

Retek® Distribution Management 10.3



Operations Guide – Volume 1: Functional Overviews



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[®] Distribution Management[™] is a trademark of Retek Inc.

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

©2003 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 – System administration	1
Introduction	1
Creating the rdmusr account on the operating system.....	1
Creating operating system accounts for all other users	3
Creating users in Oracle	4
Creating user accounts.....	4
Radio frequency operations	4
Operating system functions	5
Printer queues	5
Operating system scheduled jobs	6
Daemons	9
File management (directories)	9
System parameters.....	10
Chapter 2 – DBA administration module	33
Overview	33
Procedures	33
Open the Monitoring and Administration menu.....	33
Display locks on tables	34
Display table information	35
Display tablespace information	36
Display rollback information.....	37
Display index information	39
Display sequences information.....	40
Display the error log	41
View error log details	42
Delete error log records	43
Print the error log report	44
Chapter 3 – RDM RIB components	45
Overview	45
Acronyms	45
Terms	46
RDM Message subscription process.....	47
RDM Message publication process	48

Subscription components	49
Vendor subscription.....	49
Location subscription	51
Item subscription	52
UDA subscription	55
Differentiator subscription.....	57
Purchase order subscription.....	59
Inbound work order subscription.....	61
Inbound ASN subscription	63
Stock Order subscription	65
Outbound work order subscription	67
Pending returns subscription	69
Publish components.....	71
Inbound ASN publish	71
Appointments/receipts publish	74
Stock order status publish.....	79
Outbound ASN publish	81
Inventory adjustments publish.....	83
Inventory balance upload.....	85
Customer returns publish.....	86
Return to vendor publish	88
Streamsoft components	90
Space locations publish	90
SKU optimization subscription	92
Item warehouse publish.....	93
Retek Labor Management (RLM) components	95
Transaction publish	95
Location publish	96
Equipment publish.....	97
Equipment class publish	99
Chapter 4 – Subsystem interfaces.....	101
Batch file formats	101
Unit pick system files.....	102
Allocation data download.....	102
Inbound carton download	102
Process UPS upload.....	103
Pick by light interface	104
Files and directories.....	104
Download transactions	105
Upload transactions	109

Sortation subsystem interface.....	113
Files and directories.....	113
Download transactions	114
Upload transactions	115
Manifest mailing system	116
Files and directories.....	116
Views.....	117
Tables	118
Kewill shipping system interface	121
Triggers.....	121
Packages	122
Tables	122
Rapistan socket interface.....	123
Triggers.....	123
Packages	123
Tables	124
Third party routing interface	126
Packages	126
Download transactions	127
Upload transactions	128

Chapter 1 – System administration

Introduction

Each system user must have a unique logon ID to the operating system, Oracle, and Retek Distribution Management. This section describes how to create each of these accounts.

Creating the rdmusr account on the operating system

When Retek Distribution Management is first installed, the group rdm is created and the users rdmhost and rdmusr are created. The rdmusr user's home directory is typically /home/rdmusr and the rdmhost user's home directory is typically /home/rdmhost.

Add to the rdmusr profile script (.profile.) the following and set the protection to 740 (rwxr----):

```
ORACLE_SID=rdm;  
export ORACLE_SID  
TWO_TASK=rdm;  
export TWO_TASK  
TERM=vt220;  
export TERM  
ORACLE_TERM=vt220;  
export ORACLE_TERM  
ORACLE_HOME=/u01/app/oracle/product/dev6i;  
export ORACLE_HOME  
LD_LIBRARY_PATH=$ORACLE_HOME/lib;  
export LD_LIBRARY_PATH  
TNS_ADMIN=/u01/app/oracle/product/9i/network/admin;  
export TNS_ADMIN  
PATH=$ORACLE_HOME/bin:./:../:/usr/bin:/etc:/usr/sbin:/usr/ucb:/sbin:/usr/local/bin;  
export PATH  
TMPDIR=/tmp;  
export TMPDIR  
FORMS60_PLSQLV1_NAME_RESOLUTION=YES;  
export FORMS60_PLSQLV1_NAME_RESOLUTION  
FORMS60_OUTPUT=/tmp;  
export FORMS60_OUTPUT
```

```
FORMS60_PATH=/u01/app/rdm/bin;
export FORMS60_PATH
FORMS60_TERMINAL=/u01/app/rdm/bin;
export FORMS60_TERMINAL
REPORTS60_PATH=/u01/app/rdm/bin;
export REPORTS60_PATH
REPORTS60_TERMINAL=$REPORTS60_PATH;
export REPORTS60_TERMINAL
REPORTS_OUTPUT=/u01/app/rdm/reports;
export REPORTS_OUTPUT
RDM_BIN=/u01/app/rdm/bin;
export RDM_BIN
NLS_DATE_FORMAT="mm/dd/rr";
export NLS_DATE_FORMAT
NLS_LANG=American_America.UTF8
export NLS_LANG
RDMUSR=rdm username
    export RDMUSR
RDMPWD=rdm password
    export RDMPWD
menu.sh
exit
```

Add to the rdmhost profile script (.profile.) the following and set the protection to 740 (rwxr-----):

```
ORACLE_SID=rdm;
export ORACLE_SID
ORACLE_HOME=/u01/app/oracle/product/9i;
export ORACLE_HOME
TERM=vt220;
export TERM
ORACLE_TERM=vt220;
export ORACLE_TERM
TNS_ADMIN=/u01/app/oracle/product/9i/network/admin;
export TNS_ADMIN
LD_LIBRARY_PATH=$ORACLE_HOME/lib;
    export LD_LIBRARY_PATH
PATH=$ORACLE_HOME/bin:./:../:/usr/bin:/etc:/usr/sbin:
/usr/ucb:/sbin:/usr/local/bin;
```

```

export PATH
RDMUSR=rdm username
export RDMUSR
RDMPWD=rdm password
export RDMPWD
DOWNLOAD_DIR=/u01/app/rdm/hostcomm/download;
export DOWNLOAD_DIR
UPLOAD_DIR=/u01/app/rdm/hostcomm/upload;
export UPLOAD_DIR
SORTATION_DIR=/u01/app/rdm/hostcomm/sortation;
export SORTATION_DIR

```

Note: The value for fields shown in bold italics (above) must be set to the value appropriate for your installation.

The value of TERM is limited to the following choices:

- 1 ibm3151
- 2 vt220

Creating operating system accounts for all other users

Create the user account in the operating system. The default shell should be ksh. The home directory should be /home/rdmusr, or whatever directory was assigned to the rdmusr. This prevents each user from having an individual home directory and makes the maintenance of the .profile easier.

Use whatever security measures are appropriate for your installation. You may use the operating system's security provisions for password expiration.

Retek Distribution Management enforces access control over the System screens regardless of the operating system security measures.

Creating users in Oracle

Use Oracle's facility, such as Server Manager (svrmgrl), to create accounts in Oracle.

Set the default tablespace to USERS and the temporary tablespace to TEMP. The user's name must be the same as the account name on the operating system. The user's password must be the same as the user's name.

Grant the new user the wms_user role. It has all the sufficient privileges to operate on all application tables.

Use the following SQL syntax to create new user accounts in Oracle:

```
create user rdmusr identified by <password>
default tablespace USERS
temporary tablespace TEMP;
grant wms_user to rdmusr;
```

Creating user accounts

Add the user in Retek Distribution Management, using the User Table Editor screen. The password you specify in the system can be the same as or different than the user's operating system password.

Radio frequency operations

When RF users are ready to begin work, they log into the operating system using their own account name and password. The .profile is executed, which sets the environment and executes the Retek Distribution Management application. The application takes the user's operating system account name as the Oracle account and password, and then starts the login to the system as that user. The first screen displayed is the Retek Distribution Management copyright. The login screen follows it. The user must fill in the username, password and facility ID.

Operating system functions

This section explains how to set up print queues, describes the cron jobs and daemons, and gives instructions for file management.

Printer queues

Create print queues in the operating system for reports and labels. Retek Distribution Management does not embed any printer-specific commands in jobs being sent to report printers. Label printer queues are typically defined as standard ASCII devices.

The names of the printer queues are specified on the System Parameter screen. Please also refer to the System Parameters section for more discussion of those parameters. The names of the parameters are listed here:

- 1 pick_audit_queue
- 2 pick_label_queue
- 3 pick_package_queue
- 4 recv_audit_queue
- 5 recv_label_queue
- 6 recv_receipt_queue
- 7 reprint_label_queue
- 8 ship_bol_queue
- 9 ship_label_queue
- 10 unit_pick_lbl_queue

Operating system scheduled jobs

This table describes the programs that should be run periodically to remove obsolete data from the system, to schedule locations for cycle counting, and to close appointments that are now reconciled. You should run these programs using the operating system facility (cron) for scheduling jobs for unattended operation.

In the table, the name of the program to run is listed under the column heading "Name." The programs are in the \$RDM_BIN directory. For each routine, a system parameter exists that specifies the number of days of data to retain. These parameters are maintained on the System Parameter editor, which is described in the Retek Distribution Management User Guide.

rdmhost user must submit the jobs. The user's cron should first execute the .profile to set the environmental variables.

Name	Purpose	Parameters	Frequency
capacity_replen.sh	Releases on-hold replenishments destined for Forward Pick locations	facility_id	Every several minutes
dc_view.sh	Refreshes data in the DC VIEW and DC UTILIZATION tables	facility_id	Once daily
insert_distribution_queue.sh	Works with the Automate wave processing	facility_id	Every several minutes or time that meets customer needs.
inv_bal_upload_b.sh	Create an inventory balance upload file for each facility id, part of the facility type	facility_id	Once daily
maintain_wave_stats.sh	Updates wave statistics		Every 15 minutes
print_reprints.sh	Reprints container labels or prints shipping container labels		Every 5 minutes
print_reprints_monarch.sh	Reprints container labels or prints shipping labels for Monarch 9820.		Every 5 minutes
print_tickets.sh	Prints or reprints tickets		Every 5 minutes

Name	Purpose	Parameters	Frequency
print_wave_labels.sh	Prints or reprints picking labels		Every 10 minutes
print_wave_labels_monarch.sh	Prints or reprints Monarch 9820 picking labels		Every 10 minutes
purge_activity_based_cost.sh	Purges all old records from the Activity_Based_Cost table	facility_id	Once daily
purge_activity_log.sh	Purges all aged records from the Activity_Log table	facility_id	User Configurable
purge_appointments_b.sh	Cleans the Appointment and related tables.	Appointment_window & appt_purge_days	Once daily
purge_history.sh	Purges data from the Container History table	facility_id and time_stamps	User Configurable
purge_labor_prod_b.sh	Cleans the Labor Productivity table.	Purge_lab_prod_days	Once daily
purge_manifest_b.sh	Cleans the Manifest and related tables.	manifest_purge_days	Once daily
purge_route_data.sh	Cleans the Route Date and Route Dest tables.	facility_id	Once daily or more frequently if necessary for table size control
purge_rtv_b.sh	Cleans the Return to Vendor table.	Purge_rtv_days	Once daily
purge_uploads_b.sh	Cleans the upload tables.	Upload_purge_days	Once daily
purge_vendor_trouble_.sh	Cleans the vendor trouble history tables.	Trouble_purge_days	Once daily
report_monitor.sh	Prints reports and lavelns initiated from the GUI system	None	Every 5 minutes
run_distribution.sh	Matches inventory to allocation requests, creates pick directives and prints picking packages (if applicable)		Manually or every 15 minutes

Name	Purpose	Parameters	Frequency
schedule_cycle_count_b.sh	Schedules cycle counts (SS) for the DC.	Cycle_count_period	Once daily
schedule_rop_distribution.sh	Schedules a Re-Order Point distribution run.		Every 5 minutes
update_daily_wh_stats_b.sh	Updates statistics for daily warehouse activities		Once daily
unreconciled_appt_monitor.sh	Closes any unreconciled appointments that have had all labels scanned or nulled.	None	Hourly

Daemons

One daemon process must be run continuously. It should be run by the user rdmhost from the \$RDM_BIN directory. The Calling Syntax includes the parameter -s, which is the sleep time in seconds: how often the daemon should wake up and look for inducted or diverted cartons. A typical value is between 10 and 30 seconds. In the Calling Syntax, <user_name/password> refers to an Oracle user and password.

Name	Description	Calling Syntax
Read_divert_data	Loads the sorter intake table from a data file.	read_divert_data <username/password> <facility_id>-s[n]

File management (directories)

Discusses permissions and any file cleanup (purging needed for each directory).

Directory	Path	Purpose	Perm	Purging
Base Directory	\$RDM	This is the base directory that other directories will branch from	775	None
Reports	\$RDM/reports	Temporary holding area for reports (line and label). Any report sent to file remains here.	777	Occasional (weekly)
Host Download	\$DOWNLOAD_DIR	Temporary holding area for files to download and log files.	775	Occasional (weekly)
Host Upload	\$UPLOAD_DIR	Temporary holding area for files to upload.	775	Occasional (weekly)
Sortation	\$SORTATION_DIR		775	Occasional (weekly)
BIN	\$RDM/bin	Holds all executables.	755	None
INSTALL	\$RDM_ADMIN/create	Holds files used to build the system.	755	None

System parameters

Each facility in the DC has its own set of system level parameters. You can view and modify these in the System Parameter screen.

This table gives the name of each parameter, briefly explains its purpose and how it is used, and gives the allowable entry type, as described in this list:

Activity: Activity code, found in service_standards table.

Dest ID: Destination ID, found in Ship Destination table.

Fixed: Cannot be modified by the user.

Item ID: Item ID, found in Item Master table.

Location: Location ID, found in Location table.

Location Type: Location Type found in the Loc_type table

Number: Numeric value.

Queue: Printer (line or label) queue.

Text: Free form text.

Time: Valid time (24 hour format - HH:MI).

WIP: WIP code, found in the wip_codes table.

Y/N: “Y” = Yes, “N” = No.

The allow_user_edit column indicates which parameters are user modifiable.

Name	Purpose	Type	Allow User Edit
3 rd _party_routing	Checked ‘Y’(yes) when using an FTP interface to a 3 rd party routing package.	Y/N	Y
DC_dest_ID	Destination ID of the DC. Must be in the Ship Dest table. Used in reports (for DC return address) and to show what containers are stock (dest_ID=DC).	Dest ID	Y
LTC	Unit Pick System Code associated to the RF Unit Picking (Stationary SKU)(LTC and ltc code refer to the same operation of Less Than Case picking)	Text	Y
MM_Cycle_Count_Priv	User Privilege to execute Cycle Count on Manually Marked (MM) Location	Number	Y

Name	Purpose	Type	Allow User Edit
PPS	Unit Pick System Code associated to a Paperless Picking System(PPS and pps code refer to the same paperless picking system)	Text	Y
PPS_flag	Indicates whether PPS is turned on. Used in the distribution and picking processes.	Y/N	Y
TASK_OPT	Specifies the ordering of assigned tasks.	TEXT or Location_id	Y
able_to_ship_level	Security level to enable the F9 ship key in the shipping form.	Number	Y
active_ovrszd_putwy	Default putaway plan for an oversized item (when no putaway plan specified in Item Master)	Putaway Plan	Y
adjust_pick	Enable the F7 adjust key on the RF picking screens.	Y/N	Y
ahl_log	Log Activity History Log. 0: No AHL Logging 1: AHL Logging through SQL insert 2: AHL Logging through Oracle Queues	0, 1, 2	Y
allow_rtn_replace	When set to 'Y', RDM allows item replacement and displays a screen to the user where an alternate item is entered to replace the returned one.	Y/N	Y
allow_trble_putaway	Allows the Putaway screen to complete the putaway of a container that has a Trouble Code associated with it.	Y/N	Y
apply_qa_wip	Determines if a QA WIP needs to be applied	Y/N	Y
appointment_window	The number of days (past and future) to allow appointments to be active. Used in the Schedule Appointment screen and purge_appointments_b.sh.	Number	Y
appt_bulk_def	Sets bulk flag default on the Appointment Detail screen.	Y/N	Y

Name	Purpose	Type	Allow User Edit
appt_purge_days	Number of days after closure to purge an appointment. Used in purge_appointments_b.sh.	Number	Y
ari_enabled	Is Retek ARI Application installed with RDM?	Y/N	Y
assortment_wip_code	WIP code applied when inbound container has an assortment item. Parent Item with child SKUs.	WIP	Y
ats_calc_incl_dist	Y = Distributed inventory will be kept in the Available to Sell bucket. (Please discuss with Retek contact)	Y/N	N
auto_induct	When set to 'Y', groups assigned to the first pack wave have the Active Pick flag set to 'Y', indicating that this pack wave is to be staged in the UPS for picking. If put_to_order is enabled, allocation data is sent to the UPS for only those allocations deemed active within the UPS. If put to destination is enabled, all allocations are downloaded at one time.	Y/N	Y
Autopack	Assigned name to the Autopack Sorter	Text	Y
back_order_flag	Indicates whether to retain stock orders when the inventory is exhausted. Used in the distribution process.	Y/N	Y
best_before_wip	Used to automatically apply a WIP code to a container requiring a best before date (perishable indicator set in Item Master)	WIP	Y
bld_mix_dest_sku_pal	When set to Y, building of mixed destination pallets (Distributed) is allowed	Y/N	Y
break_by_wip_con	When set to 'Y', the distribution process creates separate Master Pick Labels for each group of WIP codes for conveyable cartons.	Y/N	Y

Name	Purpose	Type	Allow User Edit
break_by_wip_non_con	When set to 'Y', the distribution process creates separate Master Pick Labels for each group of WIP codes for non-conveyable cartons.	Y/N	Y
carton_store_putwy	Default putaway plan for a single container	Text	Y
clear_user	Y = clears the user name from the 'C' pick allowing another user to proceed with operation. 'N' = user name is still associated with the pick and that individual must finish the operation.	Y/N	Y
company_nbr	Company number to send to PPS.	Number	Y
consolidate_pend_wip	When set to 'Y', RDM allows the consolidation of WIP codes, when building pallets.	Y/N	Y
container_format	Indicates that the container identifier number is compliant with UCC128 or is generic with embedded destination ID.	UCC128 or default	Y
cs_rsv_loc_type	User Defined location type for case reserve	Loc Type	Y
cs_rsv_priority	Priority used in distribution to pull merchandise from case	Number	Y
Cubiscan	Checked 'Y' (yes) when using a TCP/IP connection to a Cubiscan device.	Y/N	N
cycle_count_period	Number of days to cycle count the entire DC. Used in schedule_cycle_count_b.sh.	Number	Y
cycle_count_type	Defines how the DC wants to count inventory, either by item, location or zone. Used when schedule cycle count runs in cron (System Scheduled Cycle Count)	Text	Y
def_bulk_replen_res	Sets the number of Bulk Replenishment resources to use to display the Wave Duration on the wave planning screens.	Number	Y

Name	Purpose	Type	Allow User Edit
def_bulk_resources	Sets the number of Bulk resources to use to display the Wave Duration on the wave planning screens.	Number	Y
def_cont_replen_res	Sets the number of Container Replenishment Pick resources to use to display the Wave Duration on the wave planning screens.	Number	Y
def_cont_resources	Sets the number of Container Pick resources to use to display the Wave Duration on the wave planning screens.	Number	Y
def_cont_replen_res	Sets the number of Container Replenishment Pick resources to use to display the Wave Duration on the wave planning screens.	Number	Y
def_random_putaway	Default putaway plan for random replenishment	Text	Y
def_unit_replen_res	Sets the number of Unit Pick Replenishment resources to use to display the Wave Duration on the wave planning screens.	Number	Y
def_unit_resources	Sets the number of Unit Pick resources to use to display the Wave Duration on the wave planning screens.	Number	Y
def_work_day_end	Default working day end. Used in Working Days Editor.	Time	Y
def_work_day_start	Default working day start. Used in Working Days Editor.	Time	Y
default_carton_group	Carton group used in cartonization if none is defined for the item.	Text	Y
default_cc_plan	default cycle count plan to be set during item master download if none is specified.	Text	Y
default_dc_cont_type	Default container type used for a pallet in FCP (Forward Case Picking)	Container Type	Y
default_kitting_wip	WIP code when creating an item that is defined as a kit	WIP	Y

