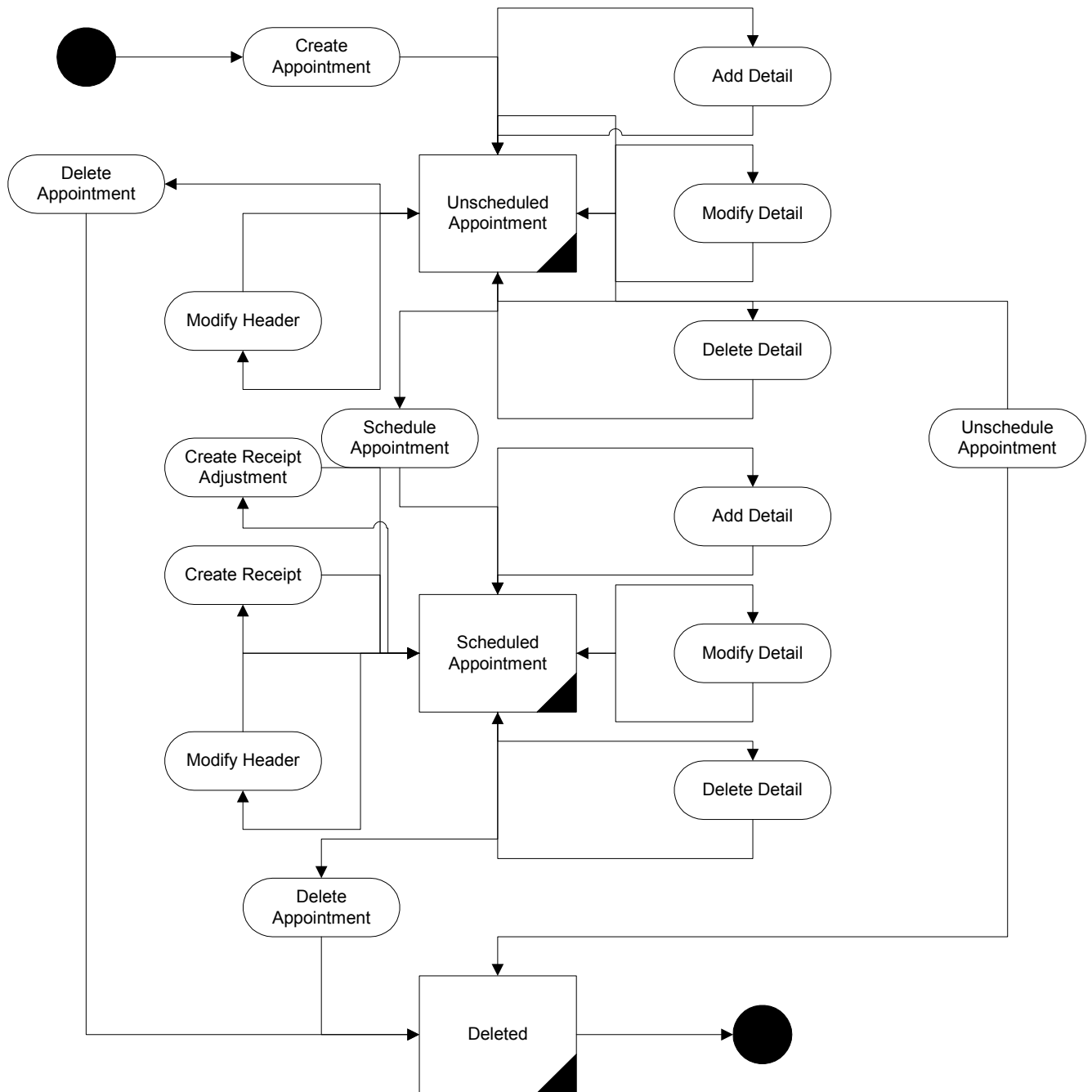
Receipt information is at the container level. It is uploaded to the host from the container level or when an appointment is closed depending on an RDM system parameter. Receipt Info Upload will include appointment information, item number, ASN number if applicable, quantity, purchase order number, disposition changes, and type of receipt.

