Oracle® Retail Warehouse Management System

Operations Guide

Release 13.0.1

June 2008

ORACLE®

Oracle® Retail Warehouse Management System Operations Guide, Release 13.0.1

Primary Author: Archana Kishore

Value-Added Reseller (VAR) Language

(i) the software component known as **ACUMATE** developed and licensed by Lucent Technologies Inc. of Murray Hill, New Jersey, to Oracle and imbedded in the Oracle Retail Predictive Application Server - Enterprise Engine, Oracle Retail Category Management, Oracle Retail Item Planning, Oracle Retail Merchandise Financial Planning, Oracle Retail Advanced Inventory Planning and Oracle Retail Demand Forecasting applications.

(ii) the **MicroStrategy** Components developed and licensed by MicroStrategy Services Corporation (MicroStrategy) of McLean, Virginia to Oracle and imbedded in the MicroStrategy for Oracle Retail Data Warehouse and MicroStrategy for Oracle Retail Planning & Optimization applications.

(iii) the **SeeBeyond** component developed and licensed by Sun MicroSystems, Inc. (Sun) of Santa Clara, California, to Oracle and imbedded in the Oracle Retail Integration Bus application.

(iv) the **Wavelink** component developed and licensed by Wavelink Corporation (Wavelink) of Kirkland, Washington, to Oracle and imbedded in Oracle Retail Store Inventory Management.

(v) the software component known as **Crystal Enterprise Professional and/or Crystal Reports Professional** licensed by Business Objects Software Limited ("Business Objects") and imbedded in Oracle Retail Store Inventory Management.

(vi) the software component known as **Access Via™** licensed by Access Via of Seattle, Washington, and imbedded in Oracle Retail Signs and Oracle Retail Labels and Tags.

(vii) the software component known as **Adobe Flex™** licensed by Adobe Systems Incorporated of San Jose, California, and imbedded in Oracle Retail Promotion Planning & Optimization application.

(viii) the software component known as **Style Report™** developed and licensed by InetSoft Technology Corp. of Piscataway, New Jersey, to Oracle and imbedded in the Oracle Retail Value Chain Collaboration application.

(ix) the software component known as **WebLogic™** developed and licensed by BEA Systems, Inc. of San Jose, California, to Oracle and imbedded in the Oracle Retail Value Chain Collaboration application.

(x) the software component known as **DataBeacon™** developed and licensed by Cognos Incorporated of Ottawa, Ontario, Canada, to Oracle and imbedded in the Oracle Retail Value Chain Collaboration application.

Contents

3 System Administration

4 DBA Administration Module

5 RWMS RIB Components

6 Subsystem Interfaces

Preface

This operations guide serves as Oracle Retail Warehouse Management System (RWMS) solution reference to explain 'backend' process.

Audience

Anyone with an interest in developing a deeper understanding of the underlying processes and architecture supporting RWMS functionality will find valuable information in this guide. There are three audiences in general for whom this guide is written:

- Business analysts looking for information about processes and interfaces to validate the support for business scenarios within RWMS and other systems across the enterprise.
- System analysts and system operations personnel who:
  - Seek information about the RWMS processes internally or in relation to the systems across the enterprise.
  - Operate RWMS regularly.
- Integrators and implementation staff with overall responsibility for implementing RWMS.

Related Documents

For more information on RWMS Release 13.0.1, see the following documents:

- *Oracle Retail Warehouse Management System Release Notes*
- *Oracle Retail Warehouse Management System Installation Guide*
- *Oracle Retail Warehouse Management System Online Help*
- *Oracle Retail Warehouse Management System Data Model*
- *Oracle Retail Warehouse Management System Radio Frequency User's Guide*
- *Oracle Retail Warehouse Management System User Interface User's Guide*

Customer Support

- https://metalink.oracle.com

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

Review Patch Documentation

For a base release (".0" release, such as 13.0), Oracle Retail strongly recommends that you read all patch documentation before you begin installation procedures. Patch documentation can contain critical information related to the base release, based on new information and code changes that have been made since the base release.

Oracle Retail Documentation on the Oracle Technology Network

In addition to being packaged with each product release (on the base or patch level), all Oracle Retail documentation is available on the following Web site:

`http://www.oracle.com/technology/documentation/oracle_retail.html`

Documentation should be available on this Web site within a month after a product release. Note that documentation is always available with the packaged code on the release date.

Conventions

The following text conventions are used in this document:

Convention	Meaning
boldface	Boldface type indicates graphical user interface elements associated with an action, or terms defined in text or the glossary.
italic	Italic type indicates book titles, emphasis, or placeholder variables for which you supply particular values.
`monospace`	Monospace type indicates commands within a paragraph, URLs, code in examples, text that appears on the screen, or text that you enter.

1

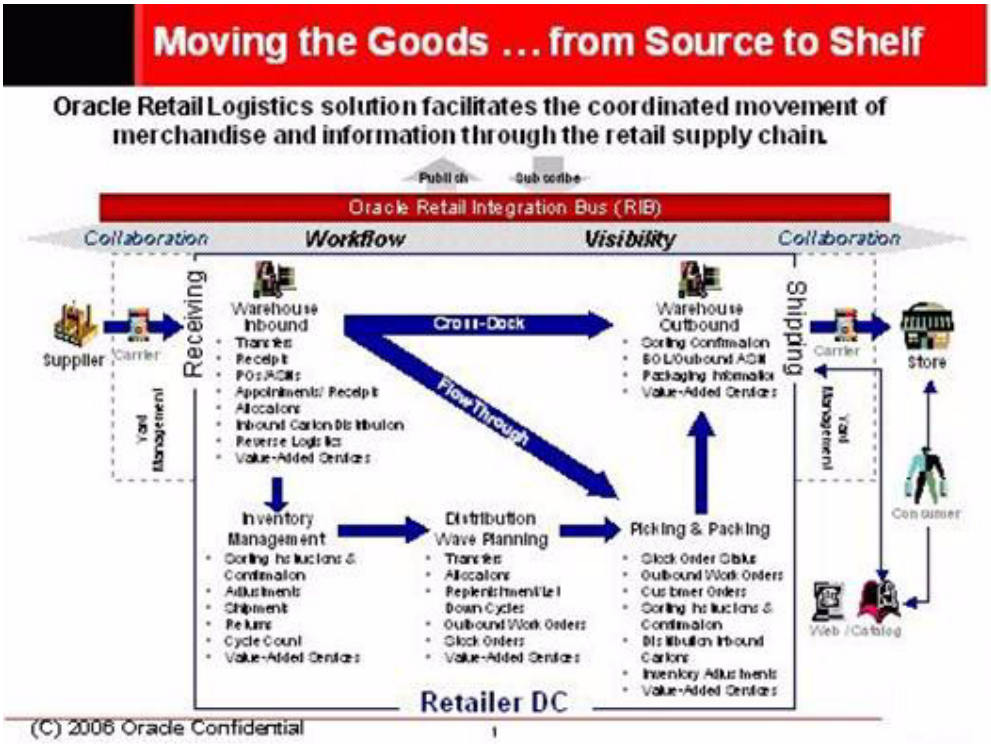
Overview

RWMS is a warehouse management system that provides all of the tools necessary to control a modern Distribution Center (DC). RWMS optimizes the flow of merchandise and resources at the DC, fulfillment center, or warehouse. RWMS functionality enables fulfillment across multiple channels. The Web architecture extends functionality and visibility to remote facilities and trading partners via the Internet.

Decision support tools help plan using facility resources efficiently and monitoring existing activities and merchandise flow. Radio frequency (RF) terminals make real-time inventory control and task management possible.

The table-driven functionality supports a variety of business formats and industries, enabling the system to change as the business changes. The following diagram shows the RWMS flow.

Figure 1–1 RWMS Functional Flow



Features

RWMS delivers a functional match to distribution management requirements, including:

- Trailer/yard management
- Inbound freight scheduling
- Carrier compliance tracking
- Receiving and pre-distribution with inbound processing
- Cross-docking, flow-through, and stock
- Full ASN functionality
- Web-based vendor ASN entry
- Quality assurance (QA) with vendor compliance tracking
- Inventory control
- Inventory management
- Item maintenance, management, and profiling
- Customer returns
- Return to vendor (RTV)
- Value-added services
- Distribution and resource planning
- Radio frequency picking and packing
- Paper picking
- Task management
- Shipping and EDI manifesting

Supported Handhelds

The handheld devices run on Windows CE 5.0 with the following screen sizes:

- Hand held: 240w x 320h pixels
- Wrist mount: 320w x 240h pixels
- Truck mount - full screen: 800w x 600h pixels
- Truck mount - half screen: 800w x 320h pixels

2

Solution and Business Process Overview

This chapter includes the following sections:

Inbound Freight Scheduling

The Inbound Freight Scheduling module schedules all incoming merchandise in preparation for receipt. The pre-receiving function optimizes warehouse resources and promotes smoothly executed operations during shipment arrivals for pre- and post-allocated, domestic, and international merchandise.