Name	Purpose	Type	Allow User Edit
default_order_level	Type of saved query for order selection.	FULL – Every line in query has to match. ORDER – If any line matches, RDM will show all distro lines. LINE – Only distros that match display.	Y
default_order_type	Determines how stock orders that are downloaded are processed. WAVE uses the pre-defined Shipping Schedule and proceeds without intervention. AUTOMATIC does not require destinations to have a pre-defined Shipping Schedule, but does proceed without intervention. MANUAL allows intervention by selecting orders to be included in a wave. PO – allocation of merchandise is tied to a specific PO. PREDIST – allocations that have predistributed merchandise.	WAVE, AUTOMATIC, MANUAL, PO and PREDIST	Y
default_putaway	default putaway plan to be set during item master download if none is specified	Text	Y
default_trailer_cube	Default size of a trailer. Used in the Schedule Appointment screen when a new trailer is scheduled. Used to calculate and display the percentage filled of a trailer on the Shipping Status.	Number	Y
default_ups	Default Unit Pick System code for Item Master download	Text	Y

Name	Purpose	Type	Allow User Edit
delete_pfl	When set to 'Y', and unit quantity in the 'from location' is 0, RDM will delete the location record once the merchandise is moved out. User override is provided on the FPL Move screen.	Y/N	Y
display_item_id	Used in the multi-item UPC functionality. If set to 'Y', item information, which matches the UPC code, will display when the item_id is scanned. When set to 'N', only the UPC code displays in the field.	Y/N	Y
distrib_unfin_wip	When set to 'Y', RDM allows allocation of merchandise from a pallet that has unfinished WIP codes associated with it.	Y/N	Y
distribute_partial	When set to 'Y', RDM will process partial distribution of a dye lot. The maximum amount of a single dye lot will be distributed even if only a partial fulfillment of the order. If 'N', the distribution is skipped.	Y/N	Y
drop_off_convey	Suggested drop-off location for pallet and case picking for conveyable merchandise.	Location	Y
dynamic_random_slot	determine whether distribution should create a random slot for active picking when needed.	Y/N	Y
enable_kitting	When set to 'Y', the Distribution process builds Kit Build directives for Master Items that have a Stock Allocation but no Inventory to satisfy the order.	Y/N	Y
entry_limit	Maximum adjustment quantity on a unit basis per user.	Number	Y
exceed_capacity	allows chutes to be overfilled during the waving process	Y/N	Y
exception_cont_type	default container type to be used during cartonization if no defined container types will hold items	Container Type	Y

Name	Purpose	Type	Allow User Edit
exceptions_stage	area specified in building (location) where exception packages should be sent for consolidation	Location	Y
fcpl_random_act_stg	Staging location for replenishments to random forward case locations.	Location	Y
fcpl_random_dest_id	Dest id for replenishments to random forward case locations	Dest id	Y
first_ctn_seen	WIP code to be applied to first carton during receiving	WIP	Y
first_time_sku	WIP code to be applied to first time SKU containers during receiving	WIP	Y
fixed_replen_wave	When set to 'Y', RDM groups all replenishment picks into Wave 1. When set to 'N' RDM associates replenishment picks with the wave that originated the need.	Y/N	Y
fpl_replen_dest_id	Destination identifier used for replenishing of Forward Pick Locations when replenishment method is Preplanned.	Dest ID	Y
fstsku_bypass_fl	Indicates to conveyor system to weigh or not weigh a carton with first time SKU WIP applied.	Y/N	Y
generate_rma	When set to 'Y', the distribution process generates a unique number that is assigned per container. This generation process happens after the 'pick-to' containers are split out based on volumetric data. When set to 'N', RMA numbers are not generated.	Y/N	Y
gift_card_wip	Defined WIP code denoting containers that require the insertion of a specialized gift card.	WIP	Y
gift_w_wip	Defined WIP code for gift wrapping	WIP	Y
group_picks_active	Determine how distribution should cartonize active picks	Y/N	Y
hold_first_time_sku	WIP applied to all like containers for items where one container has first time SKU WIP applied	WIP	Y

Name	Purpose	Type	Allow User Edit
hot_replen_putaway	When set to 'Y', Putaway looks for Unit Replenishment opportunities.	Y/N	Y
hot_replen_recvg	When set to Y, receiving allocation process looks for Unit Replenishment opportunities. When set to N, receiving allocation process functions as normal.	Y/N	Y
in_transit_loc	Location of containers in process. Used in Move, Putaway, and Picking screens.	Location	Y
interface_tcp_flag	Indicates the use of a TCP/IP interface with a conveyor system. (Future RDM Use)	Y/N	Y
interleaved_cc	When set to 'Y', RDM suggests a location for system scheduled cycle count after a putaway operation. When set to 'N', Putaway and Cycle Count task are not interleaved.	Y/N	N
kitting_activity_code	Activity code associated with kitting against which statistics are collected.	Activity	Y
labeled_picking	When set to 'Y', RDM generates a picking label packet and a report. When set to 'N', RDM assumes labelless picking and only generates a report.	Y/N	Y
labeled_receiving	When set to 'Y', RDM generates a receiving label packet and a report. When set to 'N', RDM assumes labelless receiving and only generates a report.	Y/N	Y
labeled_reserve	When set to 'Y', RDM tracks each container in reserve storage with a separate identifying label. When set to 'N', only master containers in reserve are labeled.	Y/N	Y
labeled_tote	If set to 'Y' (yes), labels for Unit picks will be printed even when labeled_picking = 'N'.	Y/N	Y

Name	Purpose	Type	Allow User Edit
load_sequencing	When set to 'Y', RDM sorts picks with respect to the defined route/destination load sequence. When set to 'N', RDM sorts according to distro number sequence.	Y/N	Y
log_interface_error	Determines whether RDM Interface APIs log an error using the log_oracle_error function when an error occurs. Note: This must be set to N in an enterprise/SeeBeyond environment because of Oracle distributed processing and support for AUTONOMOUS TRANSACTIONS	Y/N	Y
ltc_code	Unit Pick System Code associated to the RF Unit Picking (Stationary SKU)(LTC and ltc code refer to the same operation of Less Than Case picking)	Text	Y
ltc_staging_loc	Location id for replenishment drop-off going into LTC.	Location	Y
manifest_mail_flag	Checked Y(es) means a third party manifest mailing system is being used.	Y/N	Y
manifest_purge_days	Number of days after shipping to purge a manifest. Used in purge_manifest_b.sh.	Number	Y
max_group_units	used with group picks active. Numeric values that sets max number of units to be allocated to one group	Number	Y
max_wave_nbr	Maximum wave number allowed to be maintained in the distribution screens.	Number	Y
max_wave_rows	Maximum number of orders/rows that may be retrieved from a specific query. This number is used when the user does not include the max number as part of a query.	Number	Y

Name	Purpose	Type	Allow User Edit
min_auto_wave	The lowest wave number that can be used by RDM when assigning orders. The system assigns any orders retrieved by a specific query to the first wave with the status of AVAIL, type of MANUAL and greater than or equal to the min_auto_number.	Number	Y
mixed_dest_id	Destination ID where containers holding merchandise for different destinations are sent for separation..	Dest ID	Y
mixed_wip_stage_loc	Location identifier at which containers with different WIP codes are staged for separation.	Location	Y
mm_allow_distrib	determines whether or not distribution is allowed to distribute from manually marked locations	Y/N	Y
multi_open_manifest	When set to 'Y', indicates that multiple destinations can be actively loaded into a single trailer simultaneously.	Y/N	Y
multi_sku_wip	WIP code applied to inbound container that contain more than one container item record	WIP	Y
nbr_cartons_pallet	Max number of cartons per pallet, in putaway logic.	Number	Y
nbr_items_pallet	Max number of items per pallet, in putaway logic.	Number	Y
nbr_skus_per_pallet	Max number of SKUs per pallet, in putaway logic.	Number	N
oflow_replen_dest_id	Dest id for replenishments to Overflow forward picking locations.	Dest id	Y
order_line_number	'Y' setting indicates that orders are being tracked at the order line level	Y/N	Y
rder_set_stage	location in facility where outbound cartons will be directed to have order sets printed.	Location	Y
order_status_upload	Y if order status information will be uploaded to the host	Y/N	Y

Name	Purpose	Type	Allow User Edit
outb_ship_label	‘Y’ = outbound cartons/pallets are directed to a PRINT and APPLY location for the application of a shipping label. ‘N’ = outbound cartons/pallets are shipped with the generic picking label.	Y/N	Y
outbound_qa_wip	WIP code to apply for cartons assigned to Outbound QA	WIP	Y
override	Y = Allows user to override the suggested location in reserve storage. N = Denies the ability to override a suggested location in reserve storage	Y/N	Y
pack_lane_stage	staging location where outbound orders are sent to be packed	Location	Y
pack_wave_stage	staging location where cartons are sent to await induction into a unit sorter	Location	Y
pallet_flow_loc_type	user defined location type for pallet flow reserve	Location Type	Y
pallet_flow_priority	priority used during distribution to pull merchandise from case reserve	Number	Y
pallet_rsv_loc_type	user defined location type for pallet reserve	Location Type	Y
pallet_rsv_priority	priority used during distribution to pull merchandise from case reserve	Number	Y
pallet_store_putwy	default putaway plan to be used for items that do not have a putaway plan specified	Text	Y
parse_publish	Y = when publishing XML messages, parse the message prior to sending.	Y/N	Y
password_expire	Number of days since the last password change; forces users to change their password.	Number	Y
password_old	Number of days since the last password change; suggests that users change their password.	Number	Y

Name	Purpose	Type	Allow User Edit
pbl_pick_to_reserve	When set to 'Y', causes the system to generate a distribution detail record to download to the Pick-By-Light system, which will cause the excess units to be re-boxed and returned to inventory. This parameter is applicable only when the pps_round_up flag is set to 'N'.	Y/N	Y
pbl_replen_dest_id	Default destination assigned for replenishment to the PPS system.	Dest ID	Y
pend_first_time_sku	Cartons of an item on a receipt to be held on the receiving dock until the first time SKU WIP is removed	Y/N	Y
pick_audit_queue	Line printer queue where the Pick Audit List prints.	Queue	Y
pick_by_loc_flag_con	When set to 'Y', RDM is picking by location and permits mixing of conveyable cartons of varying destinations onto a single pallet during Container Picking. When set to 'N', RDM is picking by destination and does not permit mixing of conveyable cartons of varying destinations onto a single pallet during Container Picking.	Y/N	Y
pick_by_loc_flag_non	When set to 'Y', RDM is picking by location and permits mixing of non-conveyable cartons of varying destinations onto a single pallet during Container Picking. When set to 'N', RDM is picking by destination and does not permit mixing of non-conveyable cartons of varying destinations onto a single pallet during Container Picking.	Y/N	Y
pick_existing	Determines whether or not to include the inbound quantity associated to a forward pick location when determining amount of units available for picking	Y/N	Y
pick_label_queue	Label printer queue where the pick labels will print.	Queue	Y

Name	Purpose	Type	Allow User Edit
pick_label_set	Determines the scheme used to generate the Container Identifier.	UCC128 or default	Y
pick_printer_type	Printer description	Text	Y
pnad_isd_lead_time	Pick not after date/In store date lead time.	Number	Y
pps_code	Unit Pick System Code associated to a Paperless Picking System(PPS and pps code refer to the same paperless picking system)	Text	Y
pps_drop_off_loc	Location where containers bound for PPS are dropped off.	Location	Y
pps_pickup_loc	Location at which the system picks up cartons packed by PPS	Location	Y
pps_round_up	When set to 'Y', the distribution process will increase (round up) the distribution evenly across the destinations to consume the excess. When set to 'N', the process will not exceed the requested quantity. The parameter pbl_pick_to_reserve is applicable only when the pps_round_up flag is set to 'Y'.	Y/N	Y
pre_manifest_bol	Default sequence number for pre manifest BOLs. Used in the Conveyor Cutoff and Ship Trailer screens.	Number	Y
prepack_wip_dest	Internal dest id for containers with the WIP code of Prepack.	Dest id	Y
preplan_unit_replen	When set to 'Y', unit picks are planned to replenish the entire wave's needs during the Distribution Process. When set to 'N', RDM assumes the use of Re-order Point (or Max/Min) Replenishment.	Y/N	Y
print_and_apply	Location where print and apply labels occurs	Location	Y
pts_ctn_max_days	Number of days before open Put To Store carton is flagged.	Number	Y

Name	Purpose	Type	Allow User Edit
pts_loc_type	Default location type for Put To Store	Location Type	Y
purge_RTV_days	The number of days after a Return to Vendor to purge an RTV. Used in purge_rtv_b.sh.	Number	Y
purge_act_based_cost	The number of days after activity based cost figures are calculated to purge the ABC data. Used in purge_activity_based_cost.sh	Number	Y
purge_lab_prod_days	The number of days after activities to purge labor productivity data. Used in purge_labor_prod_b.sh.	Number	Y
putaway_stage_loc	Default location suggested for a two-step putaway.	Location	Y
qa_bypass_fl	Indicates if sortation system should weigh an inbound carton that has a QA WIP applied	Y/N	Y
qa_to_active	Allow cartons with QA WIPs to be sent directly to active. Works in conjunction with hot_replen_recvg	Y/N	Y
qa_wip_code	WIP code to be applied to cartons that need an inbound QA	WIP	Y
qc_audit_queue	Printer queue where the Quality Audit prints.	Text	Y
qlty_activity_code	Activity Code for the Quality Audit operation.	Activity	Y
quality_wip_code	Defined WIP code applied to cartons during the Prereceiving Process to mark for Quality Audit.	WIP	Y
quarantine_wip_code	WIP code designating quarantine.	WIP	Y
random_active_stage	Staging location where replenishment containers for random active are placed.	Location	Y
random_repln_dest_id	Destination ID for Random Active locations	Dest Id	Y
rdm_debug_close	Application debugging flag	Y/N	N
rdm_debug_dir00	Application debugging directory	File Directory	N

Name	Purpose	Type	Allow User Edit
rdm_debug_dir01	Application debugging filename	File Name	N
reassign_wip	Defined WIP code that reassigns a group of containers from a single destination to another single destination.	WIP	Y
receipt_level	Determines the level at which the receipt uploads will be processed. Valid values are 'A'ppointment and 'C'ontainer	Text	Y
recv_audit_queue	Line printer queue where the Receiving Audit List will print.	Queue	Y
recv_label_queue	Printer queue where the Receiving Label Package is printed.	Queue	Y
recv_label_set	Format of the Container Identifier used when generating Receiving Labels.	UCC128 or default	Y
recv_printer_type	Name of the printer and the size label stock that matches the label definition.	Text	Y
recv_receipt_queue	Label printer queue where the receiving labels print.	Queue	Y
reg_pack_chute	chute designator for regular packing chutes	Text	Y
replenishment_level	When a unit picking location is expected to drop below this value multiplied by its units capacity, the system generates a replenishment pick. Used in the distribution process and when preplanned_unit_replen scp parameter = 'Y'..	Number	Y
reprint_label_queue	Printer queue where the labels generated by the Reprint/Null Labels screen are printed.	Queue	Y
reprint_printer_type	Name of the printer and the size label stock that matches the label definition.	Text	Y
reserve_ovrszd_putwy	Putaway plan for oversized cartons	Text	N

Name	Purpose	Type	Allow User Edit
retain_label_file	Indicates whether the label print file that was sent to the printer is kept in the \$ RDM/reports directory.	Y/N	Y
reticketing_wip_code	Defined WIP code denoting containers that need new retail price tags.	WIP	Y
return_replace_code	Defined WIP codes denoting a returned container that holds items requiring replacement.	WIP	Y
return_to_vendor_loc	Location ID that identifies the location where return to vendor processing takes place.	Location_id	Y
return_wip	Defined WIP codes that denote a returned container.	WIP	Y
returns_location	Location ID that identifies the location where returns processing takes place.	Location_id	Y
rf_asn_position	Determines the starting position for display of the ASN Number on the RF screens	Number	Y
rf_item_position	Determines the starting position for display of the item id on the RF screens	Number	Y
routing_purge_days	Number of days to hold routing information in routing tables before it is purged.	Number	Y
ship_bol_queue	Line printer queue where the Bill of Lading prints.	Queue	Y
ship_label_queue	Printer queue where shipping labels print.	Queue	Y
ship_logical_pallet	Logical Pallet in Shipping	Text	Y
ship_printer_type	Type of printer at which shipping labels are printed.	Text	Y
ship_seal_required	Flag that allows the DC to specify if the seal number is required when shipping.	Y/N	Y

Name	Purpose	Type	Allow User Edit
ship_stage	Default location used the CSR table when creating routes using 3 rd party routing package. This location would be used if the routing package cannot supply a staging location.	Location	Y
ship_unique_seal	‘Y’ indicates that each seal number must be unique.	Y/N	Y
ship_warn_close	Flag that indicates that the user will receive shipping warnings when closing the trailer if yet to be loaded merchandise still exists in the DC.	Y/N	Y
ship_warn_pt_b	‘Y’ = User receives a warning of the existence of pending Bulk (b) picks for one or more of the destinations associated to the trailer being processed.	Y/N	Y
ship_warn_pt_c	‘Y’ = User receives a warning of the existence of pending Container (c) picks for one or more of the destinations associated to the trailer being processed.	Y/N	Y
ship_warn_pt_cb	‘Y’ = User receives a warning of the existence of pending Forward Case Pick (FCP) to Belt to Outbound (CB) picks for one or more of the destinations associated to the trailer being processed.	Y/N	Y
ship_warn_pt_cf	‘Y’ = User receives a warning of the existence of pending FCP to Pallet to Outbound (CF) picks for one or more of the destinations associated to the trailer being processed.	Y/N	Y
ship_warn_pt_u	‘Y’ = User receives a warning of the existence of pending Unit (U) picks for one or more of the destinations associated to the trailer being processed.	Y/N	Y

Name	Purpose	Type	Allow User Edit
ship_warn_pts	'Y' = User receives a warning of the existence of closed PTS cartons still residing in the PTS area for one or more destinations associated to the trailer being processed.	Y/N	Y
ship_warn_ship	Flag that indicates that the user will receive shipping warnings when shipping the trailer if yet to be loaded merchandise still exists in the DC.	Y/N	Y
ship_warn_status_d	'Y' = User receives a warning of the existence of containers with a 'D'istributed status for one or more of the destinations associated to the trailer being processed.	Y/N	Y
ship_warn_status_p	'Y' = User receives a warning of the existence of containers with a 'P'ending Pick status for one or more of the destinations associated to the trailer being processed.	Y/N	Y
ship_warn_status_t	'Y' = User receives a warning of the existence of containers with a 'T'roubled status for one or more of the destinations associated to the trailer being processed.	Y/N	Y
singles_sorter_group	Sorter group defined for Singles processing	Text	Y
smtp_domain	The domain name for emailing the BOL and Manifest.	Text	Y
smtp_host	The host name for emailing the BOL and Manifest.	Text	Y
smtp_port	The port for emailing the BOL and Manifest.	Number	Y
ticketing_wip_code	WIP code to apply for ticketing processing.	WIP	Y
tote_stage	Staging location where totes are built to pallet using the Build Tote Pallet screen.	Location	Y