Receipt types include:

- Initial Receipt
- Adjustment to an already uploaded receipt

Both types of receipts contain the same information listed above.

State Diagram



Description of Activities

Appointment Create

- 1 **Prerequisites:** A valid door and trailer must exist to create an appointment.
- 2 **Activity Detail:** None
- 3 **Messages:** When Appointment Create Messages are created, the “Appointment Create” data is inserted into the Appt_Header_To_Upload / Appt_Detail_To_Upload table. The Appointment Create message is a hierarchical message containing a full snapshot of the Appointment Message at the time the first appointment detail record is added.

Appointment Modify

- 1 **Prerequisites:** Appointment must exist.
- 2 **Activity Detail:** Change the Door, Appointment Time Stamp.
- 3 **Messages:** When Appointment Modify Messages are created, the “Appointment Modify” data is inserted into the Appt_Header_To_Upload table. The Appointment Modify message is a flat message containing a full snapshot of the Appointment Modify Message at the time the appointment status is changed.

Appointment Delete

- 1 **Prerequisites:** Appointment must exist and be in the appropriate status
- 2 **Activity Detail:** Cascade deletes to any associated detail tables.
- 3 **Messages:** When Appointment Delete Messages are created, the “Appointment Delete” data is inserted into the Appt_Header_To_Upload table. The Appointment Delete message is a flat message containing the Appointment Number that was deleted.

Appointment Detail Create

- 1 **Prerequisites:** Valid appointment header and a valid PO and Item. If related to an ASN, the ASN must be valid.
- 2 **Activity Detail:** None
- 3 **Messages:** When Appointment Detail Create Messages are created, the “Appointment Detail Create” data is inserted into the Appt_Header_To_Upload / Appt_Detail_To_Upload table. The Appointment Detail Create message is a flat message containing a full snapshot of the Appointment Detail Create Message at the time the appointment detail is created.

Appointment Detail Modify

- 1 **Prerequisites:** Appointment detail record must exist in the appropriate status.
- 2 **Activity Detail:** Appropriate checks made to maintain data integrity.
- 3 **Messages:** When Appointment Detail Modify Messages are created, the “Appointment Detail Modify” data is inserted into the Appt_Header_To_Upload / Appt_Detail_To_Upload table. The Appointment Detail Modify message is a flat message containing a full snapshot of the Appointment Detail Modify Message at the time the appointment detail was modified changed.

Appointment Detail Delete

- 1 **Prerequisites:** Appointment detail record must exist in the appropriate status.
- 2 **Activity Detail:** None
- 3 **Messages:** When Appointment Detail Delete Messages are created, the “Appointment Detail Delete” data is inserted into the Appt_Header_To_Upload / Appt_Detail_To_Upload table. The Appointment Detail Delete message is a flat message containing a full snapshot of the Appointment Detail Delete Message at the time the appointment detail was created.

Create Receipt

- 1 **Prerequisites:** Valid appointment must exist.
- 2 **Activity Detail:** Receipt of Container creates a Receipt to upload
- 3 **Messages:** When a receipt is created, the “Receipt Create” data is inserted into the Receipt_To_Upload table. The Receipt Create message is a flat message containing a full snapshot of the receipt at the time the receipt is created.

Create Receipt Adjustment

- 1 **Prerequisites:** Container must be received and the initial receipt upload must have been sent.
- 2 **Activity Detail:** Each container is individually checked using RDM functionality.
- 3 **Messages:** When a receipt adjustment is created, the “Receipt Adjustment” data is inserted into the Receipt_To_Upload table. The Receipt Adjustment message is a flat message containing a full snapshot of the receipt adjustment at the time the receipt adjustment is created.

Triggers

None

Message Family Manager Procedures

Public Procedures

GETNXT (O_status_code, O_error_msg, O_message_type, O_message, I_facility_type, O_from_location, O_appt_nbr) – This publicly exposed procedure is typically called by a RIB publication adaptor. Its parameters are well defined and arranged in a specific order. The message type is the RIB defined short message name, the message is the XML message, and the family keys are the key for the message as pertains to the family, not all of which will necessarily be populated for all message types. Status code is one of five values; these codes come from an EAI team defined RIB_CODES package. For more discussion of the status codes, refer to the Error Handling Guidelines document.