The Inbound Freight Scheduling module provides the following capabilities:

- Increases productivity by scheduling inbound freight.
- Manages both reserve and cross-dock or flow-through merchandise.
- Saves time by preprinting receiving labels for incoming merchandise.
- Provides appointment schedule and compliance reports.
- Ensures order accuracy.
- Provides purchase order (PO) and ASN inquiries.
- Assists in work flow management.
- Equips the user with door schedule, including utilization graph.

Appointments are made within an established timeframe (receiving window). The receiving window is based on the early- and late-received date as established by the buyer from the purchase order and/or ASN. ASN, PO, and allocation inquiries are available through RWMS.

Advance shipment notices may come directly from the carrier or supplier into RWMS, or they may come through the merchandising system, such as Oracle Retail Merchandising System (RMS), or through an import management system, such as Oracle Retail Trade Management (RTM). RTM is an import and customs tracking product that supports the letter of credit, PO tracking, customs tracking, across border requirements, landed costs, and transportation analysis up to the arrival at the distribution center (DC).

RWMS assists operators in managing dock doors for inbound trailers or containers, assigns merchandise for cross-docking or flow-through distribution and provides the work-flow plan required for each trailer. Detailed shipments that are not assigned a door or delivery date is visible in an unscheduled appointed bucket, which allows a quick reaction of the arrival or schedule of the delivery.

RWMS uses workload planning and the current appointment schedule to determine a time, date, and door for receiving the scheduled appointment. The requirements are based on the merchandise mix, using labor management service standards.

If the scheduled quantity does not match the allocated quantity, RWMS reallocates or notifies the buyer of the change in advance of the receipt through the Oracle Retail Active Retail Intelligence (ARI) system, allowing time to reallocate the merchandise and optimize the distribution workflow.

A variety of reports are available, in report format and online, to support workflow planning, scheduled and unscheduled receipts, ASNS, trailer information at the yard and at the door, and so on.

Web ASN

The Web-enabled architecture of RWMS creates opportunities for collaboration between DC personnel and suppliers. This is particularly true for Advance Shipment Notices (ASN).

RWMS allows the user to enter ASN information directly into the system with a standard Web browser and Internet connection. PO and allocation level information is validated to guarantee a connection. The Internet infrastructure allows DC personnel to extend the use of ASNs to suppliers previously unable to provide traditional EDI. Increased use of ASNs can improve the ability to streamline the processing and flow of goods through the facility.

Trailer/Yard Management

The Trailer/Yard Management module tracks all trailers at the DC or drop lot (if the trailer has an RWMS specific location ID) from check-in to checkout. Once a container is in the system, RWMS tracks the container no matter where it is.

The Trailer/Yard Management module provides the following capabilities:

- Tracks all trailers in the yard.
- Provides location, appointment number, and status of each trailer.
- Improves yard and fleet management.
- Allows trailers to be taken in and out of service.

- Supplies trailer and yard status reports.
- Provides location and trailer inquiries on RF terminals.
- Uses yard, trailers, and dock doors more efficiently.
- Provides content visibility that lets trailers to be used as temporary storage locations.

RWMS supplies the complete status information of each trailer, resulting in increased efficiencies in the yard and in dock door utilization. Yard and trailer status updates for inquiry at the system level, as well as through the RF terminal, defining each trailer's current activity and location. Real-time status changes update in the system as the associates perform a variety of operations and trailer movements.

The Trailer/Yard Management module acts as a tracking monitor for the trailer fleet of the DC. It keeps track of each trailer by means of the trailer identification. This identifier provides the user with the current activity of trailers in the yard.

Receiving

Through the Receiving module, RWMS logs in and receives merchandise, conducts and tracks vendor compliance sampling, resolves shipment issues, and manages the pre-allocation of any merchandise.

The Receiving module provides the following capabilities:

- Offers receiving workload planning tool.
- Manages both reserve and flow-through merchandise.
- Provides real-time status of current receiving processing.
- Allows adjustment to the receipt quantity after the receipt is closed.
- Offers user-defined trouble conditions for tracking vendor compliance issues.
- Directs merchandise to user-defined stations for troubleshooting.
- Allows receiving of both appointed and unscheduled merchandise.

Associates receive merchandise from vendors, with the aid of the RF or system terminal, at the receiving door. RWMS manages the merchandise that is destined for reserve as inventory or for processing and shipping as pre-allocated flow-through merchandise.

RWMS provides the flexibility to handle for pre-and post-allocated merchandise in one DC. User-defined services are associated with and sequenced for merchandise as a standard feature, reducing turnaround time through the DC.

Receiving Workload Planning summarizes the daily appointments assigned to a door and provides an estimated unload and putaway time to help the receiving supervisor utilize resources.

RWMS offers a receiving solution for supporting pre-labeled merchandise. A label option solution is also available, allowing the application of labels to merchandise at the time of receipt. These features are driven by the appointment and pre-distribution algorithm just prior to the arrival of the merchandise, accommodating last-minute changes in the distribution.

Vendor compliance issues, or troubleshooting issues, are managed online without interruption through user-definable parameters in the receiving process. Issues are marked at the appointment or container level, and directed to a value-added or holding area for resolution or putaway in storage to be called out later. As part of the

system configuration, issue codes may be associated to a value-added service and may be used to reconcile issues. For example, an issue code of Incorrect Retail may be paired with a value-added service of Re-Ticketing, which would direct the item to the ticketing area.

RWMS receives an authorized adjustment to the receipt quantity for issue resolution or other adjustments after the receipt is closed. As adjustments occur, updates to status and quantities are written to the Oracle Retail Integration Bus (RIB) for updates to merchandising or other inventory systems.

The receiving process starts when the trailer arrives at the receiving door. While a trailer is unloaded, the system keeps track of all the expected cases in the shipment by ASN, or purchase order and item identification.

Radio Frequency Environment Receiving

The RF receiving option allows users to open appointments and receive merchandise, scan merchandise using bar code labels and RF terminals, perform quality control, mark merchandise or appointments for issues, close appointments, release trailers, build containers, and track lot identification for dated merchandise.

The RF receiving environment supports both variable weight (catch weight) and perishable date merchandise.

Additionally, labeled containers are received via a conveyor or sortation interface, reducing the need for manual receiving with RF devices.

Radio Frequency Environment Palletization

The palletization option allows the user to build containers of merchandise, consolidate mixed products, and resolve merchandise issues.

Non-Specified Case Pack and Blind Receiving of Variable Case Packs

The Non-Specified Case Pack and Blind Receiving processes allow the validation of POs and other items. The case pack is specified at the time of receipt rather than restricted to the expected case pack entered when scheduling an appointment. Non-specified case pack receiving also handles bulk receipts.

Blind receiving does not require an appointment (for example, UPS shipments), however, to use blind receiving, DC personnel must forego the door management and inbound workload planning associated with the appointment scheduling process in RWMS. Therefore, RWMS supports a receipt method allowing variable case pack receiving in conjunction with a scheduled appointment.

Inbound Work Orders

After the completion of the receiving process, RWMS directs items with Inbound Work Orders to the appropriate processing area(s). An Inbound Work Order is a request for a VAS to be applied to a receipt, and is applied to the SKU level. For example, if the buyer knows that a particular item needs to be ticketed, the buyer can apply an Inbound Work Order of Ticketing when created the PO. This information is transmitted to RWMS and is processed through the VAS/WIP (Work In Progress) functionality. RWMS can apply WIPs at the time of receipt based on Item Attributes, which the client may associate with appropriate Value Added Services. For example, all of item X needs to go through a date capture process, based on the attribute of item X, it is directed to the date capture WIP area for processing. RWMS allows WIPs to be

automatically applied for "first time" items without the intervention of the host system.