Name	Purpose	Type	Allow User Edit
trans_wip_in_to_out	Determines whether any inbound work orders associated to a PO/Item should be applied to cross-docked containers and processed as outbound work orders	Y/N	Y
trouble_purge_days	Number of days to retain on file for Container and Appointment History.	Number	Y
ucc_container_app_id	Specific_business ID for use with UCC128 label generation.	Text	Y
ucc_container_org_id	Value to use when creating an UCC128-compliant carton serial number.	Text	Y
ucc_manufacturer_id	Value to use when creating an UCC128-compliant carton serial number.	Text	Y
ulc	RDM User License Code	Text	N
unit_block_dist_flag	When set to 'Y', RDM distributes units in Block. Block indicates that shortages are borne by the lower priority destinations. When set to 'N', RDM distributes units in Round Robin. Round Robin spreads shortages proportionally among all destinations. Used in the distribution process for LTC locations only.	Y/N	Y
unit_pick_lbl_queue	Printer queue where packing slip prints. Used in the Select Orders screen for unit picks only.	Queue	Y
unknown_item	Item ID of unknown merchandise. Used in the Build Container screen.	Item ID	Y
unknown_rma	Generic ID for returned containers that do not include the original RMA number.	Text	Y
unknown_vendor	Generic ID for vendor.	Text	Y
unlocated_location	Location of lost containers, those that cannot be found during a cycle count. Used in the Count Location screen.	Location	Y

Name	Purpose	Type	Allow User Edit
upld_convert_inv_adj	When set to 'Y', RDM uploads an inventory adjustment when converting inventory to inventory during startup.	Y/N	Y
upload_purge_days	The number of days after an upload to purge the upload data. Used in purge_uploads_b.sh.	Number	Y
use_item_dimensions	Used in cubing for forward case distribution. 'Y' = Item Master dimensions used. 'N' = Item Supplier dimensions used.	Y/N	Y
usps_priority_code	Default Service Code for the Pack Slip	Text	Y
usps_service_code	Default Route for the Pack Slip	Text	Y
va_wip_code	WIP code used for when Vendor Assurance	WIP	Y
vas_error_capture	Y = captures user ID for VAS errors. Pertains to auditing of outbound containers in a Consumer Direct world.	Y/N	Y
version_number	Number of the System version.	Fixed	Y
virtual_distro	Distro number assigned to unreconciled store orders from a Unit Pick System	Text	Y
weigh_wip_code	Defined WIP code that assigns a WIP code to weigh merchandise that has a catch weight.	WIP	Y
work_on_saturday	When set to 'Y', RDM sets Saturday as a working day. Used in the Working Days Editor.	Y/N	Y
work_on_sunday	When set to 'Y', RDM sets Sunday as a working day. Used in the Working Days Editor.	Y/N	Y
wt_round_robin_post	Applicable when unit_block_dist_flag parameter is set to Y. Indicates if weighted round robin (based on percent allocated) should be used.	Y/N	Y

Name	Purpose	Type	Allow User Edit
xzone_pick	When set to 'Y', the distribution process creates pick across multiple zones for the same distro. When set to 'N', cross-zone picking, for the same distro, is denied	Y/N	Y

Chapter 2 – DBA administration module

Overview

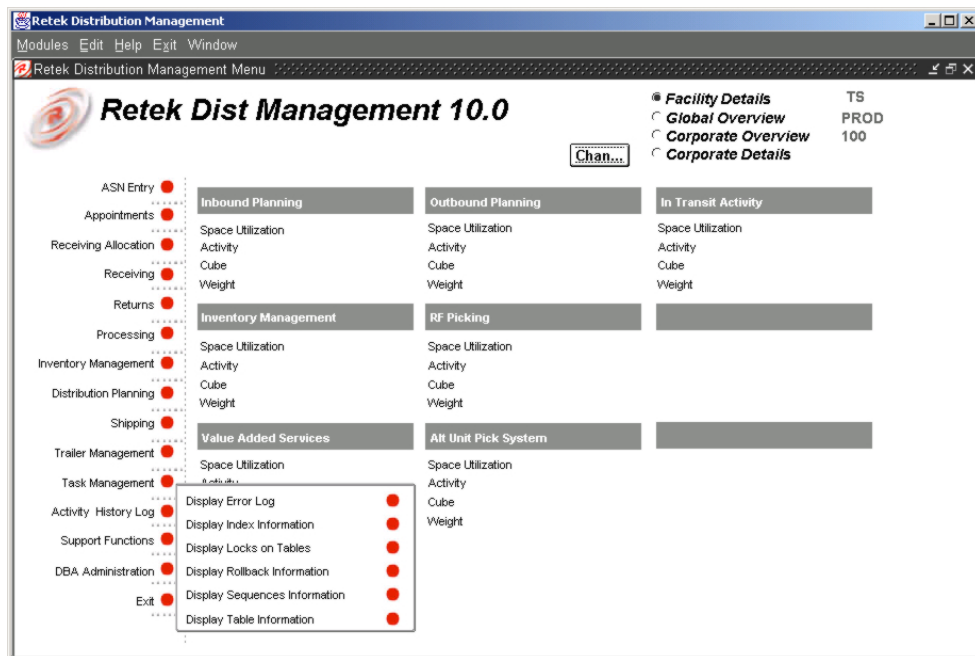
The DBA Administration module allows the DBA Administrator to monitor database information such as table locks, tablespace, indices, and errors.

The Administration section describes how to display locks on tables, table information, tablespace information, rollback information, index information, sequence information, and the error log.

Procedures

Open the Monitoring and Administration menu

- 1 Select the DBA Administration menu.



Monitoring and Administration menu

Display locks on tables

- 1 On the DBA Administration menu, select the Display Locks on Tables option.
- 2 The Table Locks Monitoring screen is displayed.

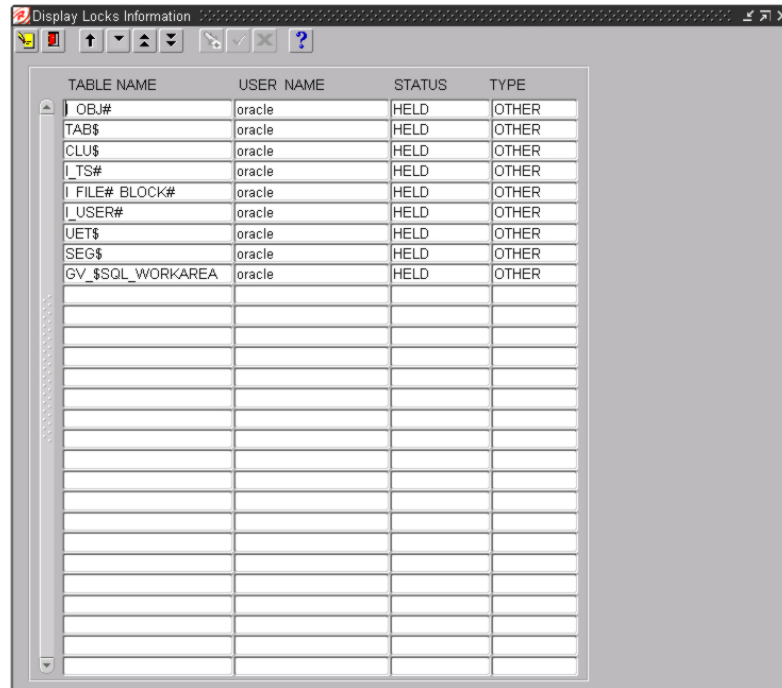


Table Locks Monitoring Screen

- 3 Click the **Refresh** button to view the table locks up-to-the-minute status.

Display table information

You can use the Display Table Information option to display specific table information.

Note: You can monitor the number of extents to detect table growth. A large extent value indicates possible table fragmentation. If the number of extents approaches the maximum, the table should be rebuilt.

- 1 On the DBA Administration menu, select the Display Table Information option.
- 2 Click the **Display** button. The Table Information screen is displayed.

The screenshot shows a window titled 'Display Table Information' with a search bar labeled 'FIND TABLE'. Below the search bar is a table with the following columns: TABLE NAME, TABLESPACE NAME, KBYTES, BLOCKS, EXTENTS, MAX EXTENTS, and BYTES. The table lists various database tables such as ORDER_QUERIES_DETAIL, ORDER_TOTALS, OUTBOUND_ASN_QUEUE, etc.

TABLE NAME	TABLESPACE NAME	KBYTES	BLOCKS	EXTENTS	MAX EXTENTS	BYTES
ORDER_QUERIES_DETAIL	DAT3	9216	1152	9	#####	9437184
ORDER_TOTALS	DAT3	5120	640	5	#####	5242880
OUTBOUND_ASN_QUEUE	USERS	64	8	1	#####	65536
OUTBOUND_CARTON	DAT1	192	24	3	#####	196608
OUTBOUND_QC	DAT2	576	72	9	#####	589824
OUTBOUND_WORK_ORDER	DAT1	2048	256	2	#####	2097152
OUTBOUND_WORK_ORDER_I	DAT2	128	16	2	#####	131072
OUTSTANDING_TOTALS	DAT3	576	72	9	#####	589824
OVERWRITE_SETUP_COLUMNS	DAT1	512	64	8	#####	524288
OVERWRITE_SETUP_TABLES	DAT1	512	64	8	#####	524288
PATCHES_INSTALLED	DAT1	64	8	1	#####	65536
PENDING_RETURNS	DAT1	3072	384	3	#####	3145728
PENDING_RETURNS_DETAIL	DAT2	9216	1152	9	#####	9437184
PENDING_RETURNS_DETAIL_DL	DAT3	3072	384	3	#####	3145728
PENDING_RETURNS_DL	DAT3	576	72	9	#####	589824
PERIOD	DAT3	64	8	1	#####	65536
PICKPACK_WAVE_DIRECTIVE	DAT3	192	24	3	#####	196608
PICK_DIRECTIVE	DAT3	192	24	3	#####	196608
PICK_FROM_LOC	DAT1	576	72	9	#####	589824
PICK_MODULE_TOTE_DIR_T	DAT5	64	8	1	#####	65536
PLAN_TABLE	USERS	512	64	8	#####	524288
PO	DAT1	640	80	10	#####	655360
PO_DETAIL	DAT2	3072	384	3	#####	3145728
PO_DETAIL_DL	DAT2	192	24	3	#####	196608

Table Information Screen

These are the fields on the Table Information screen:

Field name	Field description
Find Table	Table name for table to be queried.
Table Name	Name of the database table.
Tablespace Name	Tablespace name.
Kbytes	Number of (K) bytes in the table.
Blocks	Number of blocks the table is using.
Extents Cur.	Current table extents.
Extents Max	Maximum allowable table extents.

- 3 Click the **Display** button at the blank Find Table field to display a list of all tables.

Note: If you want to display information about a particular table, enter the specific table name at the Find Table field. You can also enter a partial table name. For example, you can enter 'APP' to display all tables that begin with these letters.

- 4 Click the **Exit** button to return to the DBA Administration menu.

Display tablespace information

You can also use the Display Tablespace Information option to display tablespace specific information, such as the amount of free space in a tablespace or the number of extents in a table space.

- 1 On the Monitoring and Administration menu, select the Display Tablespace Information option.
- 2 Click the **Display** button. The Tablespace Information screen is displayed, along with all tablespace.

[illegible]

Tablespace Information Screen

These are the fields on the Tablespace Information screen:

Field name	Field description
Tablespace Name	Name of the database tablespace.
File Name	Name of the data file.
Mbytes	Table size in mega bytes.
Status	Indicates whether tablespace is Available or offline.

- 3 Click the **Refresh** button to refresh the screen and view any new tablespace information.

Display rollback information

You can use the Display Rollback Information option to display information about rollbacks. You can also use this information to determine whether the rollback segments need to be enlarged for a specific installation.

- 1 On the DBA Administration menu, select the Display Rollback Information option.
- 2 The Rollback Information screen is displayed, along with all rollback segments.

[illegible]

Rollback Information Screen

These are the fields on the Rollback Information screen:

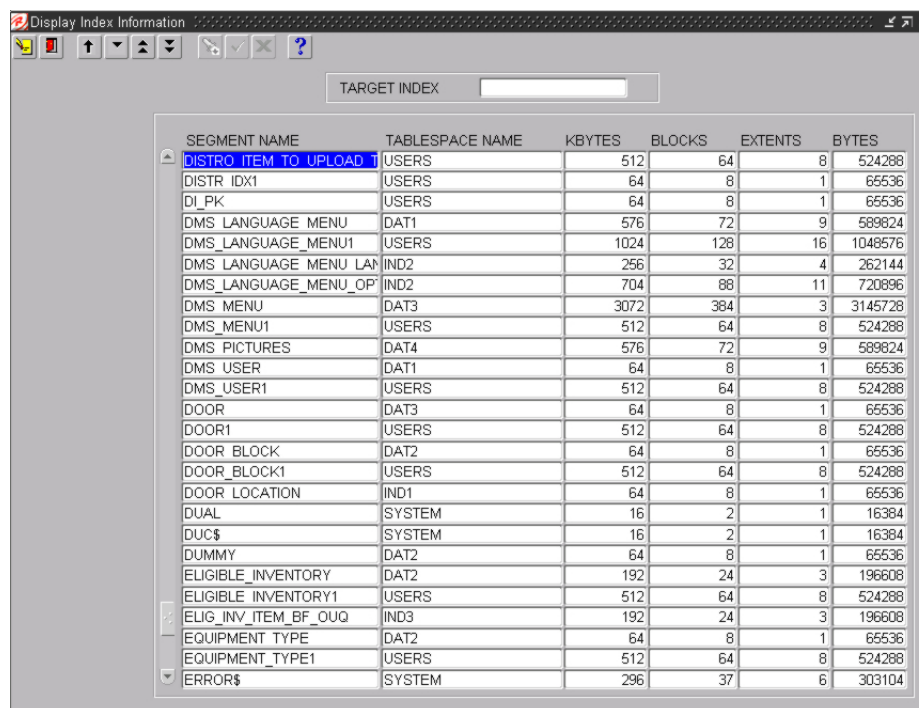
Field name	Field description
Rollback Segment	Name of the rollback segment.
Increase RB	Indicates when additional rollback segments need to be added. YES = rollback segments should be added. NO = rollback segments do not need to be added.
KSize	Size of rollback segments in bytes.
Extents	The number of times that the rollback segment had to acquire a new extent
XACTS	Number of Active Transactions
WAITS	The number of rollback segment header requests that resulted in waits
GETS	The number of rollback segment header requests
OPTSIZE	The value of the optimal parameter for the rollback segment
STATUS	Status (Online / Offline)
RRSIZE	Rollback Size

- 3 Click the **Refresh** button to refresh the screen and view any new information.
- 4 Click the **Exit** button to return to the DBA Administration menu.

Display index information

You can use the Display Index Information option to display sizing information for the database indexes. You can use this information to analyze the growth of the database indexes, which can indicate table growth.

- 1 On the DBA Administration menu, select the Display Index Information option.
- 2 Click the **Display** button. The Index Information screen is displayed.



SEGMENT NAME	TABLESPACE NAME	KBYTES	BLOCKS	EXTENTS	BYTES
DISTRO_ITEM_TO_UPLOAD_T	USERS	512	64	8	524288
DISTR_IDX1	USERS	64	8	1	65536
DI_PK	USERS	64	8	1	65536
DMS_LANGUAGE_MENU	DAT1	576	72	9	589824
DMS_LANGUAGE_MENU1	USERS	1024	128	16	1048576
DMS_LANGUAGE_MENU_LAN	IND2	256	32	4	262144
DMS_LANGUAGE_MENU_OP	IND2	704	88	11	720896
DMS_MENU	DAT3	3072	384	3	3145728
DMS_MENU1	USERS	512	64	8	524288
DMS_PICTURES	DAT4	576	72	9	589824
DMS_USER	DAT1	64	8	1	65536
DMS_USER1	USERS	512	64	8	524288
DOOR	DAT3	64	8	1	65536
DOOR1	USERS	512	64	8	524288
DOOR_BLOCK	DAT2	64	8	1	65536
DOOR_BLOCK1	USERS	512	64	8	524288
DOOR_LOCATION	IND1	64	8	1	65536
DUAL	SYSTEM	16	2	1	16384
DUC\$	SYSTEM	16	2	1	16384
DUMMY	DAT2	64	8	1	65536
ELIGIBLE_INVENTORY	DAT2	192	24	3	196608
ELIGIBLE_INVENTORY1	USERS	512	64	8	524288
ELIG_INV_ITEM_BF_OUQ	IND3	192	24	3	196608
EQUIPMENT_TYPE	DAT2	64	8	1	65536
EQUIPMENT_TYPE1	USERS	512	64	8	524288
ERROR\$	SYSTEM	296	37	6	303104

Index Information Screen

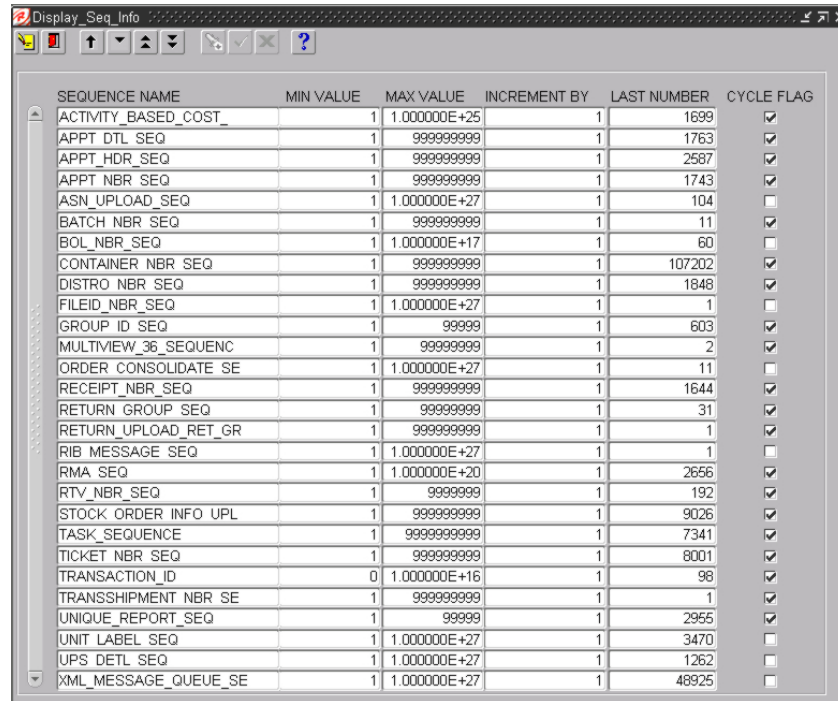
These fields are on the Index Information screen:

Field name	Field description
Target Index	Index name user wishes to query.
Segment Name	Name of index.
Tablespace Name	Tablespace name where the index resides.
Kbytes	Size of index in kilobytes.
Blocks	Number of blocks the index is using (1 block = 4096 bytes).
Extents	Current number of index extents.
Bytes	Size of index in bytes.

Display sequences information

You can use the Display Sequence Information option to display sizing information specific to the sequences used by the system. You can use this information to determine whether a sequence is being called as many times as originally planned.

- 1 On the DBA Administration menu, select the Display Sequences Information option.
- 2 The Sequences Information screen is displayed, along with the sequence information already entered into the system.