The error text parameter contains application-generated information, such as the application's sequence number of the message that failed, and the Oracle or other error that occurred when the retrieval failed.

Referenced Stored Procedures

APPOINTMENT_BUILD_XML(I_FACILITY_TYPE) – This procedure is responsible for retrieving the Appointment Family Message Data from the Appt_Header_To_Upload / Appt_Detail_To_Upload tables, creating the appropriate XML, inserting the XML into the Appointment_Queue table, and marking the Appt_Header_To_Upload->Pub_Status to 'S' and the Appt_Header_To_Upload->Transaction_TS to the SYSDATE. This procedure also calls the Receipt_XML_Builder procedure before processing Close messages. This is done to process all receipts prior to closing the appointment.

RECEIPT_XML_BUILDER(I_FACILITY_TYPE, I_APPT_NBR DEFAULT NULL) – This procedure is responsible for retrieving the Receipt and Receipt Adjustment Information from the Receipt_To_Upload table, creating the appropriate XML, inserting the XML into the Receipt_Queue table, and marking the Receipt_To_Upload->Pub_Status to 'S' and the Receipt_To_Upload->Transaction_TS to the SYSDATE.

In addition, this procedure determines when the Receipt/Receipt Adjustment records are created. RDM users have the ability to upload Receipt/Receipt Adjustments at a Container Level or at an Appointment Level. For Container Level, Receipt/Receipt Adjustments are sent as they are created. For Appointment Level, Receipt/Receipt Adjustments are sent once the Appointment has been received.

Chapter 7 – Return to vendor

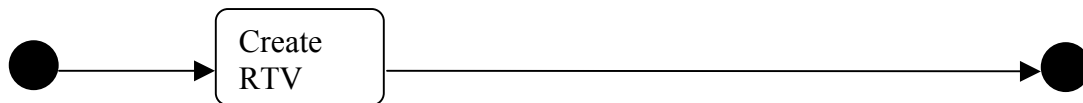
Functional Area

RTV Publication

Design Overview

RDM is responsible for communicating RTV Information to the Host System. RTV information is sent to the Host when the DC chooses to return merchandise to the Vendor. Information includes Return Authorization Numbers, Vendor Information including address, Item and Quantity Information and Inventory Disposition Statuses.

State Diagram



Description of Activities

Create RTV Messages

- 1 **Prerequisites:** Container must be in the appropriate status.
- 2 **Activity Detail:** All pending WIPs and Troubles are cleared prior to RTV.
- 3 **Messages:** When RTV Messages are created, the “RTV Create” data is inserted into the Stock_Order_Info_Upload table. The RTV Create message is a flat message containing a full snapshot of the RTV Messages at the time the inventory was affected.

Triggers

None

Message Family Manager Procedures

Public Procedures

GETNXT (O_status_code, O_error_msg, O_message_type, O_message, I_facility_type) – This publicly exposed procedure is typically called by a RIB publication adaptor. Its parameters are well defined and arranged in a specific order. The message type is the RIB defined short message name, the message is the XML message, and the family keys are the key for the message as pertains to the family, not all of which will necessarily be populated for all message types. Status code is one of five values; these codes come from an EAI team defined RIB_CODES package. For more discussion of the status codes, refer to the Error Handling Guidelines document.

The error text parameter contains application-generated information, such as the application's sequence number of the message that failed, and the Oracle or other error that occurred when the retrieval failed.

Referenced Stored Procedures

RTV_BUILD_XML(I_FACILITY_TYPE) – This procedure is responsible for retrieving the RTV Message Data from the Inv_Adjustment_To_Upload table where the adjustment_reason_code = 90, creating the appropriate XML, inserting the XML into the RTV_Queue table, and marking the Inv_Adjustment_To_Upload->Pub_Status to 'S' and the Inv_Adjustment_To_Upload->Transaction_TS to the SYSDATE.