Quality Control

Quality Control (QC) is handled through the VAS functionality, allowing QC processing requests and directives to be made through Inbound Work Orders, as part of an issue code, or through any of the other application avenues available through the standard VAS functionality. QC WIPs may be applied based upon a Vendor Inspection % and Vendor Inspection Frequency. For example if the vendor inspection percentage was two percent and the frequency was three, on every third receipt from that vendor two percent of every line item on the receipt is inspected.

Distribution and Resource Planning

The Distribution and Resource Planning module provides the following capabilities:

- Prioritizes and manages operator workflow.
- Supports pre- and post-allocations, and post-allocation by PO.
- Makes numerous user-definable distribution and resource planning models available.
- Provides a decision tool that allows the user to test scenarios.
- Supports space and time conservation pick planning model.
- Prioritizes emergency orders automatically.
- Handles directed replenishments based on user-defined roles.

The distribution module allows wave planning, set up a packing schedule, view pending picks, inquire about stock order allocations, create stock orders, and print reports. The distribution process starts through an outbound request for merchandise referred to as a stock order or allocation. In RWMS there are three types of allocations that can request merchandise:

- Pre-allocation
- Post-allocation
- Post-allocation by PO

The distribution module constructs user-defined queries during wave planning for complete flexibility on how the operation is run. The waves create picking operations in the facility using user-defined rules at the facility, wave, and item level.

Replenishment

There are a number of instances which RWMS creates user-directed tasks for the distribution of merchandise to a pick location. This is performed when:

- Merchandise falls below a user-defined quantity in a picking location.
- The demand of an item for a wave brings the quantity below a user-defined quantity.
- The demand of an item for a wave exceeds a user-defined threshold, thus directing merchandise to a high volume pick module.
- An item break case pick method is an alternate picking subsystem.

A supervisor initiates replenishments in order to top off a location. The top-off process attempts to fill the forward picking location (FPL) to capacity with merchandise from reserve.

When the number of units of an item in the DC drops below the pre-assigned replenishment level, a replenishment pick is automatically initiated. Replenishments are done on the bulk or container level. The replenishment tasks are created in RWMS based on the methods above and completed through an RF process that is initiated with the following information:

- **Reserve zone** - The zone the user picks from.
- **Unit pick zone** - The zone the user picks to.
- **Item ID** - The specific item to replenish.
- **Wave** - The wave that created the replenishment.

Users can control the release of the replenishments with a series of controls intended to prevent short picks or over-full locations. Priorities of remaining replenishment picks can be updated automatically, so the most important replenishments are presented first.

Distribution Methods

DC personnel have multiple ways to select merchandise to fulfill order/wave requirements. The item profile or wave drives the distribution method used when evaluating eligible inventory in the DC. The evaluation criteria are listed below:

- **Efficiency** - The least number of picks to accomplish the task. RWMS attempts to capture the most number of units per stop during this method. The evaluation process searches for pallet picks, then case picks, and then unit picks to fulfill the order/allocation.
- **First In First Out (FIFO)** - Picks are generated based on selection of merchandise with the earliest receipt time stamp.
- **Pick to Clean** - Picks are generated with the objective of cleaning out as many locations as possible.
- **First Expiration First Out (FEFO) or Best Before Date** - Grocery retail requires consideration of best-before dates of select merchandise to fulfill orders. Best-before dates are captured for perishable merchandise on receipt. A perishable indicator is added to the item table.

For each perishable item, a safety period (in days) is maintained that specifies the minimum number of days prior to the best-before date. A formula date (or pick-not-after date) is then calculated. The pick-not-after date is the latest date the merchandise is picked to ensure stocking in the store sufficiently prior to the best-before date. Perishable merchandise is distributed with pick-not-after date as the primary selection criterion.

Wave Preview Based on Selected Orders

The current RWMS wave review is based on actual picks generated during the distribution process. This means that the orders were selected and the distribution ran. While this approach provides the most accurate picture of the wave workload, it requires processing time for the actual creation of the picks.

From a planning perspective, there is a benefit to a quicker estimate of the wave workload based on the selected orders. Using this tool, wave planners anticipate the

size of the wave and make adjustments in the order planner's ability to properly size the wave in advance of the distribution process, while retaining the ability to fine tune the wave size and resource requirements after distribution has run, if necessary.

Picking

During the distribution process, the system assigns master container IDs to group picks. For example, it allows or disallows the mixture of destinations on an individual pallet. However, zone, pick type (bulk, case, or unit pick), and pick volume always divide the picking work assignments into workable units. In the case of a location shortage, RWMS also provides the operator with the opportunity to skip the pick and to return at the end of their picking assignment, or to perform a Hot Pick in which RWMS searches the facility for available product in other locations. RWMS also supports container swapping with respect to container and bulk picking processes. This provides the ability to select which pallet/case to pick from a location when multiple pallet/cases exist with the same attributes.

RWMS processes the stock allocations and available inventory to create four types of picks:

- **Bulk picks** - The user can pick an entire pallet of a single item for shipment to a single ship destination or unit pick system, based on a wave. Bulk picks are used for either conveyable or non-conveyable merchandise.
- **Container (or case) picks from reserve** - The user can pick one or more full cases from a bulk container in a reserve location for shipment or to a unit pick system, based on a wave. The user can also use break label packets, which allows the user to subdivide large picking packages, thus keeping locations from being visited more than once to fulfill an order.
- **Container (or case) picks from forward case locations** - Allows for case pick to belt or case pick to pallet of unlabeled cases from a forward location. Locations are SKU/item specific and are replenished in a variety of methods. Forward case picks to pallet are picked via RF, with or without a pre-printed label packet. Forward case picks to belt are picked without RF with a label packet. Picks are confirmed via user confirm pick screen or through a conveyor interface.
- **Unit picks** - The user pulls individual units out of FPL. Pick quantities are adjusted downward and problem locations are marked.
- **Replenishment picks** - The user pulls containers from the stock/reserve inventory to the unit/case picking locations. These containers are opened and used to restock the unit pick bins. Case locations are replenished by cases or full pallets.

If a FPL is full after a wave is completely picked, space constraints can prevent the overfilling of locations. Dynamic overflow provides the ability to define locations with a hard limit on capacity, and when the capacity limit is exceeded, it provides an overflow area for items directed to and picked. This provides operators in an environment with constrained locations the ability to manage FPLs during unanticipated volume peaks.

Radio Frequency to Store Multi-Store Pick Confirmation

RWMS supports RF-based Put to Store picking, in which SKUs move past stationary store cartons. Currently, puts are presented to and confirmed by the DC personnel one store at a time to guarantee accuracy.

RWMS provides an alternate means of presenting and confirming put to store pick directives. A picker is responsible for a range of stores. Replenishment containers are passed to the picker.

Upon scanning the license plate of the replenishment container, the picker receives a list of all pick requirements within their range of stores that is satisfied by the contents of the replenishment container. If no exceptions are encountered, the picker confirms all of the displayed picks at the same time, as opposed to individually confirming each store's notes.

Put to store is performed at the unit or the case level. Single containers are introduced into put to store for broken case fulfillment. Similarly, pallets are introduced for full case processing.

Volumetric-Based Determination of Outbound Container Requirements

Distribution processing includes an option to pre-determine outbound container requirements based on volumetric data (cube for flat, units per standard length for hanging, and so on), and weight. Volumetric data and weight capacities are maintained for each container type.

- Ship alone
- Rigid dimensions
- Combinability
- Preferred (smallest) carton type
- Total cube
- Total weight (items, dunnage, collateral, carton)

If picking with a pick package, the package is generated to correspond to the pre-determined container requirements.

If picking without a pick package, the picking process indicates to the user when the pick-to container is full. At the start of the picking process, the picking screen prompts the picker for the container type being used. The system displays the default container type for the pick zone, but the picker may override this value.

During the picking process, the system tracks the fill level of the container. The picker is notified when the pick to container has reached capacity. At that point, the picker closes out the current container, drops it off if necessary, and opens a new container. Conversely, the system is configured to allow users to determine when a container is full.

Based on item types, RWMS also offers combinability codes, which prevents non-combinable items (for example, chemicals, foods) from being packed into the same outbound container.

Inventory Control

The Inventory Control module manages merchandise in the DC. It provides system and RF screens for general inventory functions, directed putaway, and movement of inventory, inventory adjustments, returns processing, and cycle counting.

The Inventory Control module does the following:

- Maintains inventory tracking and queries.
- Supports system selected and exception-based cycle counting.