SEQUENCE NAME	MIN VALUE	MAX VALUE	INCREMENT BY	LAST NUMBER	CYCLE FLAG
ACTIVITY_BASED_COST_	1	1.000000E+25	1	1699	<input checked="" type="checkbox"/>
APPT DTL SEQ	1	999999999	1	1763	<input checked="" type="checkbox"/>
APPT_HDR_SEQ	1	999999999	1	2587	<input checked="" type="checkbox"/>
APPT_NBR_SEQ	1	999999999	1	1743	<input checked="" type="checkbox"/>
ASN_UPLOAD_SEQ	1	1.000000E+27	1	104	<input type="checkbox"/>
BATCH_NBR_SEQ	1	999999999	1	11	<input checked="" type="checkbox"/>
BOL_NBR_SEQ	1	1.000000E+17	1	60	<input type="checkbox"/>
CONTAINER_NBR_SEQ	1	999999999	1	107202	<input checked="" type="checkbox"/>
DISTRO_NBR_SEQ	1	999999999	1	1848	<input checked="" type="checkbox"/>
FILEID_NBR_SEQ	1	1.000000E+27	1	1	<input type="checkbox"/>
GROUP_ID_SEQ	1	99999	1	603	<input checked="" type="checkbox"/>
MULTIVIEW_36_SEQUENC	1	999999999	1	2	<input checked="" type="checkbox"/>
ORDER_CONSOLIDATE_SE	1	1.000000E+27	1	11	<input type="checkbox"/>
RECEIPT_NBR_SEQ	1	999999999	1	1644	<input checked="" type="checkbox"/>
RETURN_GROUP_SEQ	1	999999999	1	31	<input checked="" type="checkbox"/>
RETURN_UPLOAD_RET_GR	1	999999999	1	1	<input checked="" type="checkbox"/>
RIB_MESSAGE_SEQ	1	1.000000E+27	1	1	<input type="checkbox"/>
RMA_SEQ	1	1.000000E+20	1	2656	<input checked="" type="checkbox"/>
RTV_NBR_SEQ	1	9999999	1	192	<input checked="" type="checkbox"/>
STOCK_ORDER_INFO_UPL	1	999999999	1	9026	<input checked="" type="checkbox"/>
TASK_SEQUENCE	1	999999999	1	7341	<input checked="" type="checkbox"/>
TICKET_NBR_SEQ	1	999999999	1	8001	<input checked="" type="checkbox"/>
TRANSACTION_ID	0	1.000000E+16	1	98	<input checked="" type="checkbox"/>
TRANSHIPMENT_NBR_SE	1	999999999	1	1	<input checked="" type="checkbox"/>
UNIQUE_REPORT_SEQ	1	99999	1	2955	<input checked="" type="checkbox"/>
UNIT_LABEL_SEQ	1	1.000000E+27	1	3470	<input type="checkbox"/>
UPS_DETL_SEQ	1	1.000000E+27	1	1262	<input type="checkbox"/>
XML_MESSAGE_QUEUE_SE	1	1.000000E+27	1	48925	<input type="checkbox"/>

Sequences Information Screen

These are the fields on the Sequences Information screen:

Field name	Field description
Sequence Name	Name of the database sequence.
Min Value	Minimum value of the sequence.
Max Value	Maximum value of the sequence.
Increment By	Increment, amount the sequence increases after each use.
Last Number	Last sequence value used. Some sequences cache the values in memory so this number does not increase until the cache is empty and a new group of numbers is cached into memory.

These are the fields on the Error Log Inquiry screen:

Field name	Field description
Enter User, Code, Date	Enter any combination of user, error code, or error date to search for records.
User	User identification.
Error Time	The date and time the error was logged.
Code	The error code.
Source	Program where the error originated

- Click the **Display** button at the blank Enter User field to display a list of all existing errors.

Note: To display the errors for a particular user, enter the user name in the Enter User field.

- To display a specific error for a particular user, enter the user name in the Enter User field and the error code in the Code field.
- To display a specific error for a specific user for a particular date, enter the user name in the Enter User field, the error code in the Code field, and the date in the Date field.

Retek Distribution Management accepts any combination of the above fields.

View error log details

- On the Monitoring and Administration menu, select the Display Error Log option, using the keypad arrow keys to move up and down the list.
- Click the **Display** button. The Error Log Inquiry screen is displayed.
- Enter the user, error code, or date that you want to view and click the **Display** button, or click the **Display** button at the blank Enter User field to display a list of all errors.
- Select the record you want to view in more detail, using the keypad arrow keys to move up and down the list, then click the **Details** button. The Error Log screen is displayed.

The screenshot shows a window titled "Error Detail" with a close button (X) in the top right corner. Inside the window, there are several input fields and labels:

- USER:** A text box containing "PAR3214".
- TIME:** A text box containing "03/05/2002 11:01:17".
- CODE:** A text box containing "40508".
- ERROR SOURCE:** A text box containing "ITEM_MASTER_EDITOR_S".
- LOCATION ID:** A text box containing "Tr_On_Error".
- MESSAGE:** A text box containing "ORACLE error: unable to INSERT record."

At the bottom center of the window, there is a button labeled "Exit/Cancel".

Error Log screen

These are the fields on the Error Log screen:

Field name	Field description
User	Identification of the user who had the error.
Time	Date and time the error was logged.
Code	The error code.
Source	The program in which the error originated.
Location	The location within the source program where the error occurred.
Message	Full text of the error message.

- 5 Click the **Cancel** button to return to the Error Log screen.

Delete error log records

- 1 On the Monitoring and Administration menu, select the Display Error Log Inquiry option, using the keypad arrow keys to move up and down the list.
- 2 Click the **Display** button to enter the Error Log Inquiry screen.
- 3 Enter the user, error code, or date that you want to delete and click the **Display** button, or click the **Display** button at the blank Enter User field to display a list of all errors.
- 4 Select the record you want to delete, using the keypad arrow keys to move up and down the list, then click the **Delete** button. A message box asks you to confirm the deletion. The message reads:
Confirm Delete Operation (Yes/No)
- 5 Click on the Yes button to delete the error log record.

Print the error log report

- 1 On the Monitoring and Administration menu, select the Display Error Log option, using the keypad arrow keys to move up and down the list.
- 2 Click the **Display** button to enter the Enter Log Inquiry screen.
- 3 Enter the user, error code or date that you want to print and click the **Display** button at the blank Enter User field to display the Error Log report, which lists all existing errors.
- 4 Click the **Print** button. The Report Destination Pop-Up screen is displayed.
- 5 Click the **Print** button to print the report.
- 6 Click the **Cancel** button to cancel printing the report. The Error Log report is shown.

03/20/02 Page 1

Error Log

USER ID	ERROR TIME	CODE	ERROR SOURCE	ERROR LOC	MESSAGE
ANDY	03-06 07:36	41067	REPRINT_NULL_LABELS_S	Tr_On_Error	Cannot find Menu Item: invalid ID.
ANDY	03-11 09:26	-4068	INV_INQ_BY_ITEM_S	Tr_On_Error	ORA-04068: existing state of packages () has been discarded ORA-04061: existing state of package body "PAR3214.AHL" has been invalidated
ANDYS	03-06 07:34	41067	REPRINT_NULL_LABELS_S	Tr_On_Error	Cannot find Menu Item: invalid ID.
ANDYS	03-06 07:34	41067	ERROR_LOG_S	Tr_On_Error	Cannot find Menu Item: invalid ID.
ANDYS	03-06 07:35	41067	USER_TABLE_EDITOR_S	Tr_On_Error	Cannot find Menu Item: invalid ID.
ANDYS	03-06 11:37	41067	ERROR_LOG_S	Tr_On_Error	Cannot find Menu Item: invalid ID.
ANDYS	03-06 14:51	41067	RETURN_CODE_EDITOR_S	Tr_On_Error	Cannot find Menu Item: invalid ID.
ANDYS	03-06 14:52	41067	REPRINT_NULL_LABELS_S	Tr_On_Error	Cannot find Menu Item: invalid ID.
LALIT	03-18 13:19	0	v_container_tote_id	before print order	DD
LALIT	03-18 13:19	0	PRINT ORDER	after report	DD

Error Log Report

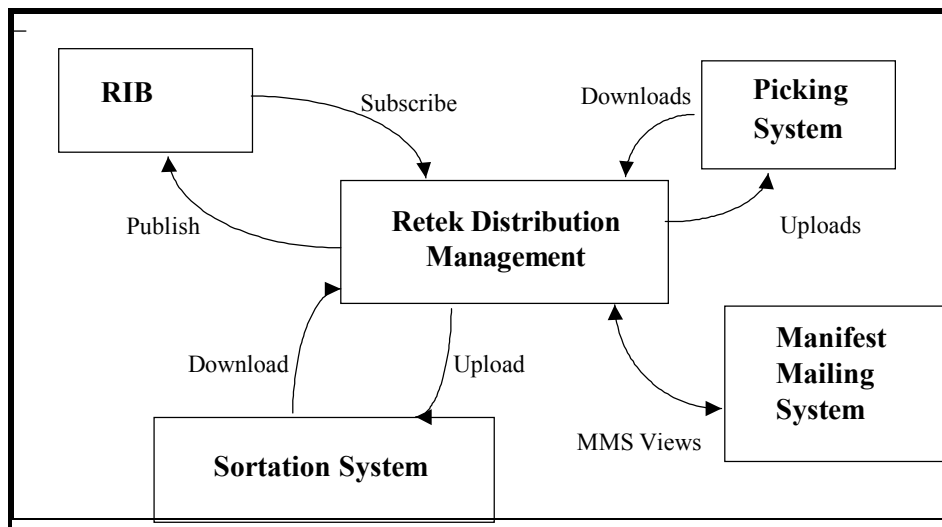
These are the fields on the Error Log report:

Field name	Field description
User ID	Identification of the user who had the error.
Error Time	Date and time the error was logged.
Code	The error code.
Error Source	Program where the error originated.
Error Loc	The location (procedure, block, etc.) within the source program where the error occurred.
Message	Full text of the error message.

Chapter 3 – RDM RIB components

Overview

This chapter and the next describe the various interfaces with the Retek Distribution Management System. These include Host System through the Retek Integration Bus (RIB), Picking System, Sortation System, and Manifest Mailing System links.



Integration Points to RDM

Acronyms

These acronyms are used throughout this section:

ASCII: American National Standard Code for Information Interchange

ASN: Advance Shipment Notice

DC: Distribution Center

PO: Purchase Order

SKU: Stock Keeping Unit

Terms

These terms are used throughout this section:

Appointment: A scheduled arrival of in-bound merchandise.

ASN: Advance Shipment Notice. A Host Download that provides either a list of containers and their contents, or a set of PO/Item/Destinations.

Container: A type of receptacle (such as a carton, pallet, tote, roll cage.) that contains items and/or other containers.

Destination: The ultimate source for containers. This covers out-bound destinations, including the DC itself and internal replenishment. Also referred to as the Shipping Destination. For consumer direct order fulfillment, this field is used to specify the shipment method or parcel carrier service.

Download: Any data file coming into Retek Distribution Management.

Field: An individual data element within a record.

File: The mechanism by which batch data is transferred. These are ASCII files.

Future use: The field is not currently used in Retek Distribution Management, but may be used in a future release.

Host: The controlling computer system. Often housed at corporate headquarters.

Item: A specified part number, SKU, etc.

Optional: The field is used for information purposes and is not required.

Predistribution: Allocation of merchandise in advance of receipt to facilitate flow through or cross-dock upon arrival, bypassing storage and going directly to break case picking area or shipping.

Purchase Order: The list of items and quantities authorized to receive from a specific vendor.

Record: A single line of data in a file.

RIB: Retek Integration Bus

Upload: Any data file going out from Retek Distribution Management to another system.

Vendor: A supplier of in-bound goods. Each PO is assigned to a vendor.

RDM Message subscription process

Principally, the new RIB architecture utilizes a uniform structure. The following is a description of the methods used by the subscription process:

- The RDM adapter recognizes that a message with the specific name with a prefix of RDMSUB(e.g. RDMSUB_LOC for Location subscription) exists on the RIB.
- The adapter calls the public PL/SQL procedure to “consume” the message. The public “consume” procedure is named:

`RDMSUB_XXXX.CONSUME`

This procedure accepts an Oracle Object containing the message information along with the message type (i.e. Locationcre/Locationmod/Locationdel). It calls `sub_XXXX.process_message` to process the message and based on the information received back, it returns a `status_code` and `error_message`. When a message is successfully consumed the `status_code` will return an ‘S’ and the `error_message` will be null. The status code might return with an error code if the message processing failed and the `error_message` will contain the error text.

- The ‘consume’ procedure calls ‘process_message’ to perform the subscription process:

`SUB_XXXX.PROCESS_MESSAGE`

This procedure is the engine of the message processing. It performs all the validation, additional processing and data changes (Inserts, Updates and Deletes) by calling other private functions and procedures.

Detail Information is provided in the corresponding component section of the specific subscription.

RDM Message publication process

Principally, the new RIB architecture utilizes a uniform structure. The following is a description of the methods used by the Publish process:

- The RDM receives a publish request from the RIB. Publisher messages all have a prefix of RDMMFM (e.g. RDMMFM_RECEIVING for publishing appointments/receipts). These are the Message Family Managers.
- The Message Family Managers (MFM) calls the public PL/SQL procedure to create the message to publish. The public procedure is named:

PUB_XXXX

This procedure accepts inputs like facility_type/dc_dest_id/pub_seq.

It returns an Oracle Object containing the message information along with the message type (e.g. AppointCre for creating appointments). It also returns an error_message if the publication of the message failed. When the message is published successfully, the error_message is null. In several messages, messages are aggregated for 'bulk processing'. This may be based on valid business criteria or simply based on 'max details to publish' as defined in the rib_settings table.

Detail Information is provided in the corresponding component section of the specific subscription.

Subscription components

Vendor subscription

Vendor messages are used by RDM to create and maintain Vendor and Vendor Address information. Vendor messages are published by a Host system.

Vendor Information is used by RDM in the inbound processing of Purchase Orders, Items, Receiving, Returns and RTV.

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.

Vendor message structure

The Vendor family of messages can create, modify and delete Vendor records as well as create, modify, and delete Vendor Addresses. All of the message types are composed of the following sections:

- Message header—This is data about the Vendor including the Number and Name as well as auditing and sampling requirements for received product.
- Address record—Address Type (i.e. Billing, Shipping, etc) Primary Indicator and basic address information.

Vendor message components

The following is a description of the Vendor message components:

(For a general description of the ‘Consume’ and Process Message’ methods please refer to the preceding ‘RDM Message subscription process’ section of this document)

- Name Of the ‘Consume’ method: RDMSUB_VENDOR
- Name Of the ‘Process Message’ method: SUB_VENDOR

Message summary

All Vendor messages belong to the Vendor message family. The structure of the message depends on the message type to be performed. The following table lists each message, its structure and the associated Rib_Object.

Message	Structure	RIB_Object Type
Vendor Create (VDR_ALL)	Header and Address	Rib_VendorDesc_Rec
Vendor Modify (VDR_MOD)	Header only	Rib_VendorHdrDesc_Rec
Vendor Delete (VDR_DEL)	Header only	Rib_VendorRef_Rec
Vendor Address Create (VDRD_ADD)	Address only	Rib_VendorAddrDesc_Rec
Vendor Address Modify (VDRD_MOD)	Address only	Rib_VendorAddrDesc_Rec
Vendor Address Delete (VDRD_DEL)	Address only	Rib_VendorAddrRef_Rec

Primary vendor tables

The following are the primary tables in RDM that hold Vendor data:

- VENDOR
- VENDOR_ADDRESS

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

Status and error messages

For status and error messages returned from the API, see the Operations Guide – Volume 2: Error Codes.

Location subscription

Location messages, known as Ship Dest to RDM, are used by RDM to create and maintain Ship Destination records.

Ship Dest information is used by the warehouse to know where to ship merchandise and what methods / carriers to use.

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.

Location message structure

The Location or Ship Dest family of messages can create, modify and delete Ship Dest records. Ship Dest messages includes a Destination Identifier, address information, Carrier Information, Currency Codes, and Country Codes.

Location message components

The following is a description of the Location message components:

(For a general description of the ‘Consume’ and Process Message’ methods please refer to the preceding ‘RDM Message subscription process’ section of this document)

- Name Of the ‘Consume’ method: RDMSUB_LOC
- Name Of the ‘Process Message’ method: SUB_LOC

Message summary

All Location messages belong to the Location message family. The structure of the message depends on the message type to be performed. The following table lists each message, it’s structure and the associated Rib_Object.

Message	Structure	RIB_Object Type
Location Create (LOCATIONCRE)	Header only	Rib_LocationDesc_Rec
Location Modify (LOCATIONMOD)	Header only	Rib_LocationDesc_Rec
Location Delete (LOCATIONDEL)	Header only	Rib_LocationRef_Rec

Primary location tables

The following are the primary tables in RDM that hold Location data:

- SHIP_DEST

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

Status and error messages

For status and error messages returned from the API, see the Operations Guide – Volume 2: Error Codes.

Item subscription

The Item messages are used by RDM to create and maintain Item and Item supporting information. Item messages are published by a Host system.

Items represent the actual merchandise that is received and shipped from the warehouse. The Item messages provide detail information about the merchandise including the Vendor, dimensions, and user defined attributes.

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.

Item message structure

The Item family of messages can create, modify, and delete Item Master and related sub Item table records. The Item messages consist of the following areas: Item, Supplier Information, Attributes, Differentiators, Bill Of Materials, and UPC. A brief description of each node is provided below.

- Item—This is data about the Item itself including Vendor, Description, basic dimensions and weight. Also included in the Item node is the Item Differentiator information which provides a cross reference between the Item and the Differentiator / Differentiator Group tables.
- Item Supplier—The list of suppliers for list item including the primary supplier indicator.
- Item Supplier Country – The list of countries for each supplier including the primary country indicator. Additional information provide includes Inner Pack Size and TI / HI.
- Item Supplier Country Dimensions – The list of dimensions by object type (EACH, CARTON) by country.
- Item Attributes – The cross reference information between the Item and Attributes / Attribute Types.
- Bill of Materials – Information to relate the Master Item to the Component Items when creating pack items.
- Item UPC – Information to relate the Item to a UPC code.

Item message components

The following is a description of the Item message components:

(For a general description of the ‘Consume’ and Process Message’ methods please refer to the preceding ‘RDM Message subscription process’ section of this document)

- Name Of the ‘Consume’ method: RDMSUB_ITEMS
- Name Of the ‘Process Message’ method: SUB_ITEMS

Message summary

All Item messages belong to the Item message family. The structure of the message depends on the message type to be performed. The following table lists each message, it’s structure and the associated Rib_Object.

Message	Structure	RIB_Object Type
Item Create (ITEMCRE)	Header and all details	Rib_ItemDesc_Rec
Item Modify (ITEMHDRMOD)	Header only	Rib_ItemHdrDesc_Rec
Item Delete (ITEMDEL)	Header only	Rib_ItemRef_Rec
BOM Create (ITEMBOMCRE)	BOM detail only	Rib_ItemBOMDesc_Rec
BOM Modify (ITEMBOMMOD)	BOM detail only	Rib_ItemBOMDesc_Rec
BOM Delete (ITEMBOMDEL)	BOM detail only	Rib_ItemBOMRef_Rec
UPC Create (ITEMUPCCRE)	UPC detail only	Rib_ItemUPCDesc_Rec
UPC Modify (ITEMUPCMOD)	UPC detail only	Rib_ItemUPCDesc_Rec
UPC Delete (ITEMUPCDEL)	UPC detail only	Rib_ItemUPCRef_Rec
UDA List of Values Create (ITEMUDALOVCRE)	UDA List of Values detail only	Rib_ItemUDALOVDesc_Rec
UDA List of Values Modify (ITEMUDALOVMOD)	UDA List of Values detail only	Rib_ItemUDALOVDesc_Rec
UDA List of Values Delete (ITEMUDALOVDEL)	UDA List of Values detail only	Rib_ItemUDALOVRef_Rec

Message	Structure	RIB_Object Type
Item Supplier Create (ITEMSUPCRE)	Item Supplier detail only	Rib_ItemSupDesc_Rec
Item Supplier Modify (ITEMSUPMOD)	Item Supplier detail only	Rib_ItemSupDesc_Rec
Item Supplier Delete (ITEMSUPDEL)	Item Supplier detail only	Rib_ItemSupRef_Rec
Item Supplier Country Create (ITEMSUPCTYCRE)	Item Supplier Country Detail only	Rib_ItemSupCtyDesc_Rec
Item Supplier Country Modify (ITEMSUPCTYMOD)	Item Supplier Country Detail only	Rib_ItemSupCtyDesc_Rec
Item Supplier Country Delete (ITEMSUPCTYDEL)	Item Supplier Country Detail only	Rib_ItemSupCtyRef_Rec
Item Supplier Country Dimension Create (ITEMISCDIMCRE)	Item Supplier Country Dimension detail only	Rib_ISCDimDesc_Rec
Item Supplier Country Dimension Modify (ITEMISCDIMMOD)	Item Supplier Country Dimension detail only	Rib_ISCDimDesc_Rec
Item Supplier Country Dimension Delete (ITEMISCDIMDEL)	Item Supplier Country Dimension detail only	Rib_ISCDimRef_Rec

Primary item tables

The following are the primary tables in RDM that hold Item data:

- ITEM_MASTER
- ITEM_SUPPLIER
- ITEM_SUPP_COUNTRY
- ITEM_SUPP_COUNTRY_DIM
- BILL_OF_MATERIALS
- ITEM_UPC
- ITEM_ATTRIBUTES

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

Status and error messages

For status and error messages returned from the API, see the Operations Guide – Volume 2: Error Codes.