Chapter 8 – Space locations

Functional Area

Space Locations Publication

Design Overview

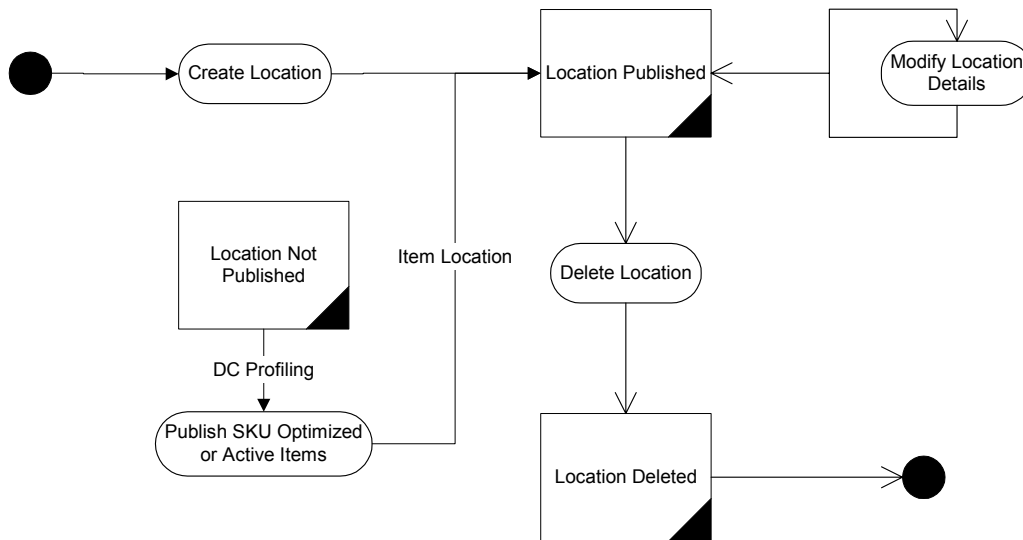
RDM is responsible for communicating Forward Unit and Forward Case Picking Location (FPL and FCPL respectively) Information to a third-party SKU profiling system for the purposes of warehouse optimization.

RDM FPL/FCPL information can be published in one of two ways. The first is through Distribution Center (DC) Profiling support function provided within RDM. The second is through a series of location related event triggers that result in the location data being sent. These event triggers include:

- Creation or Deletion of a new unit or case picking (a.k.a. published) location
- Updates to a published location's type, zone, status, put-away sequence, or pick sequence
- Deletion or Update of information pertaining to a published location's type. Such information includes description, container capacity, length, width, height, max standard units, volume type, unit cost, and whether or not the location is for unit or case picking
- Update of information pertaining to a published location's zone, such as description, pick priority, region or work area
- Insert, Delete, or Update of an item to/from a picking location when the item has been SKU optimized and is assigned for SKU publishing. In this case, the location information for the picking location where the item has been assigned/unassigned is sent

The information sent to the third-party system on an add or modify will include: location id, zone information, items assigned to that location for picking, and location type information such as whether the location is unit or case pick, length, height, etc. Deletes of location information will include only the DC Destination ID (from location) and the location id for the location being deleted.

State Diagram



Description of Activities

Create Space Locations

- 1 **Prerequisites:** The location type for the created/modified location must be a unit or case picking location, meaning the unit_pic_loc_flag = 'Y' or case_pick_loc_flag = 'Y'.
- 2 **Activity Detail:** Run DC Profiling from the RDM application support functions menu **selecting** optimized items or active items or both as the option when items exist in forward picking locations that are not yet published; Create a new location; Modify the location information for a location that has the sku_opt_published flag = 'N'.
- 3 **Messages:** When a location is created or modified, the "Space Locations Create" data is inserted into the LOCATION_UPLOAD table. The Space Locations Create message is a flat message containing details pertaining to the location at the time the upload request was processed into an XML message.