- Provides task interleaving of functions and cycle count.
- Defines locations and associated attributes throughout the warehouse.
- Allows user defined inventory adjustment codes.
- Provides daily audits of warehouse statistics.
- Supports the ability to store multiple SKUs in the same storage or prime picking location.
- Directs putaway based on parameters driven through item and location configuration, demand of merchandise, and consolidated opportunities.
- Supports multiple case pack quantities of the same SKU.

Directed Movement

When merchandise is putaway, RWMS uses the product location attributes and configurable putaway plan to suggest the location that best conserves space in the reserve locations, or it inquires opportunities such as zone, location type, and status (for example, Different SKU, Same SKU, Empty Loc). The number of storage preferences associated with a particular plan is virtually unlimited, providing significant control over the storage strategy for the overall facility. Inventory Control uses three putaway methods: cube, capacity, and unit.

- **Cube** - Dimension derived from the length, width, and height
- **Capacity** - Capacity measured in units
- **Unit** - Capacity in number of units

All merchandise in the DC is system-directed whether it is for putaway, processing, or picking goods.

Concentric Putaway allows for putaway plan creation that suggests reserve storage locations closest to an item picking location. This is done using XYZ coordinates to measure the distance of storage locations from the pick face. This functionality is controlled for each item within a putaway plan.

Tracking Information

Visibility to critical information is maintained throughout the lifecycle of the products in RWMS. Data is captured and maintained at the lowest level of bar coding that an operation supports. The container ID tracks information associated with the product including receipt number, receipt date and time, purchase order number, lot number, expiration date, putaway date, last user to move the product, current status, distribution number, and physical dimensions and weight. Within a single location, RWMS can support multiple products and multiple case packs of the same product to allow for flexibility.

Cycle Counting

System cycle counting allows automatically scheduled cycle counts based upon user-defined parameters, such as the number of times per year a location should be counted, and the number of times per year an item should be counted. Exception cycle counting is activated upon the system discovering discrepancies during the course of picking and processing.

Pack and Hold Store Reassignment

DC personnel use a pack and hold philosophy in preparing goods for shipment to a new store or for major store expansions. After picking and processing, the merchandise is staged and waits for shipment to the new or expanded store.

At times a store fails to meet the opening date or expansion date. In such instances, DC personnel undo the picks and free up the merchandise for reassignment. RWMS enables the DC to reassign the merchandise to another store or return the merchandise to stock for reallocation.

Container Consolidation/Deconsolidation

RWMS enables picked merchandise to move, in part or in whole, from one container to another for the same destination via hand-held RF terminals. This feature facilitates consolidation of multiple containers of merchandise to maximize container fill, thereby minimizing transportation costs. This functionality also converts flat-pack merchandise to hanging, so that it may be delivered floor-ready to the store.

Order Status

RWMS maintains a specific status for each order. The status is available upon inquiry by buyers, DC personnel, and customers. This information is also published to the RIB for other systems to utilize. An order can be at various states of processing, assumptions must be made on the overall order status when multiple statuses actually apply.

Valid order statuses include:

- Open unprocessed.
- Selected for inclusion in a wave for distribution.
- Pending picks generated, but not yet confirmed.
- Picked picks completed.
- Loaded/manifested loaded onto trailer, awaiting shipment.
- Shipped.

Return to Vendor

RTV processing is fully initiated by the DC. Merchandise is marked for return and moved to a processing area to wait for return to vendor processing. The DC then performs the RTV function to remove the merchandise from the facility and post the transaction to the host.

Space Utilization

One of the primary operating constraints for many DCs is the availability of physical space. As such, it is critical to effectively use that space. Current RWMS location management is based on numbers of containers. This is expanded to account for volumetric considerations. For example, cube for flat, units per standard length for hanging.

Putaway logic supports the use of volumetric data when specifying a location for putaway. The specified putaway location is temporarily reserved to account for available space in the location to subsequent putaway attempts. Upon confirmation of the putaway, the user may either confirm to the specified location or override (based

on the item configuration settings) with a separate valid location. While the system does not direct movements that exceed location capacities in storage, the user is allowed to exceed the available space in a location on override.

Support Functions

The Support Functions module houses the main functionality of RWMS and serves as an aid in the supervision of the DC data. It helps top privilege level users to maintain specifications for every integral part of RWMS through separate editors. With the Support Functions module, the user views information about company vendors, designate print queues, item rules, location configuration, and generate user/activity productivity reports.

Item Distribution Center Characteristics Setup by Merchandise Hierarchy

Traditional host systems may not be able to maintain certain item characteristics specific to the DC processing, such as dimensions, weight, unit pick system, round-able, preferred zone.

RWMS enables the user to enter default values for categories of items based on various item hierarchy attributes, such as department, class, and subclass. The DC specifies default values for a specific combination of these criteria. This table is scanned upon download of new items to determine the desired default attributes for the item, reducing the item maintenance overhead for the DC.

Location Configuration

RWMS provides the system tools to setup a DC using locations, zones, location types, and facilities. Each level provides data that determines characteristics such as size and processing related to the locations.

Processes

Processes are defined for picking and distribution to allow flexibility to how a DC is configured for processing. The intersection of processes at the item level and the location level defines how an item is processed in the building. For example, items that are only received and processed in bulk quantities (full pallet) are configured with processes to only allow for putaway into certain areas and to only distribute at the full pallet level.

Item Class

RWMS allows for the creation of DC-specific Item Classes (which can be different from item master hierarchical classes) to allow for the maintenance of like items. These classes are used to apply DC-specific defaults, attributes, and processes to groups of items which control the processing in the facility.

Location Class

RWMS supports the creation of DC-specific Location Classes, which are used to apply like characteristics to large quantities of locations. This prevents locations from being managed one location at a time. Additionally, location classes are used to manage processes at the individual location level, which determines what type of DC function can occur at a given location.

Shipping

The Shipping module provides visibility and directs the loading and shipping of trailers. Both fluid and stages loading, for conveyable and non-conveyable items, are supported by the system. RF terminals verify the loading process and provide real-time data for manifesting electronically and with paper. All necessary shipping documents are created at the time of shipping to guarantee order accuracy.

The Shipping module provides the following capabilities:

- Supports fluid and staged loading processes.
- Supports multiple carrier, service, routes available for each customer.
- Supports loading customer cartons through multiple doors concurrently.
- Provides real-time shipping and loading status.
- Supports print and apply feature of UCC-128 compliant shipping labels.
- Guarantees order accuracy and loading accuracy.
- Creates EDI and paper manifest for customers.
- Allows trailers to close and reopen for optimal shipping and trailer management.
- Supports pack and hold merchandise.

For UCC-128 manufacturer labeled cartons, RWMS provides the operator the option of printing and applying an address-shipping level on a local label printer.

Merchandise arrives at the shipping doors by conveyor or by manual delivery. Shipping doors are opened by carrier, service, and route codes for all customers scheduled to ship merchandise by this code.

Multiple carrier, service, and route codes are assigned to any customer. Conveyable cartons divert to the proper shipping door by the sorting system defined by the carrier, service, and routine codes. The sorting system controller verifies the diversion of the carton, adding the carton to the current open manifest.

For non-conveyable merchandise, the product arrives by pallet, trolley, or cart and is scanned onto the trailer by using the master tracking ID and the trailer ID. Because of the unique parent/child relationship between the carton ID and the master ID, the carton level details are retained in the manifest.

RWMS allows the loading of multiple customers or destinations on the same trailer creating separate paper and electronic manifests for each one. Trailers may close and reopen to accommodate the management of shipments and trailers. Upon the sealing and the shipment of a trailer, final electronic and paper manifests are created for the DC and the carrier. The EDI 856 manifest created at the close of the trailer is passed to the customer providing carton level detail for every carton, item, and order shipped.

For consumer direct facilities, a key objective of fulfillment operations is to guarantee that the entire customer order is shipped together. This helps minimize shipping costs, and improves customer satisfaction by delivering the complete order. At times, portions of the order may not arrive together as planned at the packing operation. To help manage this situation, RWMS provides an Order Consolidation function to join the items together in the order. Regardless of which arrives first, the SKU or the order carton is staged until the corresponding order carton or SKU arrives, at which time the order is made whole and proceeds to shipping.

The shipping process can be configured to warn users of closing or shipping trailers when merchandise a store is yet to be loaded. Additionally, RWMS supports load

sequencing to match store delivery routes. Users are warned if they attempt to load stores out of delivery sequence.