UDA subscription

UDA Information, known in RDM as Attributes and Attribute Types, is used to allow the user to define additional attributes for an Item. For example, for a cotton T-shirt, an Attribute of COTTON, meaning Cotton Short Sleeve T-Shirt, can be created and related to an Item through the Item Attribute table (see the ITEM subscription documentation for more information concerning the Item Attribute message).

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.

UDA message structure

The UDA family of messages consists of two message types: UDA (Attribute Types) and UDA Values (Attributes). Both messages are single node structures.

UDA Type

- This message includes the UDA Identifier and Description.

UDA Value Type

- This message includes the UDA Value Identifier and Description and the UDA Identifier.

UDA message components

The following is a description of the UDA message components:

(For a general description of the ‘Consume’ and Process Message’ methods please refer to the preceding ‘RDM Message subscription process’ section of this document)

- Name Of the ‘Consume’ method: RDMSUB_UDAS
- Name Of the ‘Process Message’ method: SUB_UDAS

Message summary

All UDA messages belong to the UDA message family. The structure of the message depends on the message type to be performed. The following table lists each message, its structure and the associated Rib_Object

Message	Structure	RIB_Object Type
UDA Create (UDACRE)	Header only	Rib_UDADesc_Rec
UDA Modify (UDAMOD)	Header only	Rib_UDADesc_Rec
UDA Delete (UDADEL)	Header only	Rib_UDAREf_Rec
UDA Detail Create (UDAVALCRE)	Detail only	Rib_UDAVALDesc_Rec
UDA Detail Modify (UDAVALMOD)	Detail only	Rib_UDAVALDesc_Rec
UDA Detail Delete (UDAVALDEL)	Detail only	Rib_UDAVALRef_Rec

Primary UDA tables

The following are the primary tables in RDM that hold UDA data:

- ATTRIBUTES
- ATTRIBUTE_TYPE

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

Status and error messages

For status and error messages returned from the API, see the **Operations Guide – Volume 2: Error Codes**.

Differentiator subscription

Differentiators and Differentiator Groups are created and then associated to Items through the Item Differentiator table (see the ITEM subscription documentation for more information concerning the Item Differentiator message). This information allows the user further characterize and group Items.

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.

Differentiator message structure

The Differentiator family of messages consists of three message types: Differentiators, Differentiator Groups and Differentiator Group Details. All of these messages are single node structures.

Differentiator Type

- This message includes a Differentiator Identifier, Description and Type.

Differentiator Group Type

- This message includes a Differentiator Group Identifier, Group Description and Type.

Differentiator Group Details Type

- This message includes the Differentiator Identifier, Differentiator Group Identifier, and Description.

Differentiator message components

The following is a description of the Differentiator message components:

(For a general description of the ‘Consume’ and Process Message’ methods please refer to the preceding ‘RDM Message subscription process’ section of this document)

- Name of the ‘Consume’ methods: RDMSUB_DIFFS, RDMSUB_DIFFGRP
- Name of the ‘Process Message’ methods: SUB_DIFFS, SUB_DIFFGRP

Message summary

All Differentiator messages belong to the Differentiator message family. The structure of the message depends on the message type to be performed. The following table lists each message, its structure and the associated Rib_Object.

Message	Structure	RIB_Object Type
Differentiator Create (DIFFCRE)	Header only	Rib_DiffDesc_Rec
Differentiator Modify (DIFFMOD)	Header only	Rib_DiffDesc_Rec
Differentiator Delete (DIFFDEL)	Header only	Rib_DiffRef_Rec
Differentiator Group Create (DIFFGRPCRE)	Header only	Rib_DiffGrpHdrDesc_Rec
Differentiator Group Modify (DIFFGRPMOD)	Header only	Rib_DiffGrpHdrDesc_Rec
Differentiator Group Delete (DIFFGRPDEL)	Header only	Rib_DiffGrpRef_Rec
Differentiator Group Detail Create (DIFFGRPDTLCRE)	Detail only	Rib_DiffGrpDtlDesc_Rec
Differentiator Group Detail Modify (DIFFGRPDTLMOD)	Detail only	Rib_DiffGrpDtlDesc_Rec
Differentiator Group Detail Delete (DIFFGRPDTLDEL)	Detail only	Rib_DiffGrpDtlRef_Rec

Primary differentiator tables

The following are the primary tables in RDM that holds Differentiator data:

- DIFFERENTIATOR
- DIFFERENTIATOR_GROUP
- DIFF_GROUP_DETAIL

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

Status and error messages

For status and error messages returned from the API, see the **Operations Guide – Volume 2: Error Codes**.

Purchase order subscription

Purchase Order messages are used by RDM to create and maintain PO and PO Detail information. Purchase Order messages are published by a Host system.

Purchase Order messages authorize a warehouse to be able receive merchandise from a Vendor. These messages provide information to the warehouse about the amount of each item that can be received into the warehouse as well as acceptable date ranges for delivery.

The Purchase Order messages are specific to a particular warehouse instance and therefore contain routing information so that the bus can guarantee successful delivery of the message to the appropriate DC.

Purchase order message structure

The Purchase Order family of messages can create, modify and delete Purchase Order records as well as create, modify, and delete Purchase Order details. All of the message types are composed of the following sections:

Purchase order message components

The following is a description of the Purchase Order message components:

(For a general description of the 'Consume' and Process Message' methods please refer to the preceding 'RDM Message subscription process' section of this document)

- Name of the 'Consume' methods: RDMSUB_ORDER
- Name of the 'Process Message' methods: SUB_PO

Message summary

All PO messages belong to the PO message family. The structure of the message depends on the message type to be performed. The following table lists each message, its structure and the associated Rib_Object.

Message	Structure	RIB_Object Type
Purchase Order Create (POPhysCre)	Header and Detail	Rib_PoPhyDesc_Rec
Purchase Order Modify (POPhysMod)	Header only	Rib_PoPhyDesc_Rec
Purchase Order Delete (POPhysDel)	Header only	Rib_PoRef_Rec
Purchase Order Detail Create (PODtIPhysCre)	Header and Detail	Rib_PoPhyDesc_Rec
Purchase Order Detail Modify (PODtIPhysMod)	Header and Detail	Rib_PoPhyDesc_Rec
Purchase Order Detail Delete (PODtIPhysDel)	Detail only	Rib_PoDtIRef_Rec

Primary purchase order tables

The following are the primary tables in RDM that hold PO data:

- PO
- PO_DETAIL

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

Status and error messages

For status and error messages returned from the API, see the Operations Guide – Volume 2: Error Codes.

Inbound work order subscription

Inbound Work Order messages are used by RDM to create and maintain work order information. Inbound Work Order messages are published by a Host system.

Inbound Work Order messages represent a request for the warehouse to perform work on the merchandise before it is shipped to the stores or customers.

The Inbound Work Order messages are specific to a particular warehouse instance and therefore contain routing information so that the bus can guarantee successful delivery of the message to the appropriate DC.

Inbound work order message structure

The Inbound Work Order family of messages can create, modify and delete Inbound Work Order records. The message includes the following information: Item, WIP Code, Sequence and Instructions.

Inbound work order message components

The following is a description of the Inbound Work Order message components:

(For a general description of the ‘Consume’ and Process Message’ methods please refer to the preceding ‘RDM Message subscription process’ section of this document)

- Name of the ‘Consume’ methods: RDMSUB_WOIN
- Name of the ‘Process Message’ methods: SUB_WOIN

Message summary

All Inbound Work Order messages belong to the Inbound Work Order message family. The structure of the message depends on the message type to be performed. The following table lists each message, it’s structure and the associated Rib_Object.

Message	Structure	RIB_Object Type
Inbound Work Order Create (INBDWOCre)	Header only	Rib_WOInDesc_Rec
Inbound Work Order Modify (INBDWOMod)	Header only	Rib_WOInDesc_Rec
Inbound Work Order Delete (INBDWODEl)	Header only	Rib_WOInRef_Rec

Primary inbound work order tables

The following are the primary tables in RDM that hold Inbound Work Order data:

- INBOUND_WORK_ORDER

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

Status and error messages

For status and error messages returned from the API, see the Operations Guide – Volume 2: Error Codes.

Inbound ASN subscription

Inbound ASN messages are used by RDM to create and maintain Advanced Shipment Information within the system. Inbound ASN messages are published by an outside Vendor or by another warehouse through the publication and transformation on an Outbound ASN.

Inbound ASN messages represent an Advanced Ship Notice of incoming merchandise. These messages provide information to the warehouse about the amount of each item that is coming to the DC.

The Inbound ASN messages are specific to a particular warehouse instance and therefore contain routing information so that the bus can guarantee successful delivery of the message to the appropriate DC.

Inbound ASN message structure

The Inbound ASN messages come in two style depending on the type. PO Type ASNs provide information about the Items being shipped to the warehouse. Carton Type ASNs also provide information about the Items and in addition supply all of the carton information as well. The two structures share common nodes, detailed below:

- Message header–ASN Number, Type, Carrier.
- PO record–Purchase Order information.
- Carton–(for Carton Type ASNs) Container Identifier, dimensions
- Items– Details about all items in the Container.

Inbound ASN message components

The following is a description of the BOL message components:

(For a general description of the ‘Consume’ and Process Message’ methods please refer to the preceding ‘RDM Message subscription process’ section of this document)

- Name of the ‘Consume’ methods: RDMSUB_ASNIN
- Name of the ‘Process Message’ methods: SUB_ASN

Message summary

All ASN messages belong to the ASN message family. The structure of the message depends on the message type to be performed. The following table lists each message, it's structure and the associated Rib_Object.

Message	Structure	RIB_Object Type
ASN Inbound PO Create (ASNINPOCre)	Header and Detail	Rib_ASNInDesc_Rec
ASN Inbound PO Modify (ASNINPOMod)	Header and Detail	Rib_ASNInDesc_Rec
ASN Inbound PO Delete (ASNINPODel)	Header only	Rib_ASNInRef_Rec
ASN Inbound Container Create (ASNINCTNCre)	Header and Detail	Rib_ASNInDesc_Rec
ASN Inbound Container Modify (ASNINCTNCre)	Header and Detail	Rib_ASNInDesc_Rec
ASN Inbound Container Delete (ASNINCTNCre)	Header only	Rib_ASNInRef_Rec

Primary inbound ASN tables

The following are the primary tables in RDM that hold ASN data:

- ASN
- ASN_ITEM
- CONTAINER
- CONTAINER_ITEM
- PO
- PO_DETAIL

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

Status and error messages

For status and error messages returned from the API, see the Operations Guide – Volume 2: Error Codes.

Stock Order subscription

Stock Order messages are used by RDM to create and maintain stock order, stock allocation, and component ticketing information. Stock Order messages are published by a Host system.

Stock Order messages represent a request for merchandise to be sent to another location. These messages provide information to the warehouse about the amount of each item that needs to be processed and shipped to the provided destination along with billing and shipping address information.

The Stock Order messages are specific to a particular warehouse instance and therefore contain routing information so that the bus can guarantee successful delivery of the message to the appropriate DC.

Stock order message structure

The Stock Order family of messages can create, modify and delete Stock Order records as well as create, modify, and delete Stock Orders details, Stock Allocation and Component Ticketing. All of the message types are composed of the following sections:

- Message header—This is data about the Stock Order including billing and shipping information, picking dates, and cartonization information.
- Allocation record – Requested Items, Destinations, and quantities.
- Component Ticketing record – Master and Component Item relationships.

Stock order message components

The following is a description of the Stock Order message components:

(For a general description of the ‘Consume’ and Process Message’ methods please refer to the preceding ‘RDM Message subscription process’ section of this document)

- Name of the ‘Consume’ methods: RDMSUB_SO
- Name of the ‘Process Message’ methods: SUB_SO

Message summary

All Stock Order messages belong to the Stock Order message family. The structure of the message depends on the message type to be performed. The following table lists each message, its structure and the associated Rib_Object.

Message	Structure	RIB_Object Type
Stock Order Create (SOCRE)	Header and Detail	Rib_SoDesc_Rec
Stock Order Modify (SOMOD)	Header only	Rib_SoDesc_Rec
Stock Order Delete (SODEL)	Header only	Rib_SoRef_Rec
Stock Allocation Create (SODCRE)	Header and Detail	Rib_SoDesc_Rec
Stock Allocation Modify (SODMOD)	Header and Detail	Rib_SoDesc_Rec
Stock Allocation Delete (SODDEL)	Detail only	Rib_SoDtIRef_Rec

Primary stock order tables

The following are the primary tables in RDM that hold Stock Order data:

- STOCK_ORDER
- STOCK_ALLOCATION
- COMPONENT_TICKETING
- STOCK_ALLOCATION_CID

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

Status and error messages

For status and error messages returned from the API, see the Operations Guide – Volume 2: Error Codes.

Outbound work order subscription

Outbound Work Order messages are used by RDM to create and maintain work order information. Outbound Work Order messages are published by a Host system.

Outbound Work Order messages represent a request to the warehouse to perform work on the merchandise before it is shipped to the stores or customers.

The Outbound Work Order messages are specific to a particular warehouse instance and therefore contain routing information so that the bus can guarantee successful delivery of the message to the appropriate DC.

Outbound Work Order message structure

The Outbound Work Order family of messages can create, modify and delete Outbound Work Order records. The message includes the following information: distro, destination, item, WIP sequence number, WIP code, personalization, instructions, order line number, and the auto complete flag.

Outbound work order message components

The following is a description of the Outbound Work Order message components:

(For a general description of the ‘Consume’ and Process Message’ methods please refer to the preceding ‘RDM Message subscription process’ section of this document)

- Name of the ‘Consume’ methods: RDMSUB_WOOUT
- Name of the ‘Process Message’ methods: SUB_WOOUT

Message summary

All Outbound Work Order messages belong to the Outbound Work Order message family. The structure of the message depends on the message type to be performed. The following table lists each message, it’s structure and the associated Rib_Object.

Message	Structure	RIB_Object Type
Outbound Work Order Create (OUTBDWOCre)	Header and Detail	Rib_WoOutDesc_Rec
Outbound Work Order Modify (OUTBDWOMod)	Header and Detail	Rib_WoOutDesc_Rec
Outbound Work Order Delete (OUTBDWODEl)	Header only	Rib_WoOutRef_Rec

Primary outbound work order tables

The following descriptions are for the primary tables in RDM that hold Outbound Work Order data:

- OUTBOUND_WORK_ORDER

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

Status and error messages

For status and error messages returned from the API, see the Operations Guide – Volume 2: Error Codes.

Pending returns subscription

Pending Return messages are used by RDM to create and maintain Pending Returns information. Pending Returns messages are published by a Host system.

Pending Returns messages represent a notification to the warehouse of merchandise that is being returned to the warehouse. These messages provide information to the warehouse about the amount of each item that is being returned.

The Pending Returns messages are specific to a particular warehouse instance and therefore contain routing information so that the bus can guarantee successful delivery of the message to the appropriate DC.

Pending returns message structure

The Pending Returns family of messages can create, modify and delete Pending Returns records as well as create, modify, and delete Pending Returns details. All of the message types are composed of the following sections:

- Message header—This is data about the RMA Number, PRO Number, Receipt Date.
- Detail record – The item and quantity.

Pending returns message components

The following is a description of the Pending Returns message components:

(For a general description of the ‘Consume’ and Process Message’ methods please refer to the preceding ‘RDM Message subscription process’ section of this document)

- Name of the ‘Consume’ methods: RDMSUB_PENDRETURN
- Name of the ‘Process Message’ methods: SUB_PENDRETURN

Message summary

All Pending Return messages belong to the Pending Returns message family. The structure of the message depends on the message type to be performed. The following table lists each message, its structure and the associated Rib_Object.

Message	Structure	RIB_Object Type
Pending Returns Create (PendRetCre)	Header and Detail	Rib_PendRtrnDesc_Rec
Pending Returns Modify (PendRetMod)	Header only	Rib_PendRtrnDesc_Rec
Pending Returns Delete (PendRetDel)	Header only	Rib_PendRtrnRef_Rec
Pending Returns Detail Create (PendRetDtlCre)	Detail only	Rib_PendRtrnDtlDesc_Rec
Pending Returns Detail Modify (PendRetDtlMod)	Detail only	Rib_PendRtrnDtlDesc_Rec
Pending Returns Detail Delete (PendRetDtlDel)	Detail only	Rib_PendRtrnDtlRef_Rec

Primary pending returns tables

The following are the primary tables in RDM that hold Pending Returns data:

- PENDING_RETURNS
- PENDING_RETURNS_DETAIL

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

Status and error messages

For status and error messages returned from the API, see the Operations Guide – Volume 2: Error Codes.

Publish components

Inbound ASN publish

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.

Inbound ASN tables

The RDM tables are populated when a record is recreated in RDM's screens. These are the tables that stage the ASN records to be published:

ASN_UPLOAD, ASN_ITEM_UPLOAD, ASN_CONT_UPLOAD, and ASN_PO_UPLOAD.

Inbound ASN message components

The following is a description of the Inbound ASN message components:

(For a general description of the 'GetNxt' and message builder methods please refer to the preceding 'RDM Message publication process' section of this document)

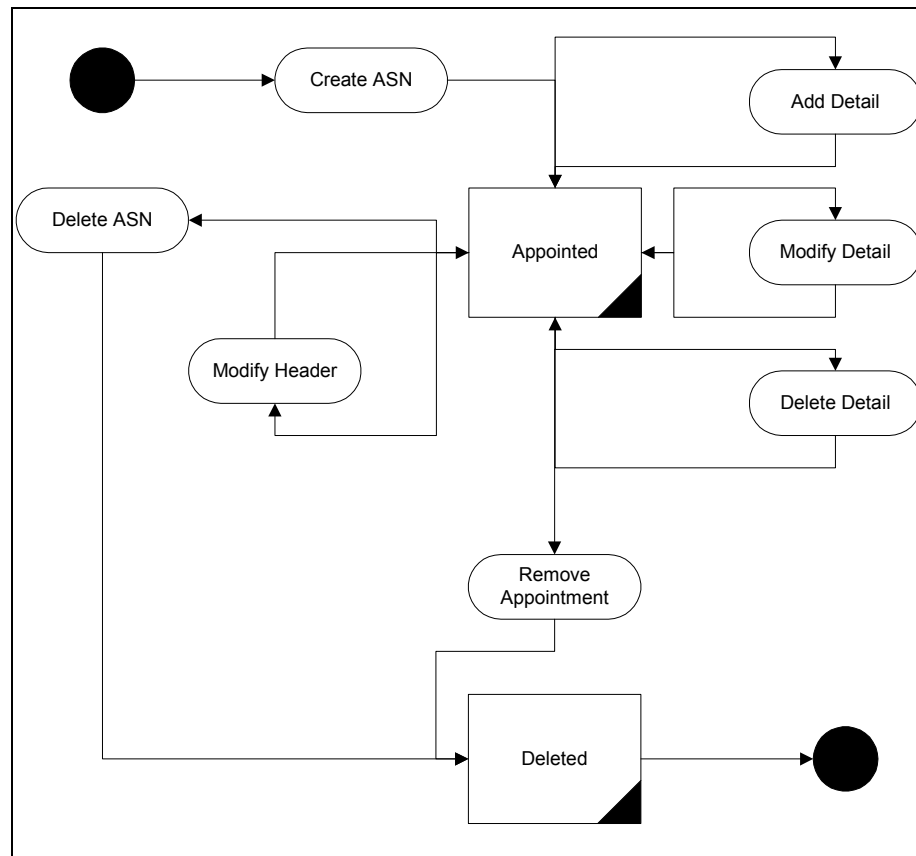
- Name of the 'GetNxt' methods: RDMMFM_ASNIN.getnxt
- Name of the message builder procedure: PUB_INBOUND_ASN

Message summary

All Inbound ASN messages belong to the Inbound ASN message family. The structure of the message depends on the message type to be performed. The following table lists each message, its structure and the associated Rib_Object.