Modify Space Locations

- 1 **Prerequisites:** The location type for the created/modified location must be a unit or case picking location, meaning the `unit_pic_loc_flag = 'Y'` or `case_pick_loc_flag = 'Y'`. In addition, the location must be published, meaning the `sku_opt_published` flag on the location must = 'Y'.
- 2 **Activity Detail:** Modify location information or add/delete/change an item assigned to a published location (`sku_opt_published` flag = 'Y').
- 3 **Messages:** When a published location or an item assigned to a published location is modified, the “Space Locations Modify” data is inserted into the `LOCATION_UPLOAD` table. The Space Locations Modify message is a flat message containing details pertaining to the location at the time the upload request was processed into an XML message.

Delete Space Locations

- 1 **Prerequisites:** The location type for the created/**modified** location must be a unit or case picking location, meaning the `unit_pic_loc_flag = 'Y'` or `case_pick_loc_flag = 'Y'`. In addition, the location must be published, meaning the `sku_opt_published` flag on the location must = 'Y'.
- 2 **Activity Detail:** Delete a published location (`sku_opt_published = 'Y'`).
- 3 **Messages:** When a published location is deleted, the “Space Locations Delete” data is inserted into the `LOCATION_UPLOAD` table. The Space Locations Delete message is a flat message containing the DC Destination ID for the facility where the location exists (`from_location`) and the location id for the location that was deleted.

Triggers

- `location_opt_trig` (before update, insert, delete on location)
- `loc_type_opt_trig ()`
- `zone_opt_trig ()`
- `pick_from_loc_opt_trig` (after update, insert, delete on `pick_from_loc`)

Message Family Manager Procedures

Public Procedures

GETNXT (O_STATUS_CODE, O_ERROR_MSG, O_MESSAGE_TYPE, O_LOCATION_ID, O_MESSAGE, I_NUM_THREADS, I_THREAD_VAL, I_FACILITY_TYPE) – This publicly exposed procedure is typically called by a RIB publication adaptor. Its parameters are well defined and arranged in a specific order. It should be noted that only the RIB uses the `I_NUM_THREADS` and `I_THREAD_VAL` parameters.

The message type is the RIB defined short message name, the location id is the location associated with the message, the message is the XML message, and the family keys are the key for the message as pertains to the family, not all of which will necessarily be populated for all message types. Status code is one of five values; these codes come from an EAI team defined RIB_CODES package. For more discussion of the status codes, refer to the Error Handling Guidelines document.

The error text parameter contains application-generated information, such as the application's sequence number of the message that failed, and the Oracle or other error that occurred when the retrieval failed.

Referenced Stored Procedures

SPACE_LOCATION_BUILD_XML (I_FACILITY_TYPE) – This procedure is responsible for retrieving the Space Locations information from the LOCATION_UPLOAD table, creating the appropriate XML, inserting the XML into the LOCATION_QUEUE table, and marking the LOCATION_UPLOAD records as published (by setting pub_status to 'S' and transaction_ts to the SYSDATE).

Seebeyond can only handle 30 character procedure names. So, the entire name of the call, package.procedure has to be within 30 characters. This only applies to public procedures, not internal, private functions and procedures.

Chapter 9 – Stock order status

Functional Area

Stock Order Info Publication

Design Overview

RDM is responsible for communicating Stock Order status Information to the Host System.

RDM will generate stock order status information upon detection of any changes to a stock order.

These statuses include:

- Successful Insert
- Successful Delete
- Store Reassign
- Detail Selected
- Detail Unselected
- Pick Created
- Pick Deleted
- Return to Stock
- Cartonization Complete
- Cartonization Reversed
- Expired Stock Order
- No Inventory

Information includes distro number, distro type, item information and quantities, and status.

State Diagram



Description of Activities

Create Stock Order Info Messages

- 1 **Prerequisites:** Valid distro number.
- 2 **Activity Detail:** Generate throughout the system per normal use of the system.
- 3 **Messages:** When Stock Order Info Messages are created, the “Stock Order Info Create” data is inserted into the Stock_Order_Info_Upload table. The Stock Order Info Create message is a flat message containing a full snapshot of the Stock Order Info Messages at the time the inventory was affected.