Outbound Quality Control Audit

Picking errors create problems throughout the supply chain. One issue is store confidence in the integrity of shipments from the DC. As such, RWMS enables the user to verify the accuracy of the pickers prior to the shipment of merchandise. Each audit maintains a User ID, units audited, units over, and units short. The DC auditor staff determines the actual workload and number of units audited manually.

Task Management

In the GUI environment, the user can access the Task Management module. This functionality promotes the increase to labor efficiency by controlling the delegation of work to individuals in a real-time manner.

Each task eligible for assignment must be defined in the task queue. This task queue uniquely defines a specific task or group of tasks.

The following tasks are eligible for task management:

- Bulk picking
- Bulk replenishment picking
- Container picking
- Container replenishment picking
- Unit picking
- Move
- Putaway
- Cycle count
- Container loading

The RF Task Administration module provides the user with the capability to increase labor efficiency by controlling the delegation of work to individuals in a real-time manner. At the start of a RF session, the user enters the type of equipment used and a starting location. Tasks are assigned based on the rules defined by the DC.

Value-Added Services

The Value-Added Services module offers user-defined value-added service functionality in the distribution center. These value-added services, or WIP codes, allow the user to define physical locations and activities in the DC. Then the user offers a directed workflow while managing the task.

The Value-Added Services module does the following:

- Links merchandise issues to WIP codes for resolution and vendor compliance tracking.
- Supports pack and hold through WIP functionality.
- Tracks continuous, real-time work in progress status by container identification and user.
- Supports integrated, user-definable ticketing functionality.

- System-directed tasks.

In practice, these value-added services might include ticketing, kitting, repackaging, flat to hang, and others. Processes are associated with each container ID or license plate that is sequenced to allow an orderly process to the tasks. As operators are directed through the processes and move merchandise from one workstation to the next, the WIP codes are cleared, and the container status is updated and tracked.

The WIP processing operation changes the values of the start and end time stamps on each container's WIP list. Each processing operation sets the appropriate values, time stamps the task upon initiation, and updates the status and time stamp upon conclusion of the task. The QA process, a part of vendor compliance, is an example of an operation where the start and end times are set.

RWMS offers a multi-level bill of material functionality. Tracking is managed at both the component level as well as the kit or group level. Merchandise is assembled into kits or pre-packs as part of an inbound or flow-through process, or is assembled for outbound orders or allocations from inventory. For merchandise maintained in pre-packs in primary picking locations, components are automatically directed to pull and assemble into pre-packs for distribution via the WIP process. Merchandise is assembled or disassembled into product sets.

Return Processing

For returns, there are two primary objectives:

- Reconcile the account.
- Determine disposition of returned merchandise and process accordingly.

The first step in reconciling the account is to identify the appropriate customer and order information to acknowledge the return. To assist in this process, assign a return merchandise authorization (RMA) number to any shipment to a customer. Returns are processed with or without an RMA number.

Ideally, the customer includes the RMA with the returned merchandise. However, if the RMA is not included, RWMS provides query tools to assist in identifying the customer base on limited data, such as item, name, or postal code.

If available, the RMA number is used to access the list of items associated with the original shipment. The returned items are then selected from the list. If the returned item is not found on the list, then further investigation occurs to determine the appropriate RMA, or the item is entered manually.

The returned merchandise is inspected to determine its disposition and any subsequent processing or routine requirements. RWMS allows multiple SKUs in dynamic active locations. This is useful for returns processing putaway. RWMS tracks user-defined reason costs and action codes.

Audit Log

An audit log feature is assists with warehouse-related issues and inventory control. Audit logging is activated for **any** RWMS data entity (for example, POs, items, inventory), and is specified at the field or screen level, if desired. For example, the user audits all container touches (for example, track creation, movement, transfer, and deletion of containers, and modified attributes), as well as monitor selected field updates to the item master table. Inquiry screens and baseline reports are provided.

RWMS Interaction with Other Systems

With the release of RWMS 10, the application now utilizes RIB, the enterprise Application Integration platform that supports a "publish and subscribe" architecture. By utilizing a single repository, RIB - of interfaced information (such as shipped merchandise information published by RWMS) - allows any application that needs the information (such as merchandising, store systems, customer order management systems) to subscribe to the information without direct lines going to/from the application to every other application, thus eliminating redundant integration points. Legacy systems are integrated to RWMS via flat file transfer between the legacy system and the RIB. RWMS also provides standard interfaces to material-handling subsystems and third party picking systems (for example, pick-to-lights, tilt trays, bombay sorters).

RF integration is handled through RF emulation, allowing RWMS to perform on any RF vendor's hardware that supports terminal emulation. RF hand-held and truck-mount formatted screens provide real-time updates for operational activities.

RWMS is also closely integrated with the ARI application through an event indicator on the RWMS menu, which prompts the user to check the alerts by navigating directly into the ARI application.

3

System Administration

Each system user must have a unique logon ID to the operating system, Oracle, and RWMS. This chapter describes how you can create each of these accounts and contains the following sections:

Creating the rdmusr Account on the Operating System

When RWMS 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
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 the installation.

The value of TERM is limited to the following choices:

- ibm3151
- 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 the installation. The user may use the operating system's security provisions for password expiration.

RWMS enforces access control over the System screens regardless of the operating system security measures.