Message	Structure	RIB_Object Type
Advance Shipping Notice Create (ASNInCre)	Header and Detail	Rib_ASNInDesc_Rec
Advance Shipping Notice Delete (ASNInDel)	Header only	Rib_ASNInRef_Rec

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

Status and error messages

For status and error messages returned from the API, see the Operations Guide – Volume 2: Error Codes.

Appointments/receipts publish

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 as 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.

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.

Receipt/appointment tables

The RDM tables are populated when a record is created in RDM. These are the tables that stage the Receipts and Appointments:

APPT_DETAIL_TO_UPLOAD, APPT_HEADER_TO_UPLOAD, and RECEIPT_TO_UPLOAD.

Receipt/appointment message components

The following is a description of the Receipt/Appointment message components:

(For a general description of the 'GetNxt' and message builder methods please refer to the preceding 'RDM Message publication process' section of this document)

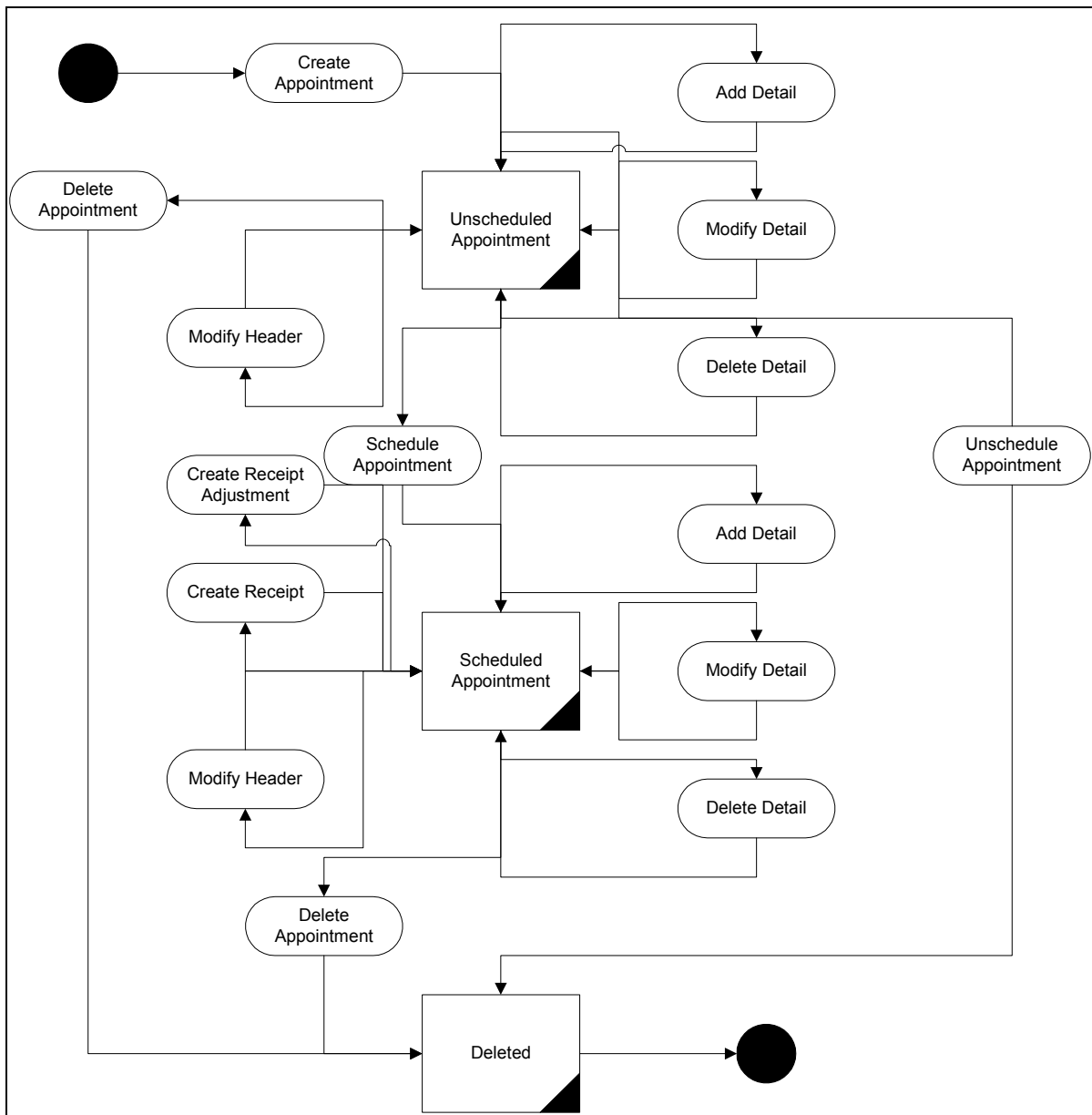
- Name of the 'GetNxt' methods: RDMMFM_RECEIVING.getnxt
- Name of the message builder procedures: PUB_RECEIPT and PUB_APPOINTMENT.

Message summary

All Receipt and Appointment messages belong to the Receipt message family. The structure of the message depends on the message type to be performed. The following table lists each message, it's structure and the associated Rib_Object.

Message	Structure	RIB_Object Type
Appointment Create (AppointCre)	Header and Detail	Rib_AppointDesc_Rec
Appointment Modify (AppointMod)	Header only	Rib_AppointDesc_Rec
Appointment Delete (AppointDel)	Header only	Rib_AppointRef_Rec
Appointment Detail Create (AppointDtlCre)	Header and Detail	Rib_AppointDesc_Rec
Appointment Detail Modify (AppointDtlMod)	Header and Detail	Rib_AppointDesc_Rec
Appointment Detail Delete (AppointDtlDel)	Header and Detail	Rib_AppointDtlRef_Rec

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

Status and error messages

For status and error messages returned from the API, see the Operations Guide – Volume 2: Error Codes.

Stock order status publish

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.

Stock order status tables

The RDM tables are populated when a record is created in RDM. This is the table that stage the Stock Order Status records to be published:

- STOCK_ORDER_INFO_UPLOAD

Stock order info upload message components

The following is a description of the Stock Order Status message components:

(For a general description of the 'GetNxt' and message builder methods please refer to the preceding 'RDM Message publication process' section of this document)

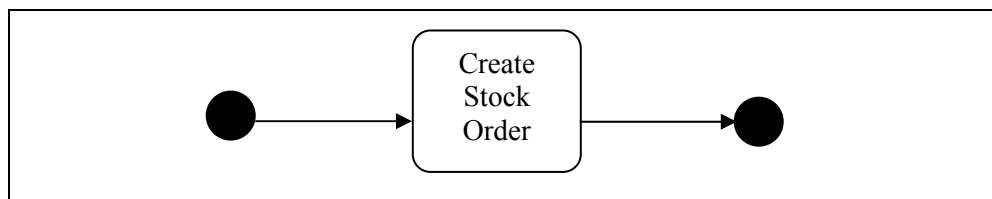
- Name of the 'GetNxt' methods: RDMMFM_SOSTATUS.getnxt
- Name of the message builder procedures: PUB_STOCKORDER_STATUS.

Message summary

All Stock Order Status messages belong to the Stock Order Status message family. The structure of the message depends on the message type to be performed. The following table lists each message, its structure and the associated Rib_Object.

Message	Structure	RIB_Object Type
Stock Order Create (SOStatusCre)	Header and Detail	Rib_SOStatusDesc_Rec

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

Status and error messages

For status and error messages returned from the API, see the **Operations Guide – Volume 2: Error Codes**.

Outbound ASN publish

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.

Outbound ASN tables

The RDM tables are populated when a record is recreated in RDM's screens. This is the table that stage the Outbound ASN records to be published:

- BOL_TO_UPLOAD

Outbound ASN message components

The following is a description of the Outbound ASN message components:

(For a general description of the 'GetNxt' and message builder methods please refer to the preceding 'RDM Message publication process' section of this document)

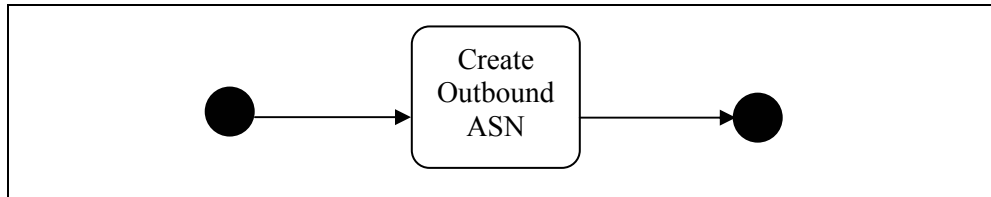
- Name of the 'GetNxt' methods: RDMMFM_ASNOUT.getnxt
- Name of the message builder procedures: PUB_OUTBOUND_ASN.

Message summary

All Outbound ASN messages belong to the Outbound ASN message family. The structure of the message depends on the message type to be performed. The following table lists each message, it's structure and the associated Rib_Object.

Message	Structure	RIB_Object Type
Outbound ASN Create (ASNOutCre)	Header and Detail	Rib_ASNOutDesc_Rec

State diagram



Description of activities

Create outbound ASN 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

Status and error messages

For status and error messages returned from the API, see the Operations Guide – Volume 2: Error Codes.

Inventory adjustments publish

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 (RIP)
- Available to Sell (ATS)
- 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

Inventory adjustments tables

The RDM tables are populated when a record is recreated in RDM's screens. This is the table that stage the Inventory Adjustment records to be published:

- INV_ADJUSTMENT_TO_UPLOAD

Inventory adjustment message components

The following is a description of the Inventory Adjustment message components:

(For a general description of the 'GetNxt' and message builder methods please refer to the preceding 'RDM Message publication process' section of this document)

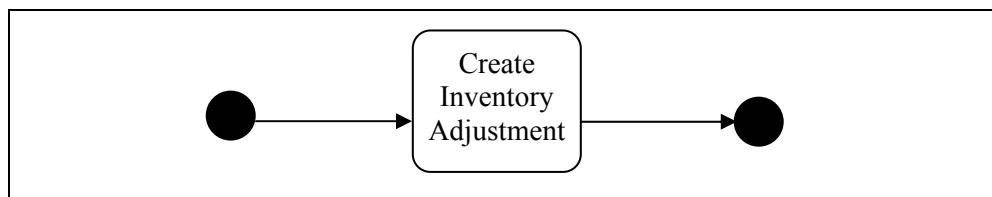
- Name of the 'GetNxt' methods: RDMMFM_INVADJUST.getnxt
- Name of the message builder procedures:
PUB_INVENTORY_ADJUSTMENT.

Message summary

All Inventory Adjustment messages belong to the Inventory Adjustment message family. The structure of the message depends on the message type to be performed. The following table lists each message, its structure and the associated Rib_Object.

Message	Structure	RIB_Object Type
Inventory Adjustment Create (InvAdjustCre)	Header and Detail	Rib_InvAdjustDesc_Rec

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

Status and error messages

For status and error messages returned from the API, see the **Operations Guide – Volume 2: Error Codes**.

Inventory balance upload

When requested, Retek Distribution Management uploads an image of the current inventory. The format of the inventory balance record is as follows:

Field Description	Template	Description
Location (DC)	X (10)	Destination ID of the DC.
Transaction Date/Time	YYYYMMDDHHMI	Date of run.
Item ID	X (25)	Item identifier.
Available Units	N (8) v N (4)	Units available for distribution.
Distributed Units	N (8) v N (4)	Units distributed includes: <ul style="list-style-type: none"> • Units distributed but not yet picked. • Units picked but not yet manifested. • Units manifested but not yet shipped.
Received Units	N (8) v N (4)	Units received but not put away.
Total Units	N (8) v N (4)	Sum of all units that physically exist: container status of: I, D, M, R, T, X.
Available Weight	N (8) v N (4)	Weight available for distribution of catch weight items.
Distributed Weight	N (8) v N (4)	Weight distributed includes: <ul style="list-style-type: none"> • Weight distributed but not yet picked. • Weight picked but not yet manifested. • Weight manifested but not yet shipped. Values only for catch weight items.
Received Weight	N (8) v N (4)	Weight received but not putaway for catch weight items.
Total Weight	N (8) v N (4)	Sum of all weight that physically exist: container status of: I, D, M, R, T, X. For catch weight items.

Customer returns publish

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.

Customer returns tables

The RDM tables are populated when a record is created in RDM. This is the table that stage the Customer Returns records to be published:

- RETURNS_UPLOAD

Customer returns message components

The following is a description of the Customer Returns message components:

(For a general description of the 'GetNxt' and message builder methods please refer to the preceding 'RDM Message publication process' section of this document)

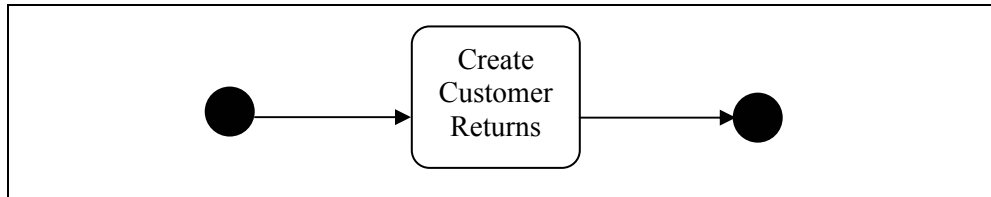
- Name of the 'GetNxt' methods: RDMMFM_CUSTRETURN.getnxt
- Name of the message builder procedures: PUB_CUSTOMER_RETURNS.

Message summary

All Customer Returns messages belong to the Customer Return message family. The structure of the message depends on the message type to be performed. The following table lists each message, it's structure and the associated Rib_Object.

Message	Structure	RIB_Object Type
Customer Return Create (CORetCre)	Header and details	Rib_CustRetDesc_Rec

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

Status and error messages

For status and error messages returned from the API, see the Operations Guide – Volume 2: Error Codes.

Return to vendor publish

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.

RTV tables

The RDM tables are populated when a record is posted in RDM. These are the tables that stage the Return to Vendor records to be published:

- INV_ADJUSTMENT_TO_UPLOAD
- RTV

Return to vendor message components

The following is a description of the Return to Vendor message components:

(For a general description of the 'GetNxt' and message builder methods please refer to the preceding 'RDM Message publication process' section of this document)

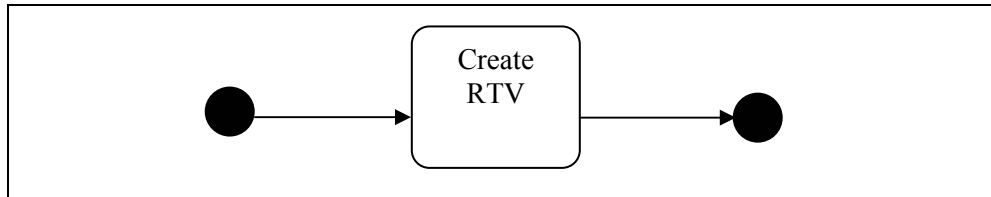
- Name of the 'GetNxt' methods: RDMMFM_RTV.getnxt
- Name of the message builder procedures: PUB_RETURN_TO_VENDOR.

Message summary

All Return To Vendor messages belong to the Return To Vendor message family. The structure of the message depends on the message type to be performed. The following table lists each message, it's structure and the associated Rib_Object.

Message	Structure	RIB_Object Type
Return To Vendor Create (RTVCre)	Header only	Rib_RTVDesc_Rec

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

Status and error messages

For status and error messages returned from the API, see the Operations Guide – Volume 2: Error Codes.

Streamsoft components

Space locations publish

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 a 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.

Space locations tables

The RDM tables are populated when a record is posted in RDM. This is the table that stage the Space Locations records to be published:

- LOCATION_UPLOAD

Space Location message components

The following is a description of the Space Location message components:

(For a general description of the 'GetNxt' and message builder methods please refer to the preceding 'RDM Message publication process' section of this document)

- Name of the 'GetNxt' methods: RDMMFM_SPACELOCS.getnxt
- Name of the message builder procedures: PUB_SPACE_LOCATION.

Message summary

All Space Locations messages belong to the Space Location message family. The structure of the message depends on the message type to be performed. The following table lists each message, its structure and the associated Rib_Object.

Message	Structure	RIB_Object Type
Space Locations Create (SpaceLocsCre)	Header and Details	Rib_SpaceLocsDesc_Rec
Space Locations Modify (SpaceLocsMod)	Header and Details	Rib_SpaceLocs_Desc_Rec
Space Locations Delete (SpaceLocsDel)	Header only	Rib_SpaceLocsRef_Rec

Status and error messages

For status and error messages returned from the API, see the **Operations Guide – Volume 2: Error Codes**.

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 SKU Optimization family of messages can create Task Queue records. Records contain location, group number, move number, sequence number and item information.

SKU optimization message components

The following is a description of the SKU Optimization message components:

(For a general description of the ‘Consume’ and Process Message’ methods please refer to the preceding ‘RDM Message subscription process’ section of this document)

- Name of the ‘Consume’ methods: RDMSUB_SKUOPTM
- Name of the ‘Process Message’ methods: SUB_SKUOPTM

Message summary

All SKU Optimization messages belong to the SKUOptm message family. The structure of the message depends on the message type to be performed. The following table lists each message, its structure and the associated Rib_Object.

Message	Structure	RIB_Object Type
SKU Optimization Create (SKUOptmCre)	Header only	Rib_SKUOptmDesc_Rec

Primary SKU optimization 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 Data Model document.

Status and error messages

For status and error messages returned from the API, see the Operations Guide – Volume 2: Error Codes.

Item warehouse publish

RDM is responsible for communicating items that reside in a forward picking / forward case picking location 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 a allocation.

The information sent to the third-party system for add or modify requests includes: 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.

Item tables

The following table is populated when the item information is published by RDM.

- ITEM_MASTER_UPLOAD

Item warehouse message components

The following is a description of the Item Warehouse message components:

(For a general description of the 'GetNxt' and message builder methods please refer to the preceding 'RDM Message publication process' section of this document)

- Name of the 'GetNxt' methods: RDMMFM_ITEMWH.getnxt
- Name of the message builder procedures: PUB_ITEMWH.

Message summary

All Item Warehouse messages belong to the Item Warehouse message family. The structure of the message depends on the message type to be performed. The following table lists each message, its structure and the associated Rib_Object.

Message	Structure	RIB_Object Type
Item Warehouse Create (ItemWHCre)	Header and details	Rib_ItemWHDesc_Rec
Item Warehouse Modify (ItemWHMod)	Header and detail	Rib_ItemWHDesc_Rec
Item Warehouse Delete (ItemWHDel)	Header only	Rib_ItemWHRef_Rec

Status and error messages

For status and error messages returned from the API, see the Operations Guide – Volume 2: Error Codes.

Retek Labor Management (RLM) components

Transaction publish

RDM is responsible for communicating Transaction Information to the Labor Management System. Transaction data is a variety of data sent to the host to help keep the systems in unison.

Transaction messages are communicated to the Labor Management System once the Transaction data has been added or modified in RDM's system.

Transaction tables

The RDM tables are populated when a record is recreated in RDM's screens. This is the table that stage the Transaction records to be published:

- TRANSACTION_UPLOAD

Warehouse transaction message components

The following is a description of the Warehouse Transaction message components:

(For a general description of the 'GetNxt' and message builder methods please refer to the preceding 'RDM Message publication process' section of this document)

- Name of the 'GetNxt' methods: RDMMFM_WHTRANS.getnxt
- Name of the message builder procedures: PUB_WAREHOUSE_TRANS.

Message summary

All Warehouse Transaction messages belong to the Warehouse Transaction message family. The structure of the message depends on the message type to be performed. The following table lists each message, it's structure and the associated Rib_Object.