Triggers

None

Message Family Manager Procedures

Public Procedures

GETNXT (O_status_code, O_error_msg, O_message_type, O_message, I_facility_type) – This publicly exposed procedure is typically called by a RIB publication adaptor. Its parameters are well defined and arranged in a specific order. The message type is the RIB defined short message name, the message is the XML message, and the family keys are the key for the message as pertains to the family, not all of which will necessarily be populated for all message types. Status code is one of five values; these codes come from an EAI team defined RIB_CODES package. For more discussion of the status codes, refer to the Error Handling Guidelines document.

The error text parameter contains application-generated information, such as the application’s sequence number of the message that failed, and the Oracle or other error that occurred when the retrieval failed.

Referenced Stored Procedures

SOS_XML_BUILDER(I_FACILITY_TYPE) – This procedure is responsible for retrieving the Stock Order Info Message Data from the Stock_Order_Info_Upload table, creating the appropriate XML, inserting the XML into the Stock_Order_Info_Queue table, and marking the Stock_Order_Info_Upload->Pub_Status to ‘S’ and the Stock_Order_Info_Upload->Transaction_TS to the SYSDATE.

Chapter 10 – SKU OPTIMIZATION Subscription

SKU Optimization Information is used to receive recommended slotting information from a third-party item optimization vendor.

This family of messages is considered to be Foundation Data. Foundation Data indicates that the data is used as the basis for building other data models and is routed to every RDM installation in the enterprise.

SKU OPTIMIZATION message structure

The SKUOptm family of messages consists of one message type: create. This message is used to populate the RDM TASK_QUEUE table. This message is in a single node structure

Message subscription process

The following is a description of the SKU OPTIMIZATION message subscription process:

- 1 The RDM SKU Optimization adapter recognizes that a message with the SKU Optimization-specific name (i.e. SKUOPTMCRE) exists on the RIB.
- 2 The adapter calls the public PL/SQL procedure to “consume” the message. The public “consume” procedure is named:
 - RDMSUB_SKU_OPTMCRE.CONSUME
- 3 The public procedure calls a set of generic consuming procedures and packages. There are four generic functions:
 - MESSAGE_OBJECT
 - MESSAGEPRETABLE
 - MESSAGEPOSTTABLE
 - MESSAGEPOSTMESSAGE

These procedures and packages perform all of the parsing, additional processing and validation against the database.

The list of procedures and packages described in this overview is not exhaustive. For a more detailed view of them, see the RDM Generic Subscription Messages in this guide.

Message summary

All Sku Optimization messages belong to the SKUOptm message family. The following table lists the single message that RDM subscribes to by its short name along with the document type definition (DTD) used for validation and parsing of the data during the message subscription process and the mapping documents that describes the data contained in the message.

Message Name(API)	Type (DTD)	Mapping Document
SKUOPTMCRE	SKUOPTMDesc.dtd	Map_SKUOPTMDesc.xls

Consult the Retek 10 Integration Guide to view the DTD and mapping document that pertains to the message in which you are interested.

PL/SQL procedures

This section describes the procedures and functions listed in the earlier “Message subscription process” section.

Public procedures

The RIB calls the public consume procedure that, in turn, passes the message to RDM’s SKU OPTIMIZATION adapter:

RDMSUB_SKUOPTMCRE.CONSUME-This procedure accepts an XML file in the form of an Oracle CLOB data type from the RIB. This message contains a SKU Optimization create message.

RDM Generic Processing

The list of procedures and packages used for generic processing perform all of the parsing, additional processing, and validation against the database. For a more detailed view of the generic procedures and packages, see the RDM Generic Subscription Messages in this guide.

Primary SKU OPTMIZATION tables

The following descriptions are for the primary tables in RDM that hold SKU Optimization data:

- TASK_QUEUE

Detailed descriptions of these tables are in the RDM 10.0 Data Model document.