Creating Users in Oracle

Use the Oracle 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>
temporary tablespace TEMP;
grant wms_user to rdmusr;
default tablespace USERS
```

Creating User Accounts

Add the user in RWMS, using the User Table Editor screen. The password the user specifies in the system can be the same as or different than the user's operating system password.

Radio Frequency Operations

When Radio Frequency 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 RWMS 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 RWMS copyright. The login screen follows it. The user must fill in the username, password, and facility ID.

Operating System Functions

This section describes the print queues, cron jobs, daemon process, and system parameters.

Printer Queues

Create print queues in the operating system for reports and labels. RWMS 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:

- pick_audit_queue
- pick_label_queue
- pick_package_queue
- recv_audit_queue
- recv_label_queue
- recv_receipt_queue
- reprint_label_queue
- ship_bol_queue
- ship_label_queue
- 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. The user 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 *Oracle Retail Distribution Management User Guide*.

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

Table 3–1 Programs to remove obsolete data

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

Table 3–1 Programs to remove obsolete data

Name	Purpose	Parameters	Frequency
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
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 and 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 levels initiated from the GUI system.	None	Every 5 minutes

Table 3–1 Programs to remove obsolete data

Name	Purpose	Parameters	Frequency
run_dcplan.sh	Populates planned production activity data into the dc_plan_ data table, which is used for the Operational Overview screen.	facility_id	Once or twice a day at the beginning of the working day
run_distribution.sh	Matches inventory to allocation requests, creates pick directives and prints picking packages (if applicable).		Manually or every 15 minutes
run_kpidata.sh	Populates actual production data into the operational_data_ store table, which is used for the Operational Overview screen	facility_id	Every 15 minutes
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 had all labels scanned or nulled.	None	Hourly

Daemon Process

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.

Table 3–2 Daemon Process

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).

Table 3–3 File Management

Directory	Path	Purpose	Perm	Purging
Base Directory	$RDM	This is the base directory that other directories 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)

Table 3–3 File Management

Directory	Path	Purpose	Perm	Purging
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. The user can view and modify these in the System Parameter screen.

This table provides the name of each parameter, briefly explains its purpose and how it is used, and provides 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.

Table 3–4 File Management

Name	Purpose	Type	Allow User Edit
3rd_party_routing	Checked Y (yes) when using an FTP interface to a third 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

Table 3–4 File Management

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, RWMS 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
appt_asset_default	Sets the default value for receiving appointments for the capture of asset tracking…specifies how user is prompted to enter assets during receiving. Valid values are "Start", "End", or "During". Start: user prompted to enter assets at beginning of appointment. End: enter at end before closing appt. During: user is prompted for asset quantities after each item is received that is associated with asset tracking.	Text	Y
appt_purge_days	Number of days after closure to purge an appointment. Used in purge_ appointments_b.sh.	Number	Y
ari_enabled	Is ARI installed with RWMS?	Y/N	Y

Table 3–4 File Management

Name	Purpose	Type	Allow User Edit
asset_tracking	For transport asset tracking. When enabled, RWMS will support tracking of transport asset inventory and will enable the messaging from RWMS to communicate asset movements to the host. When Asset tracking is set to Y, RWMS tracks transport assets.	Y/N	Y
assortment_wip_code	WIP code applied when inbound container has an assortment item. Parent Item with child SKUs.	WIP	Y
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 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
blind_bulk_receiving	When set to "Y", the system generated PO receiving labels will not print any container quantities or unit quantities on bulk container labels. It will also not pre-populate the carton quantity on the RF receiving screen when receiving bulk pallets. When set to "N", the carton quantity will appear on the labels and RF.	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
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

Table 3–4 File Management

Name	Purpose	Type	Allow User Edit
consolidate_pend_wip	When set to Y, RWMS 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
def_bulk_resources	Sets the number of Bulk resources to use to display the Wave Duration on the wave planning screens.	Number	Y
def_catch_weight_uom	The default unit of measure for catch weights. This is used for upload transaction records to RMS, which requires a unit of measure.	Text	Y
def_cont_replen_res	Sets the number of case replenishment resources to use to display the Wave Duration on the wave planning screens.	Text	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

Table 3–4 File Management

Name	Purpose	Type	Allow User Edit
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
default_order_level	Type of saved query for order selection. FULL -Every line in query has to match. ORDER - If any line matches, RWMS shows all distribution lines. LINE - Only distributions that match display.	FULL, ORDER, and LINE	
default_order_type	Determines how downloaded stock orders 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
delete_pfl	When set to Y, and unit quantity in the 'from location' is 0, RWMS deletes the location record once the merchandise moves out. User override is provided on the FPL Move screen.	Y/N	Y

Table 3–4 File Management

Name	Purpose	Type	Allow User Edit
deposit_item_detail	For transport asset tracking. When deposit_item_detail is set to Yes, RWMS will itemize the deposit assets based on predefined relationships in all inventory messages (receipts, inventory adjustment, return to vendor, shipments, inventory balance). When set to No, RWMS will not communicate the deposit assets with the content item in inventory messages. When set to No it is assumed that the host system can calculate the deposit assets.	Y/N	Y
dflt_labeled_rcving	Defines the default labeled receiving (Y/N) value for appointments. Starting with version 12.0.1, the labeled_receiving global parameter is no longer used - this setting is now available at the appointment level.	Y/N	Y
display_item_id	Used in the multi-item UPC functionality. If set to Y, item information, which matches the UPC code, displays 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, RWMS allows allocation of merchandise from a pallet that has unfinished WIP codes associated with it.	Y/N	Y
distribute_partial	When set to Y, RWMS processes partial distribution of a dye lot. The maximum amount of a single dye lot is 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 used during cartonization if no defined container types hold items.	Container Type	Y
exceptions_stage	Area specified in building (location) where exception packages are sent for consolidation.	Location	Y
fcp_random_act_stg	Staging location for replenishments to random forward case locations.	Location	Y
fcp_random_dest_id	Destination ID for replenishments to random forward case locations.	Dest id	Y

Table 3–4 File Management

Name	Purpose	Type	Allow User Edit
first_ctn_seen	WIP code applied to first carton during receiving.	WIP	Y
first_time_sku	WIP code applied to first time SKU containers during receiving.	WIP	Y
fixed_replen_wave	When set to Y, RWMS groups all replenishment picks into Wave 1. When set to N, RWMS associates replenishment picks with the wave that originated the need.	Y/N	Y