Message	Structure	RIB_Object Type
Warehouse Transaction Create (WhTransCre)	Header only	Rib_WHTransDesc_Rec

Status and error messages

For status and error messages returned from the API, see the Operations Guide – Volume 2: Error Codes.

Location publish

RDM is responsible for communicating Location Information to the Labor Management System. Location is defined as Locations originating in the RDM System

Location messages are communicated to the Host once the Location has been added, modified, or deleted out of RDM's system.

Location tables

The RDM tables are populated when a record is recreated in RDM's screens. This is the table that stage the Location records to be published:

- LMS_LOCATION_UPLOAD

Location message components

The following is a description of the Location message components:

(For a general description of the 'GetNxt' and message builder methods please refer to the preceding 'RDM Message publication process' section of this document)

- Name of the 'GetNxt' methods: RDMMFM_WHSPACELOCS.getnxt
- Name of the message builder procedures:
PUB_WAREHOUSE_SPACE_LOCS.

Message summary

All Warehouse Space Location messages belong to the Warehouse Space Location message family. The structure of the message depends on the message type to be performed. The following table lists each message, it's structure and the associated Rib_Object.

Message	Structure	RIB_Object Type
Warehouse Space Locations Create (WhSpaceLocsCre)	Header only	Rib_WHSpaceLocsDesc_Rec
Warehouse Space Locations Modify (WhSpaceLocsMod)	Header only	Rib_WHSpaceLocsDesc_Rec
Warehouse Space Locations Delete (WhSpaceLocsDel)	Header only	Rib_WHSpaceLocsRef_Rec

Triggers

There is a trigger on the location table, which populates the lms_location_upload table with the upload info for all updates/inserts to the location table.

Status and error messages

For status and error messages returned from the API, see the Operations Guide – Volume 2: Error Codes.

Equipment publish

RDM is responsible for communicating Equipment Information to the Labor Management System. Equipment is defined as Equipment originating in the RDM System

Equipment messages are communicated to the Labor Management System once the equipment has been added, modified, or deleted out of RDM's system.

Equipment tables

The RDM tables are populated when a record is recreated in RDM's screens. This is the table that stage the Equipment records to be published:

- EQUIPMENT_UPLOAD

Equipment message components

The following is a description of the Equipment message components:

(For a general description of the 'GetNxt' and message builder methods please refer to the preceding 'RDM Message publication process' section of this document)

- Name of the 'GetNxt' methods: RDMMFM_WHEQUIP.getnxt
- Name of the message builder procedures: PUB_WAREHOUSE_EQUIP.

Message summary

All Warehouse Equipment messages belong to the Warehouse Equipment message family. The structure of the message depends on the message type to be performed. The following table lists each message, it's structure and the associated Rib_Object.

Message	Structure	RIB_Object Type
Warehouse Equipment Create (WhEquipCre)	Header only	Rib_WHEquipDesc_Rec
Warehouse Equipment Modify (WhquipMod)	Header only	Rib_WHEquipDesc_Rec
Warehouse Equipment Delete (WhEquipDel)	Header only	Rib_WHEquipRef_Rec

Triggers

There is a trigger on the equipment table, which populates the equipment_upload table with the upload info for all updates/inserts to the equipment table.

Status and error messages

For status and error messages returned from the API, see the Operations Guide – Volume 2: Error Codes.

Equipment class publish

RDM is responsible for communicating Equipment Class Information to the Labor Management System. Equipment Class is defined as a class grouping of equipments.

Equipment Class messages are communicated to the Labor Management System once the Equipment Class has been added, modified, or deleted out of RDM's system.

Equipment class tables

The RDM tables are populated when a record is recreated in RDM's screens. This is the table that stage the Equipment Class records to be published.

- EQUIPMENT_CLASS_UPLOAD

Equipment class message components

The following is a description of the Equipment Class message components:

(For a general description of the 'GetNxt' and message builder methods please refer to the preceding 'RDM Message publication process' section of this document)

- Name of the 'GetNxt' methods: RDMMFM_WHEQUIPCLS.getnxt
- Name of the message builder procedures: PUB_WAREHOUSE_EQUIPCLS.

Message summary

All Warehouse Equipment Class messages belong to the Warehouse Equipment Class message family. The structure of the message depends on the message type to be performed. The following table lists each message, its structure and the associated Rib_Object.

Message	Structure	RIB_Object Type
Warehouse Equipment Class Create (WhEquipClsCre)	Header only	Rib_WHEquipClsDesc_Rec
Warehouse Equipment Class Modify (WhEquipClsMod)	Header only	Rib_WHEquipClsDesc_Rec
Warehouse Equipment Class Delete (WhEquipClsDel)	Header only	Rib_WHEquipClsRef_Rec

Triggers

There is a trigger on the equipment_class table which populates the equipment_class_upload table with the upload info for all updates/inserts to the equipment_class table.

Status and error messages

For status and error messages returned from the API, see the Operations Guide – Volume 2: Error Codes.

Chapter 4 – Subsystem interfaces

Batch file formats

All batch files passed between an outside system and Retek Distribution Management consist of one or more records in the upload or download files. These records contain printable ASCII characters (with space characters between each field) and are of a fixed length based on the transaction type.

Fields that are defined within transaction records have an associated template that defines the arrangement, length, and logical content of the field. They appear as one of the following types:

Template	Meaning
A	A character data type.
N	A numbered digit (0 through 9).
N(p)	An unsigned p-digit number.
N(p)vN(q)	A fixed point number with a decimal point, p digits to the left of the decimal and q digits to the right.
sN(p)	A p-digit number that has a sign ('+' or '-') as its first significant character.
X	An alphanumeric character.
X(p)	A p-character string.
YYYYMMDDHHMM	A date/time, with a 4-digit year followed by a 2-digit month followed by a 2-digit day followed by a 2-digit hour, a 24 hour format, followed by a 2-digit minute.

Note: Numeric fields are always right justified with leading zeros. Character fields are left justified with trailing blanks, unless otherwise stated.

Unit pick system files

Allocation data download

This file specifies the outstanding store orders to be fulfilled to the Unit Pick System.

Field Description	Template	Destination
facility id (dc)	X (2)	Code for the DC
unit pick system code	X (4)	Code for Unit Pick System
Wave number	N (3)	Unique identifier of wave
item id	X (25)	Unique identifier of the item.
dest id	X (10)	Identifier of the ship destination.
unit qty	N (8) v N (4)	Number of units
logical chute	X (10)	Logical chute assigned to group
Group id	N (4)	Identifier for a set of orders
Slot	N (3)	Identifier with a group associated to an order
to container id	X (20)	System generated container ID merchandise is to be packed into

Inbound carton download

This file specifies the carton content and the associated wave to the Unit Pick System.

Field Description	Template	Destination
facility id (dc)	X (2)	Code for the DC
unit pick system code	X (4)	Code for Unit Pick System
Wave number	N (3)	Unique identifier of wave
container id	X (20)	Unique identifier of the source container.
item id	X (25)	Unique identifier of the item.
requested unit qty	N (8) v N (4)	Number of units

Process UPS upload

This file serves as a notification from a Unit Pick System to RDM concerning contents of a picked container, the associated wave number and the outbound destination ID.

Field Description	Template	Destination
Facility ID (DC)	X (2)	Code for the DC
Transaction Date/Time	YYYYMMDD HH24MI	Date and time this record was created.
Wave Number	N (3)	Unique identifier of wave
Container ID	X (20)	Unique identifier of the container.
Item ID	X (25)	Unique identifier of the item
Distributed Unit Qty	N (8) v N (4)	Number of distributed units
Dest ID	X (10)	Identifier of the ship destination.

Pick by light interface

The Pick By Light system (PBL) requires a variety of information from a host in order to drive its paperless picking processes. These transactions are sent periodically; the frequency is determined by the urgency of the transaction type. The host can be either Retek Distribution Management or, as in standalone operations, some other application. Data is exchanged through text files. With text file data exchange, PBL does not need to be concerned with the specifics of how the files were created or how they arrived in the upload or download directories. Each customer selects an approach to suit the preferred communication methods.

Files and directories

All download files are placed in a directory that is named by the UNIX environment variable "DOWNLOAD_DIR". All upload files are placed in a directory that is named by the UNIX environment variable "UPLOAD_DIR".

The download and upload files have set names as listed in the following table. They are listed in the order in which they should be run because each download may depend upon a previous one.

Interface Name	Script Name	File Name
Destination Container Download	dest_container_download.sh	dest_container_download.dat dest_cont_item_download.dat
Distribution Item Download	distro_item_download.sh	distro_item_download.dat
Inventory Adjustment Download	inv_adj_download.sh	inv_adj_download.dat
Ship Destination Download	pps_ship_dest_upload.sh	ship_dest_upload.dat
Distro Item Upload	create_distro_item_upload.sh	distro_item_upload.dat
Expected Source Container Upload	create_exp_container_upload.sh	exp_container_upload.dat
Source Container Upload	generate_source_container_upld.sh	source_container_upload.dat

Download transactions

The PBL downloads include several fields that are future use. These fields are included to allow for the future growth in Retek Distribution Management and to allow the PBL to work standing alone, without Retek Distribution Management. PBL download errors are recorded in the local Retek Distribution Management error log, and are not uploaded to the host. You can view and maintain this log in the Error Log screen.

Destination container download

The Destination Container download files are built by PBL for use by Retek Distribution Management. They contain PBL built containers and the items and quantities in them that are going to be added back to inventory or shipped. If the destination is marked as the DC, the container is sent to stock; otherwise, a distribution is assumed and the container is routed appropriately. When PBL has finished creating the files, they are first copied to the download directory by PBL. Then, the script in Retek Distribution Management for this download is started by PBL. The script reads the files, loads the data into Retek Distribution Management, and adds the container information to Retek Distribution Management.

The Destination Container Download consists of a Header file and a Detail file.

The Header file, which describes the container, has the following format:

Field Description	Template	Description
Transaction Date/Time	YYYYMMDD HH24MISS	Date and time this record was created (future use).
Record Type	A	Record type in PBL. Always a 'Z' for a Destination Container Download header record (future use).
Facility ID	X (2)	Identifier for the facility.
Company Number	N (1)	Company Number (future use).
Destination ID	X (10)	Identification of the ship destination.
Container ID	X (20)	Identifier for the container.
Destination Name	X (30)	Descriptive name of the ship destination (future use).
Address 1	X (30)	First address line of the ship destination (future use).
Address 2	X (30)	Second address line of ship destination (future use).
Address 3	X (30)	Third address line of the ship destination (future use).

The Detail file, which describes the contents of the closed picking container, has the following format:

Field Description	Template	Description
Transaction Date/Time	YYYYMMDD HH24MISS	Date and time this record was created (future_use).
Record Type	A	Record type in PBL. Always a 'Y' for a Destination Container Download detail (future use).
Facility ID	X (2)	Identifier for the facility.
Container ID	X (20)	Identifier for the container.
Distribution/Order Number	X (10)	Identifier for the distribution or order.
Item ID	X (25)	Identifier for the item.
Unit Qty	N (8) v N (4)	Unit quantity that was picked for this item.
Item Description	X (60)	Text description of the item (future use).

Errors due to data integrity with the download are recorded in the error log and the record is ignored. Possible errors include:

- Facility ID does not exist in Retek Distribution Management.
- Container already exists in Retek Distribution Management.
- Non-existent Destination ID.
- Duplicate Item ID/Distro Nbr (or Item/Order) on the container detail.
- Non-existent Item ID.

Distribution item download

The Distribution Item Download file is built by PBL and sent to Retek Distribution Management. Therefore, pick directive records can be deleted and stock allocations adjusted as needed. When PBL has finished creating the files, they are first copied to the download directory by PBL. Then, the script in Retek Distribution Management for this download is started by PBL. The script reads the file, loads the data into Retek Distribution Management, and updates the picking information in Retek Distribution Management as required.

The format for the Distribution Item Download is as follows:

Field Description	Template	Description
Transaction Date/Time	YYYYMMDD HHMISS	Date and time this record was created (future use).
Record Type	A	Record type in PPS. Always a 'X' for a Destination Container Download header record (future use).
Facility ID	X (2)	Identifier for the facility.
Company Number	N (1)	Company Number (future use).
Distro Number	X (10)	Identifier for the distribution or order.
Item ID	X (25)	Identifier for the item.
Destination ID	X (10)	Identifier for the shipping destination.
Requested Unit Qty	N (8) v N (4)	Number of units of this item requested for picking.
Distributed Unit Qty	N (8) v N (4)	Number of units of this item actually picked.

Errors due to data integrity with the download are recorded in the error log and the record is ignored. Possible errors include:

- Non-existent Facility ID.
- Non-existent Destination ID.
- Non-existent Item ID.
- No pick for the distro/item/destination.

Inventory adjustment download

The Inventory Adjustment Download file is built by PBL and sent to Retek Distribution Management when there is a difference between the quantity sent on the Source Container Upload and the actual quantity picked. RDM validates the data in the file and sends the information in an Inventory Adjustment Upload to the host system. This is the only action RDM takes on this; no change in RDM data occurs. After PBL creates the files, they are copied to the download directory by PBL. Then, PBL starts the script in Retek Distribution Management, or this download. The script reads the files, validates the data, and inserts the information into the Inventory Adjustment Upload table in RDM for upload to the host (reason code to the host for this adjustment is 30).

The format for the Inventory Adjustment Download is as follows:

Field Description	Template	Description
Transaction Date/Time	YYYYMMDD HHMISS	Date and time this record was created (future use).
Record Type	A	The record type in PBL. This is always sent as 'W' for an Inventory Adjustment Download (future use).
Facility ID	X (2)	Identifier for the facility.
Company Number	N (1)	A single digit number for the company. This is a new system parameter in Retek Distribution Management (future use).
Distro Number	X (10)	The identifier for the distribution (future use).
Item ID	X (25)	The identifier for the item.
Adjusted Unit Qty	sN (8) v N (4)	The difference between source container units and the number of units of this item actually picked. A positive number means more were picked than expected; a negative number means fewer were picked than expected.

Errors due to data integrity with the download are recorded in the error log and the record is ignored. Possible errors include:

- Non-existent Item ID.

Upload transactions

Ship destination upload

The Ship Destination Upload file is spooled from the Shipping Destination table and sorted by Facility ID, Company Number, and Shipping Destination. This file is empty unless an adjustment action (add/modify/delete) is sent to Retek Distribution Management from the host, or performed in the Destination Editor screen. Whenever an adjustment is performed, all shipping destinations that Retek Distribution Management knows about are sent to PBL via the upload. Thus, this upload is an all or nothing data file.

The format for the Ship Destination Upload is as follows:

Field Description	Template	Description
Transaction Date/Time	YYYYMMDD HHMISS	Date and time this record was created.
Record Type	A	Record type in PBL. Always a 'A' for a Ship Destination Upload.
Facility ID	X (2)	Identifier for the facility.
Company Number	N (1)	Company Number.
Destination ID	X (10)	Identification of the ship destination.
Destination Name	X (30)	Descriptive name of the ship destination.
Address 1	X (30)	First address line of the ship destination.
Address 2	X (30)	Second address line of ship destination.
Address 3	X (30)	Third address line of the ship destination.

Distro item upload

The Distro Item Upload file contains records that indicate to PBL which items, and how many, should be shipped to specified destinations. After the distribution process runs, this file is built from all remaining sorted allocation records that are eligible to be processed by PBL (the item does not have a forward picking location defined). Records are sorted by facility number, company number, distribution/order number, item, distro/order creation time stamp, and shipping destination.

The format for the Distro Item Upload is as follows:

Field Description	Template	Description
Transaction Date/Time	YYYYMMDD HHMISS	Date and time this record was created.
Record Type	A	Record type in PBL. Always a 'B' for a Distro Item Upload.
Facility ID	X (2)	Identifier for the facility.
Company Number	N (1)	Company Number.
Distro Number	X (10)	Identifier for the distribution or order.
Item ID	X (25)	Identifier for the item.
Distro Create Date/Time	YYYYMMDD HHMISS	Date and time the distro/order was created.
Destination ID	X (10)	Identifier for the shipping destination.
Unit Qty	N (8) v N (4)	Number of units of this item to be shipped.
Item Dept	X (4)	Department of the item.
Item Description	X (60)	Item Description.

Expected source container upload

This Expected Source Container Upload file contains records identifying all Inventory containers necessary to fulfill the PBL requirements determined by the last distribution run. This information is used by PBL to "know" ahead of time what containers will be needed by PBL. This file is built after each distribution run. Records are sorted by facility number, company number, distribution/order number, item, distro/order creation time stamp, and container ID.

The format of the Expected Source Container Upload is as follows:

Field Description	Template	Description
Transaction Date/Time	YYYYMMDD HH24MISS	Date and time this record was created.
Record Type	A	Record type in PBL. Always a 'C' for an Expected Source Container Upload.
Facility ID	X (2)	Identifier for the facility.
Company Number	N (1)	Company Number.
Distro Number	X (10)	Identifier for the distribution or order.
Item ID	X (25)	Identifier for the item.
Distro Create Date/Time	YYYYMMDD HH24MISS	Date and time the distro/order was created.
Container ID	X (20)	Identifier for the container.
Requested Qty	N (8) v N (4)	Unit quantity that will be requested for picking of this item.

Source container upload

The Source Container Upload file is built as PBL Source containers are picked and dropped off at the PBL staging area. The upload file is used to match up expected containers with actual source containers delivered to PBL. It has no sorted order.

Note: The value of Actual Quantity in the Upload will be 0 (zero) if the pick was 'canceled' either by the user or indirectly via a system function (such as a location marked for cycle count).

The format for the Source Container Upload is as follows:

Field Description	Template	Description
Transaction Date/Time	YYYYMMDD HH24MISS	Date and time this record was created.
Record Type	A	Record type in PBL. Always a 'D' for a Source Container Upload.
Facility ID	X (2)	Identifier for the facility.
Container ID	X (20)	Identifier for the container.
Actual Qty	N (8)	Unit quantity in the container.

Sortation subsystem interface

Due to the increased use of UCC-128 labeled containers and the addition of WIP code functionality to Retek Distribution Management, the Retek Distribution Management Sortation module will now send "container divert instruction" messages to the sortation system to control the flow of containers on the conveyor.

Each message that Retek Distribution Management sends to the sortation system informs it of the next logical destination for a container. The divert instruction could be, but is not limited to, one of the following: any type of processing area, QA sampling, palletization, putaway staging, or shipping lane divert instructions. Initially, a message is sent to the sorter whenever a container is created on Retek Distribution Management. However, subsequent messages can be sent to the sorter if the container is assigned one or more WIP codes. The sortation system is only sent the next logical destination for a container.

The sortation system continues to notify Retek Distribution Management of any container diverts that occur on the conveyor system. Depending on the type of divert that has taken place, Retek Distribution Management either attempts to auto-receive, move, or manifest the container.

Files and directories

All files are placed in a directory that is named by the UNIX environment variable 'SORTATION_DIR'. The files have set names as listed below. They are listed in the order in which they should be run because each download may depend upon a previous one.

Interface Name	Script Name	File Name
Container Divert Download	sorter_dnld.sh	sorter_dnld.dat
Container Divert Instruction Upload	sorter_upload.sh	sorter_upload.dat

Download transactions

Container divert message

The sortation system sends a message when a container is scanned, indicating whether it was scanned as an induction, diverted to a processing area, or diverted to a shipping lane. If the container was inducted, Retek Distribution Management performs an auto-receiving function. If the container is diverted to a processing area, Retek Distribution Management updates the location of the container. When a divert to a shipping lane is sent, Retek Distribution Management adds the container to the manifest for the trailer if one is available. For more details, refer to the Retek Distribution Management Shipping Module in the *Retek Distribution Management User Guide*.

The Container Divert Message Download file has the following format:

Field Description	Template	Description
Container ID	X (20)	Identifier of the container.
Divert Type	A	I = Induction D = Shipping Lane Divert.
Logical Destination	X (4)	Area of the DC to which the container was sorted.
Tracking ID	X (25)	Tracking ID (if any) applied to the container by a carrier for tracking purposes.
Divert Timestamp	YYYYMMDD HH24MISS	Date/Time the container was scanned by the sortation system.