FNC1_ASCII	This designates the end of string character that will be used to determine the last character of variable length fields in EAN 128 bar codes used during receiving.	Text	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
hot_replen_fulfill	Determines level at which hot replenishments will be directed toward forward pick locations. If set to "Y", it will always direct hot Replens to forward locations if the entire container will fit in the slot. If set to "N" it will only direct hot putaways if the forward location is below the replenishment level.	Y/N	Y
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

Table 3–4 File Management

Name	Purpose	Type	Allow User Edit
incl_xdock_appt_zone	Used during the appointment creation process to recommend the best fit door. When set to "Y", the system will check for the existence of PREDIST stock allocations and use the PTS induct zone for items with allocations. (Based on the unit pick system code assigned to the item.) If set to "N", cross-dock items will be ignored.	Y/N	Y
interface_tcp_flag	Indicates the use of a TCP/IP interface with a conveyor system. (Future RWMS use.)	Y/N	Y
interleaved_cc	When set to Y, RWMS 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, RWMS generates a picking label packet and a report. When set to N, RWMS assumes labeless picking and only generates a report.	Y/N	Y
labeled_receiving	When set to Y, RWMS generates a receiving label packet and a report. When set to N, RWMS assumes labeless receiving and only generates a report.	Y/N	Y
labeled_reserve	When set to Y, RWMS 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, labels for Unit picks prints even when labeled_picking = N.	Y/N	Y
load_sequencing	When set to Y, RWMS sorts picks with respect to the defined route/destination load sequence. When set to N, RWMS sorts according to distribution number sequence.	Y/N	Y
log_interface_error	Determines whether RWMS 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

Table 3–4 File Management

Name	Purpose	Type	Allow User Edit
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 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
metrics_purge_days	Specifies the number of days of data to retain in the distribution_metrics and unfilled_allocations tables. The purge batch scripts remove data older than the specified number of days.	Number	Y
min_auto_wave	The lowest wave number used by RWMS 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 contains more than one container item record.	WIP	Y
nbr_cartons_pallet	Max number of cartons per pallet, in putaway logic.	Number	Y
nbr_divert_w_ putaway	This is used with conveyor receiving where received containers are conveyed to a Putaway area. The number is the amount of containers diverted to each Putaway lane (represents a pallet). (Note: this was for the Walmart/Gap versions and is not effective for other versions.)	Number	Y
nbr_items_pallet	Max number of items per pallet, in putaway logic.	Number	Y
oflow_replen_dest_id	Destination ID for replenishments to Overflow forward picking locations.	Dest ID	Y

Table 3–4 File Management

Name	Purpose	Type	Allow User Edit
order_line_number	Y setting indicates that orders are being tracked at the order line level.	Y/N	Y
order_set_stage	Location in facility where outbound cartons are directed to have order sets printed.	Location	Y
order_status_upload	Y if order status information is uploaded to the host.	Y/N	Y
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 used for items that do not have a putaway plan specified.	Text	Y
pallet_tare_height	The average height of a pallet in terms of inches. This number is used in the calculation of rigid cube during the putaway process. Adds this value to the actual height to figure if it will fit. For adding pallet height.	Number	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

Table 3–4 File Management

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 causes 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, RWMS is picking by location and allows mixing of conveyable cartons of varying destinations onto a single pallet during Container Picking. When set to N, RWMS is picking by destination and does not allow 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, RWMS is picking by location and allows mixing of non-conveyable cartons of varying destinations onto a single pallet during Container Picking. When set to N, RWMS is picking by destination and does not allow 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 prints.	Queue	Y
pick_label_set	Determines the scheme used to generate the Container Identifier.	UCC128 or default	Y
pick_printer_type	Printer description.	Text	Y
pick_seq_setup	For full case picking, determines if picks will be displayed to the user in pick sequence order or location ID order. If Y, then the pick seq will be used; if N, then the location ID seq will be used.	Y/N	Y
pnad_isd_lead_time	Pick not after date/In store date lead time.	Number	Y

Table 3–4 File Management

Name	Purpose	Type	Allow User Edit
po_pack	For transport asset tracking. When set to Yes, RWMS does not expect to see the deposit components on the Purchase Order and will assume PO information will be at a master pack level. If set to No, RWMS will expect to see the deposit components on the Purchase Order. Communication of detail back to the host for the receipt will be determined by the deposit_item_detail SCP above	Y/N	Y
pod_break_group	Used for Print on Demand functionality. Determines if pickers can specify a quantity of bulk pallet labels that is not evenly divisible by the pallet group size. If set to "Y" and a pallet group is broken, the next print job will first print the remaining pallets from the broken group before printing new groups. If set to "N", only full group quantities can be printed.	Y/N	Y
pod_nbr_groups	Used for Print on Demand functionality. It defines the maximum number of pallet groups that the system will allow the user to request for printing.	Number	Y
populate_cont_weight	Determines whether or not calculated container weights are pre-populated to the RF screens.	Y/N	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_pick_up_loc	Location at which the system picks up cartons packed by PPS.	Location	Y
pps_round_up	When set to Y, the distribution process increases (round up) the distribution evenly across the destinations to consume the excess. When set to N, the process does 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 Destination 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, RWMS 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

Table 3–4 File Management

Name	Purpose	Type	Allow User Edit
pts_adj_trbl_code	Trouble code applied to cartons with PTS adjustments when the user did not have the required security level to approve adjustments in case PTS	Text	Y
pts_ctn_max_days	Number of days before open Put To Store carton is flagged.	Number	Y
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 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
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

Table 3–4 File Management

Name	Purpose	Type	Allow User Edit
rec_cases_per_hour	Used for the new Operational Overview screen added in V12.0.1. This value is used to calculate the expected unloading time for appointment lines received at the CARTON level.	Number	Y
rec_pallet_per_hour	Used for the new Operational Overview screen added in V12.0.1. This value is used to calculate the expected unloading time for appointment lines received at the PALLET level.	Number	Y
receipt_level	Determines the level at which the receipt uploads are processed. Valid values are 'A'ppointment and 'C'ontainer.	Text	Y
recv_audit_queue	Line printer queue where the Receiving Audit List prints.	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 unit capacity, the system generates a replenishment pick. This is used when a forward picking location is not assigned a specific replenishment level. It is also the default level used on the startup/convert RF screens.	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
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

Table 3–4 File Management

Name	Purpose	Type	Allow User Edit