Upload transactions

Container divert instruction message

Retek Distribution Management sends a Container Divert Instruction Message to the sortation system to control the flow of containers on the conveyor system. If a container must be diverted to several areas in the distribution center before it is ready to be putaway or shipped, Retek Distribution Management will only inform the sortation system of the next logical destination for the container. This way, the sortation system does not need to keep track of all divert instructions for a container. The first divert instruction for a container is sent when the container ID is first created on Retek Distribution Management.

When the receiving allocation module creates a container, Retek Distribution Management calculates a pallet group identifier in order to give a palletization operator a quickly recognized code that helps to group cartons together on pallets. Retek Distribution Management assigns a four-digit number to each PO/Item/Destination and concatenates this with the total number of cartons expected for the pallet group to make up the pallet group identifier.

The Container Divert Instruction Message Upload file has the following format:

Field Description	Template	Description
Container ID	X (20)	Container identifier.
Logical Destination	X (4)	Next area of DC to which the container should be diverted.
Transaction Date/Time	YYYYMMDD HHMISS	Date/Time of upload to sortation system.

Manifest mailing system

The Manifest Mailing System uses the merchandise carton ID to form an Oracle Data Base Compliant (ODBC) query into the Retek Distribution Management data base. This query gathers the information necessary for generating a shipping label and manifesting the carton.

Merchandise, planned and picked, using the logic currently implemented in Retek Distribution Management, is first taken to a shipping station. Each shipping station is a PC running a manifest mailing system with interfaces to a user interface terminal and often to a scale.

The label applied by the Retek Distribution Management picker is then scanned to retrieve the carton ID necessary for the ODBC query.

Files and directories

In addition to the normal BOL upload records, MMS information is uploaded to the host. This additional information is prepared for upload to the host system upon completion of the normal BOL upload operation.

Each Container ID, pro number combination in the shipment has one detail record in the MMS upload. The BOL Sequence Number is incremental and unique for each BOL.

An MMS upload consists of a single detail file with the following format:

Field Description	Template	Description
Batch Number	N (7)	Numeric Sequence of the upload
BOL Number	X (17)	BOL number
Container ID	X (20)	Identifier of container
Pro Number	X (18)	Shipper's tracking number
Cube	N (10) v N (2)	Container cube
Weight	N (8) v N (4)	Container weight
Freight	N (6) v N (2)	Freight charge
Markup	N (6) v N (2)	Markup charge
Charge type	X (6)	Carrier charge code
Service Code	X (6)	Carrier service code
Service Level	X (12)	Carrier service level
Tracking ID	X (25)	Tracking ID

Views

The Retek Distribution Management data base, which the Manifest Mailing System queries is actually two views: the MMS Container View and the MMS Container Item View.

MMS container view

The value in the "CARRIER_SERVICE_CODE" is set by the host system, or it could be blank. If the value is blank, the user must input data at the shipping station. The Manifest Mailing System can change the value of the CARRIER_SERVICE_CODE, even if the field is not blank.

Retek Distribution Management downloads the SHIP_TO_ADDRESS with the stock order. If this field is blank in the order download, the information will be supplied by the SHIP_DEST table. The Manifest Mailing System can change the value of the SHIP_TO_ADDRESS, even if the field is not blank.

The format of the MMS Container view is as follows:

Field Description	Template	Description
Facility ID	X (2)	Identifier for the facility
Container ID	X (20)	Identifier for the container
Ship Address Description	X (30)	The description (such as store or ship-to name). This is the first line of the address block.
Ship Address1	X (30)	Shipping Address Line 1
Ship Address2	X (30)	Shipping Address Line 2
City	X (25)	Shipping City
State	X (3)	Shipping State
Zip	X (10)	Shipping Zip
Dest ID	X (10)	Destination identifier
Carrier Service Code	X (6)	Carrier service code for the delivery (such as First Class)
Bill Address Description	X (30)	The first line of the address block. A description, such as company or bill-to name.
Bill Address1	X (30)	Billing Address line 1
Bill Address2	X (30)	Billing Address line 2
Bill Address3	X (30)	Billing Address line 3
Amount1	N (8) v N (4)	Amount Charge 1
Amount2	N (8) v N (4)	Amount Charge 2
Amount3	N (8) v N (4)	Amount Charge 3

Field Description	Template	Description
DL Comment	X (30)	Download comment that will be printed on the label (optional)

Note: A default value using 'nvl' or decode statements should be supplied for any null values.

MMS container item view

The data item called "DISTRO" is analogous to "CUSTOMER_ORDER_NUMBER" in a wholesale system.

The format of the Container Item View is as follows:

Field Description	Template	Description
Facility ID	X (2)	Identifier for the facility
Container ID	X (20)	Container identifier
Item ID	X (25)	Unique item identifier
Unit Qty	N (8) v N (4)	Standard unit quantity for an item
Weight	N (8) v N (4)	Item weight or unit quantity weight
Retail Price	N (16) v N (4)	Retail selling price
Class	X (7)	Class of merchandise (optional)
Distro/Order	X (10)	Unique identifier of a distribution or order
Ticket Type	X (4)	Refers to Ticket Type table (optional)

Tables

The Manifest Mailing System populates two tables: the MMS Manifest Table and the MMS Container Table.

MMS manifest table

Retek Distribution Management and MMS work congruently to generate each manifest. Before a new grouping of containers is started, PRO_NBR (pickup) numbers are assigned by MMS. Next, a row is inserted into the MMS MANIFEST table and the STATUS is set to 'OPEN'. When MMS inserts a row into the MMS MANIFEST table, Retek Distribution Management looks up the corresponding BOL number in the RETEK DISTRIBUTION MANAGEMENT MANIFEST table. If the BOL does not exist, Retek Distribution Management inserts a MANIFEST record with a new system-generated BOL number (PRO_NBR=PICK_UP_NBR, DEST_ID = (SCP Mixed_dest_id)).

When a shipment is released by MMS and/or MMS is ready to have a BOL uploaded, the 'STATUS' in the MMS_MANIFEST table is updated to 'SHIPPED'. Retek Distribution Management recognizes the status change and administers normal ship operations (adjust manifest, set container_status = 'S' for all containers for BOL (pick_up_nbr), upload BOL).

The MMS_MANIFEST table format is as follows:

Field Description	Template	Description
Facility ID	X (2)	Identifier for the facility
PRO NBR	X (18)	Pickup number
MANIFEST STATUS	X (10)	Status of manifest: <ul style="list-style-type: none"> • OPEN: a new grouping of containers is started • SHIPPED: shipment is released
Trailer ID	X (12)	Identifier for trailer
Carrier Code	X (4)	Code of the carrier for the order

Note: If TRAILER_ID does not exist, create TRAILER and CARRIER records.

Set:

TRAILER_STATUS = 'LOADING'

TRAILER_CUBE=SCP

'default_trailer_cube'=CARRIER_CODE

CARRIER_NAME=CARRIER_CODE

MMS container table

Operators scan the containers and the Manifest Mailing System writes records to the MMS_CONTAINER table. Retek Distribution Management uses this information to set the CONTAINER_STATUS to 'M' (manifested) and update the CUBE and WEIGHT for each record inserted into the MMS_CONTAINER table.

The MMS_CONTAINER table is as follows:

Field Description	Template	Description
FACILITY ID	X (2)	Identifier for the container
CONTAINER ID	X (20)	Identifier for the container
PRO NBR	X (18)	Pickup number
CONTAINER CUBE	N (6) v N (2)	Container cube (dimensionless)
CONTAINER WEIGHT	N (4) v N (3)	Container weight (dimensionless)
FREIGHT CHARGE	N (6) v N (2)	Freight charge.
MARKUP CHARGE	N (6) v N (2)	Markup charge.
CHARGE TYPE	X (6)	Billing Method (such as COD and 30-day invoice).
CARRIER SERVICE CODE	X (6)	Carrier service code for the delivery (such as First class).
SERVICE LEVEL	X (12)	Code for shipment (Next Day, Second Day Air).
TRACKING ID	X (25)	Tracking Identifier
CREATION TS	YYYYMMDDHH24MI	Date and Time record created

Kewill shipping system interface

RDM interfaces with Kewill, shipping and transportation management system. As outbound cartons are created in RDM during the distribution process, information about them will be sent to Kewill so that Kewill can prepare a shipping label and other rate information. Kewill will send RDM the shipping label, which will be stored as a database field.

Kewill provides a shipping system for managing the use of common carriers (i.e. Fed-Ex, UPS, etc.) This interface will include the TCP/IP sockets layer between RDM and the Kewill K-Ship shipping information such as name, shipping address, package dimension, and estimated weight to the Kewill K-Ship system. K-Ship will respond with the estimated rate, tracking number, and label information to be stored in RDM database. When the actual weight is determined, RDM will send a message to K-Ship, and a response is returned with the actual rate and tracking number to be loaded into the RDM database. Once the package is actually loaded to be shipped, RDM will send a message to K-Ship so that it can update its manifest information.

Triggers

ship_carton_trg

This new trigger will notify Kewill when the status of outbound cartons have been updated to 'S'hipped.

create_sorter_instructions_trg

When the Kewill interface is enabled in RDM, this trigger will call the socket interface package, on the downloads from RDM to Kewill.

Packages

label_info_received

This stored procedure is used to receive shipping label information from the Kewill system. This information will be inserted in database tables within RDM.

package_weighed

This stored procedure is used to receive outbound carton tracking numbers once Kewill receives that actual weight of these containers. The tracking number will be updated on the outbound carton record in RDM.

ship_lane_upload

This stored procedure is used to call the Kewill socket interface procedure, case_weighted, and notify Kewill of the actual container weight.

Tables

Cont_Ship_Label

RDM table to hold shipping label information and error message returned from Kewill. This will be a child table to the container table, (facility_id and container_id must exist in the container table).

Column Name	Column Type	Primary Key?	Req?	Valid Values	Description
facility_id	VARCHAR2(2)	Y	Y		Facility ID
container_id	VARCHAR2(20)	Y	Y		Unique container ID on shipping label.
binary_label_info	LONG	N	N		Carrier compliant shipping label for outbound carton
label_type	VARCHAR2(4)	N	N		Type of label – EX: ‘PNG’, ‘ZEBR’, ‘MONA’
label_size	NUMBER	N	N		Size in bytes of the binary information contained in binary_label_info
rejection_reason	VARCHAR2(50)	N	N		Error message returned from Kewill when Kewill is unable to create a shipping label for the container

Rapistan socket interface

RDM interfaces with Rapistan through socket interfaces. RDM still generates directives based on logical dest ID's associated to locations setup in RDM.

For RDM generated directives (message sent to the control system), a trigger will call stored procedures used in the socket interface for various message types. Different message types are generated depending on where the container is going in the facility due to data required by the control system. RDM will determine the message type and call the appropriate procedure.

For control system confirmations (message received from the control system), divert confirmations will be sent to RDM via stored procedures. Similar to the directive procedures, the upload confirmation procedures will be created based on the message type sent from the control system.

Triggers

create_sorter_instruction_trg

When the Rapistan interface is enabled in RDM, this trigger will call the socket interface package, on the message transfer from RDM to Rapistan.

appt_rec_dir_trig

When the Rapistan interface is enabled in RDM, this trigger will write receiving directive records for ASN appointments when the appointment status is updated to 'PEND'.

cont_dest_trg.sql

Send the dest ID (carrier service) to Rapistan if the dest ID changes for an outbound container.

Packages

process_diverts_a.sql

Select additional fields from the sorter_intake table, in addition to performing palletization logic.

receiving_upload.osp

Accept receiving location directive confirmations from the control system. It will insert records into the sorter_intake table to be processed by the process_diverts_a.sql script.

divert_confirmation.osp

Accept divert directive confirmations from the control system. It will insert records into the sorter_intake table to be processed by the process_diverts_a.sql script.

ship_lane_upload.osp

Accept shipping location directive confirmations from the control system. It will insert records into the sorter_intake table to be processed by the process_diverts_a.sql script.

pack_wave_release_upload.osp

Receive pack wave release confirmations from the control system. It will call the new unit_sorter_directive.osp stored procedure to send unit sortation information to the control system for the pack wave that was released by the control system.

unit_control_sorter_upload.osp

Receive unit sorter confirmations from the control system. It will update the container_item table for the outbound carton being sorted.

combine_wip_codes.osp

WIP processing associated with outbound cartons.

receive_container2.osp

Write receiving_directive records upon receipt for non-ASN, specified case pack PO receiving.

Tables

Sorter_Intake

This table is used for all container transactions from the control system to RDM, including inbound, outbound, and movements within the facility.

Field Name	Field Type	Primary Key?	Req?	Description
facility_id	X (2)	N	N	Facility identifier
sorter_seq	N (9)	Y	Y	Sorting Sequence
container_id	X (20)	Y	Y	RDM container identifier
logical_dest_id	X (4)	N	Y	Logical destination ID that relates to a location within RDM.
divert_type	X (1)	N		
divert_ts	DATETIME	N	N	Date/timestamp
tracking_id	X (25)	N	N	Current field.
pallet_id	N (6)	N	N	Rapistan pallet identifier.
expected_cont_qty	N (6)	N	N	Number of cases on pallet ID.

Field Name	Field Type	Primary Key?	Req?	Description
scale_weight	N (4) v N (3)	N	N	Scale weight
Length	N (4) v N (2)	N	N	Measured length
Width	N (4) v N (2)	N	N	Measured width
Height	N (4) v N (2)	N	N	Measured height
Packer_id	X (10)	N	N	Populated for shipping cartons for audit purposes.

Third party routing interface

The third party routing interface determines the order in which outbound containers are picked and loaded onto trailers. The estimated cube and weight to be shipped for a given set of stores for a specified ship date is loaded into a file in RDM for the 3rd party routing system to process. The routing system then defines the routes used for that date and the order in which each store's stock will be loaded onto trailers shipped that day. This information is then used in RDM to determine the order in which outbound containers will be picked so that they can be loaded in the proper sequence.

Packages

ship_cube_inquiry.pkg

Procedures and functions used to select stock orders based on user-defined criteria, calculate estimated weight and cube by ship destination, populate the Ship Cube Inquiry screen with the results, and generate a route data file used as input for the 3rd party routing system.

route_data_upload.pkg

Procedures used to read and process route information returned from the 3rd party routing system and update stock orders with the carrier service route and ship date provided.

de_sort_picks.osp

Process used in distribution to order picks on a wave based on the carrier service route and ship date assigned to a stock order by the 3rd party routing system.

route_data_upload.sh and route_data_upload.sql

Batch process used to read data files provided by the 3rd party routing system to create routes and route sequences and then assign a route and ship date to designated stock orders.

Download transactions

Retek Distribution Management sends a file containing estimated weights and cubes for the ship destinations by stock order number for which picking and shipping will occur on a given ship date. This file is created in the Ship Cube Inquiry screen, and placed in the directory specified in the “DOWNLOAD_DIR” environmental variable.

Route data files created have the following naming convention:

route_data_facility ID_YYYYMMDDHH24MISS.dat

where *facility ID* is the 2-character facility identifier and *YYYYMMDDHH24MISS* is a date time stamp from the point of creation. Multiple route data files may be created for a single ship date, though this is not a recommended practice.

The route data file has the format:

Field Name	Field Type	Primary Key?	Req?	Description
Dest_id	X (10)	Y	Y	Ship destination identifier
Distro_nbr	X (10)	Y	Y	Stock order identifier
Total_cube	N(10)V(2)	N	N	Total cube for dest id in the stock order
Total_weight	N(9)V(3)	N	N	Total weight for dest id in the stock order.
Ship_date	YYYYMMDD	Y	Y	Date stock is to be picked and shipped.
Order_cube_UDA1	X(10)	N	N	User defined attribute
Order_cube_UDA2	X(10)	N	N	User defined attribute
Order_cube_UDA3	X(10)	N	N	User defined attribute
Order_cube_UDA4	X(10)	N	N	User defined attribute

Upload transactions

The files created by the 3rd party routing package are placed in a directory named by the UNIX environment variable “UPLOAD_DIR”. The files may be named with any set of numbers or characters as a prefix, but must end with the following character strings:

Distro Route Upload	...distro_route.dat
Carrier Service Route Upload	...carrier_service_route.dat
Route Date Upload	...route_date.dat
Route Dest Upload	...route_dest.dat

Multiple files named with different prefixes may exist for a single data type (ie. 001_route_dest.dat, 002_route_dest.dat). RDM processes spool data from the routing files into corresponding upload tables, then each file processed is renamed with an extension of .bak nnn , where nnn is a UNIX session ID.

There are no interdependencies across the four routing upload files. Any or all of them may exist in the upload directory and be processed at the same time. **Distro route upload**

The distro route data file contains the ship date, carrier, service, and route codes to be assigned to a given stock order. The carrier service route must be a valid entry in the carrier_service_route table in RDM.

Field Name	Field Type	Primary Key?	Req?	Description
Facility_id	X(2)	Y	Y	Facility identifier
Transaction_ts	YYYYMMDDHH24MI	N	N	Date timestamp
Distro_nbr	X (10)	Y	Y	Stock order identifier
Carrier_code	X(4)	Y	Y	Carrier identifier
Service_code	X(6)	Y	Y	Service identifier
Route	X(10)	Y	Y	Route identifier
Ship_date	YYYYMMDDHH24MI	Y	Y	Date stock is to be picked and shipped.
Distro_route_UDA1	X(10)	N	N	User defined attribute
Distro_route_UDA2	X(10)	N	N	User defined attribute
Distro_route_UDA3	X(10)	N	N	User defined attribute
Distro_route_UDA4	X(10)	N	N	User defined attribute

Carrier service route upload

The carrier service route data file contains carrier service route combinations to be stored in RDM's carrier_service_route table. Carrier code and route must be valid entries in the carrier and route tables in RDM respectively.

Field Name	Field Type	Primary Key?	Req?	Description
Facility_id	X(2)	Y	Y	Facility identifier
Transaction_ts	YYYYMMDD DHH24MI	N	N	Date timestamp
Carrier_code	X(4)	Y	Y	Carrier identifier
Service_code	X(6)	Y	Y	Service identifier
Route	X(10)	Y	Y	Route identifier
Location_id	X(12)	N	N	Optional location id where containers for this carrier service route will be staged for loading

Route date upload

The route date upload file contains routes and the order in which they are to be picked and loaded for a given ship date. If a route coming from the 3rd party routing system is not already defined in RDM's route table, an entry in this table will be created for it.

Field Name	Field Type	Primary Key?	Req?	Description
Facility_id	X(2)	Y	Y	Facility identifier
Transaction_ts	YYYYMMDD DHH24MI	N	N	Date timestamp
Route	X(10)	Y	Y	Route identifier
Ship_date	YYYYMMDD DHH24MI	Y	Y	Shipping date for which route sequence applies.
Route_sequence	N(3)	N	N	Sequence in which the route will be loaded with other routes for the same day.

Route dest upload

The route dest upload file contains all of the ship dests for a given route and the order in which they are to be loaded onto a trailer. The route and dest ID values must be valid in RDM's route and ship_dest tables respectively.

Ship date is a required entry for each record in the route dest upload file. A default value of 01-Jan-1900 (190001011200) may be used for static route dest sequences that do not change from day to day. Ship dest sequences loaded for any other date will be valid only for that ship date.

Field Name	Field Type	Primary Key?	Req?	Description
Facility_id	X(2)	Y	Y	Facility identifier
Transaction_ts	YYYYMMDDHH24MI	N	N	Date timestamp
Route	X(10)	Y	Y	Route identifier
Dest_id	X (10)	Y	Y	Ship destination identifier
Ship_date	YYYYMMDDHH24MI	Y	Y	Shipping date for which load sequence applies.
Load_sequence	N(3)	N	N	Sequence in which the containers for a given dest id will be loaded with other ship dests in the same route.