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
rop_dist_method	Determines if the distribution method is based on FIFO or PROCESS.	Text	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_door_scan	When set to "Y", this forces the user to scan both the door and the container id when loading containers in shipping	Y/N	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_requried	Flag that allows the DC to specify if the seal number is required when shipping.	Y/N	Y
ship_stage	Default location used the CSR table when creating routes using third 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 receives 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

Table 3–4 File Management

Name	Purpose	Type	Allow User Edit
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
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 receives 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
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

Table 3–4 File Management

Name	Purpose	Type	Allow User Edit
transport_balance	When Transport_balance is set to Yes, RWMS will send the inventory balance of transport assets to the host. When set to No, RWMS will NOT send the inventory balance of transport assets to the host.	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
unit_block_dist_flag	When set to Y, RWMS distributes units in Block. Block indicates that shortages are borne by the lower priority destinations. When set to N, RWMS 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
unlocated_location	Location of lost containers, those that cannot be found during a cycle count. Used in the Count Location screen.	Location	Y
upld_convert_inv_adj	When set to Y, RWMS 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

Table 3–4 File Management

Name	Purpose	Type	Allow User Edit
virtual_distro	Distribution 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, RWMS sets Saturday as a working day. Used in the Working Days Editor.	Y/N	Y
work_on_sunday	When set to Y, RWMS 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
xzone_bulk_picking	Applies to Bulk picking. Once started picking the bulk picking form restricts the user to zone of the first pick performed for subsequent picks. Users need to F3-exit every time bulk picks from current zone have been exhausted to scan next container ID in term of zone change. If = 'Y' the pick continues after the 1st container from new zone has been scanned. If = 'N' the pick is stopped short after the 1st container from new zone has been scanned.	Y/N	Y
xzone_grp_fcpgen_pick	For FCP to pallet w/generic labels. When a user scans a location to begin this type of picking, we need to know whether to look in all zones and zone groups for a pick or restrict only to the zone/zone group for the scanned location. If = Y, we look all over. If=N we only look in the zone or zone group for the location that was scanned as the start loc.	Y/N	Y
xzone_pick	When set to Y, the distribution process creates pick across multiple zones for the same distribution. When set to N, cross-zone picking, for the same distribution, is denied.	Y/N	Y

Internationalization

Internationalization is the process of preparing software in order to ensure that it can efficiently handle multiple languages. In other words, the software is created so that it can be released into international markets.

RWMS is internationalized to handle multiple languages in the same instance. That is, the technical infrastructure of RWMS supports languages other than English. An application that runs in various languages must transform into somewhat of a 'generic' product. Locale-specific information is placed in files external to the application.

This section describes configuration settings and features of RWMS that ensure that the base application can handle multiple languages.

Supported Languages

The supported languages are:

- Brazilian Portugese
- French
- German
- Italian
- Korean
- Spanish
- Simplified Chinese
- Traditional Chinese

Translation

Translation is the process of interpreting and adapting text from one language into another. Although the code itself is not translated, components of the application that are translated may include the following, among others:

- Graphical user interface (GUI)
- Error messages

RWMS Tables

RWMS stores translated text in three main tables:

Table 3–5 Internationalization Tables

Table	Description
DMS_LANGUAGE_MENU	contains the strings for the menus
TRANSLATOR	contains the strings for the forms
USER_LANGUAGE_MESSAGE	contains the strings for the messages and alerts

The DMS_LANGUAGE_MENU table contains the strings for the menus for each installed language:

Table 3–6 DMS_LANGUAGE_MENU Table

FACILITY_ID	MENU_NAME	OPTION_TITLE	LANGUAGE_ CODE	OPTION_TEXT
PR	ITEM_SETUP_MENU	Transport Asset Editor	AM (American English)	Transport Asset Editor
PR	ITEM_SETUP_MENU	Transport Asset Editor	For example -FR (French)	Editeur transport élément d'actif
PR	DISTRIBUTION_ MENU	Order Query Editor	AM	Order Query Editor
PR	DISTRIBUTION_ MENU	Order Query Editor	FR	Editeur demande commande

The translator table contains the strings for the forms for each installed language:

Table 3–7 Translator Table

FACILITY_ID	LANGUAGE_CODE	DATA_BASE_VALUE	DISPLAY_VALUE
PR	AM (American English)	APPROVE	Approve
PR	For example - FR (French)	APPROVE	Approuver
PR	AM	PRODUCT	Product
PR	FR	PRODUCT	Produit

The USER_LANGUAGE_MESSAGE table contains the strings for the messages and alerts for each installed language:

Table 3–8 USER_LANGUAGE_MESSAGE Table

FACILITY_ID	LANGUAGE_CODE	MESSAGE_CODE	MESSAGE_TEXT
PR	AM (American English)	ITEM_NOT_AVAIL	Required item not available.
PR	For example - FR (French)	ITEM_NOT_AVAIL	Article requis non disponible.
PR	AM	DUP_WAVE_ITEM	Item already exists from a different wave
PR	FR	DUP_WAVE_ITEM	Article existe déjà pour une rafale différente

4

DBA Administration Module

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

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

The following procedures are included in this section:

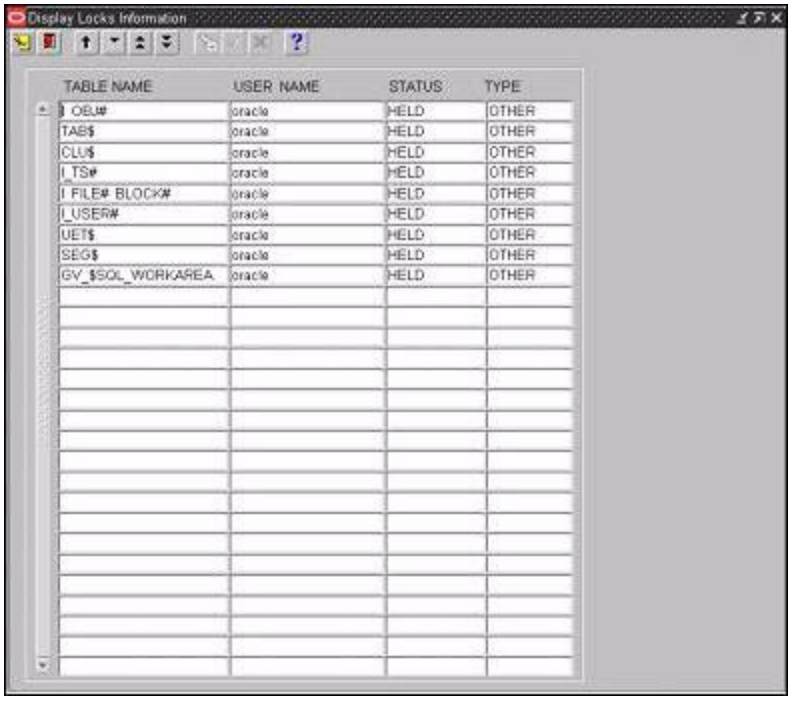
- Display Locks on Tables
- Display Table Information
- Display Tablespace Information
- Display Rollback Information
- Display Index Information
- Display Sequences Information
- Display the Error Log
- View Error Log Details
- Delete Error Log Records
- Print the Error Log Report

Display Locks on Tables

The user can use the Display Locks on Tables to display the status of the tables.

1. Select the DBA Administration menu.
2. Select the Display Locks on Tables option. The Table Locks Monitoring screen is displayed.

Figure 4–1 Table Locks Monitoring Screen



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

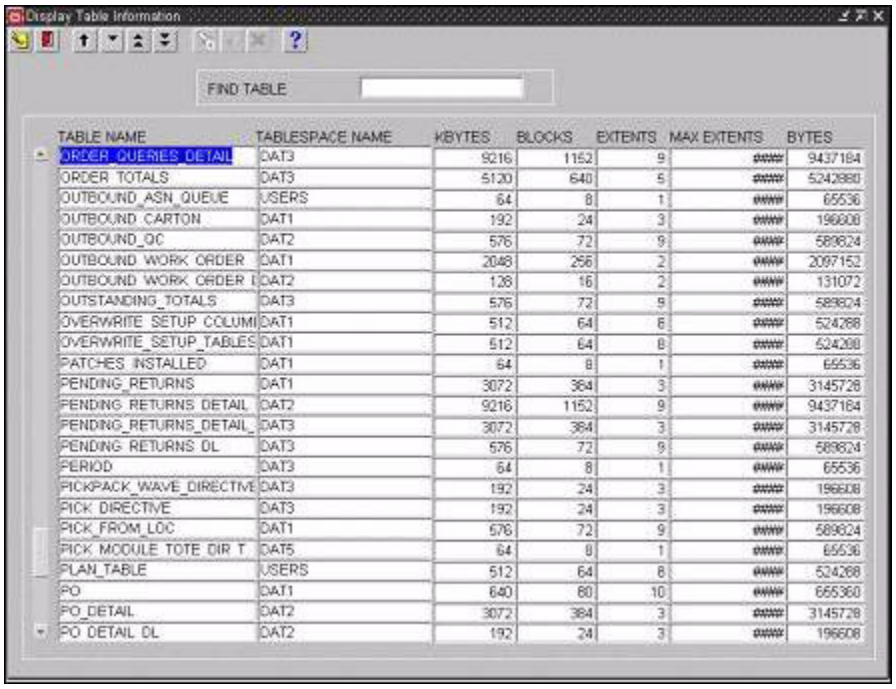
Display Table Information

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

> **Note:** The user 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.

Figure 4–2 Table Information Screen



The fields on the Table Information screen are explained in the Table 4-1:

Table 4–1 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 the user wants to display information about a particular table, enter the specific table name at the Find Table field. The user can also enter a partial table name. For example, the user can enter APP to display all tables that begin with these letters.

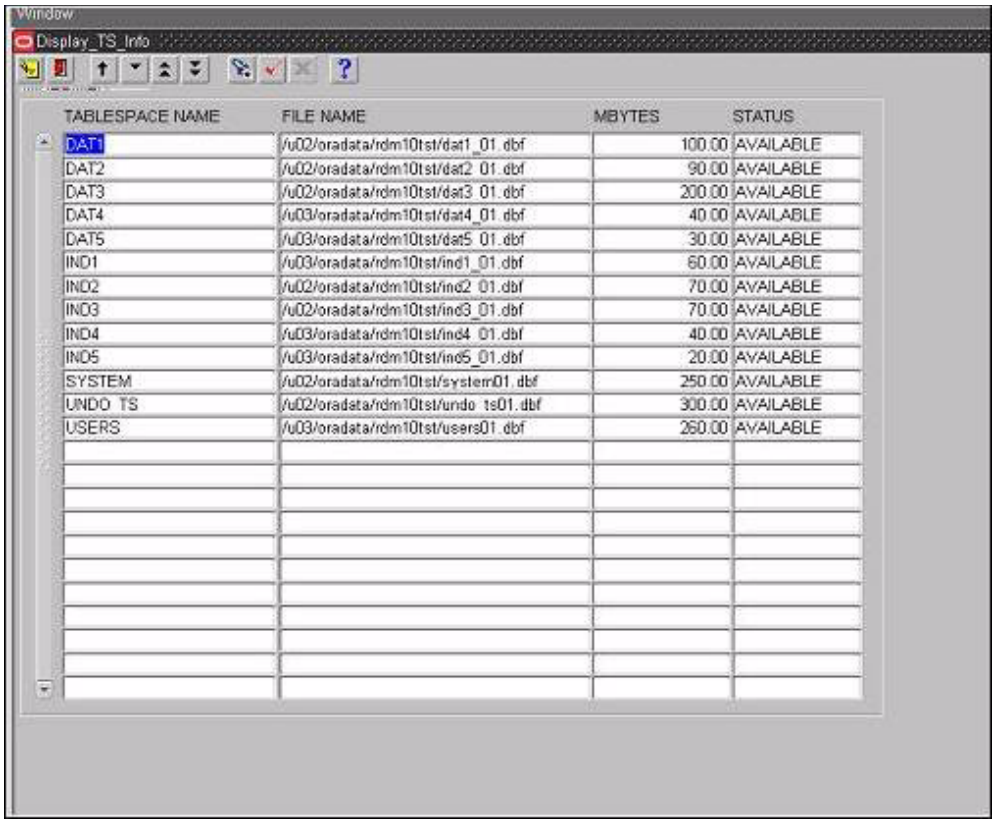
4. Click the **Exit** button to return to the DBA Administration menu.

Display Tablespace Information

The user 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.

Figure 4–3 Tablespace Information Screen



The fields on the Tablespace Information screen are explained in the Table 4-2:

Table 4–2 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.

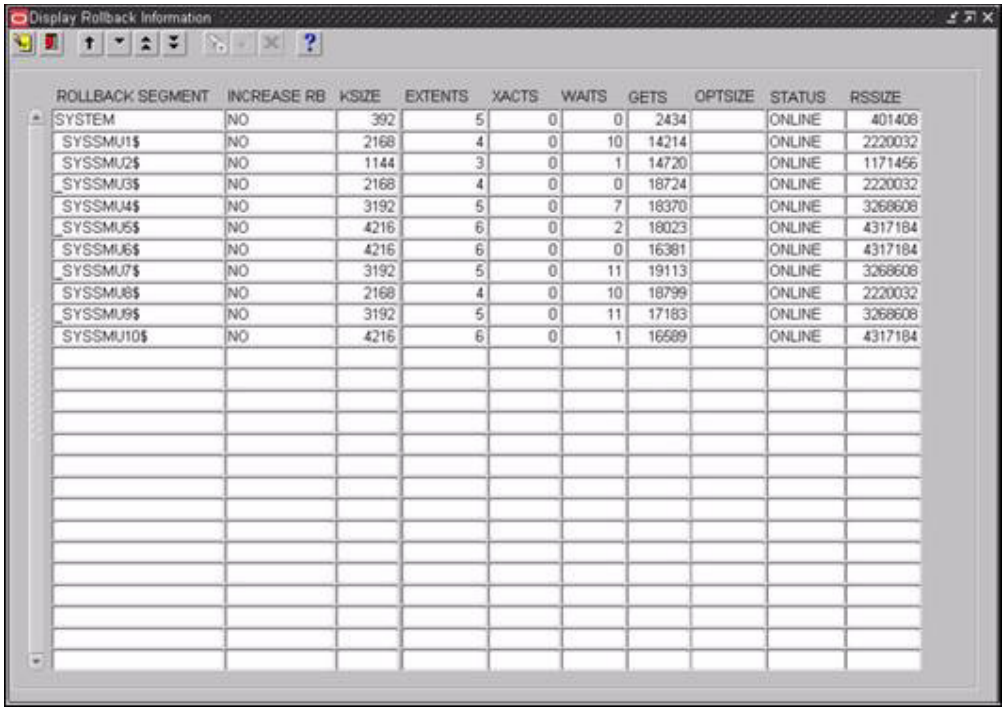
3. Click the **Refresh** button to refresh the screen and view any new tablespace information.

Display Rollback Information

The user can use the Display Rollback Information option to display information about rollbacks. The user 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. The Rollback Information screen is displayed, along with all rollback segments.

Figure 4–4 Rollback Information Screen



ROLLBACK SEGMENT	INCREASE RB	KSIZE	EXTENTS	XACTS	WAITS	GETS	OPTSIZE	STATUS	RSSIZE
SYSTEM	NO	392	5	0	0	2434		ONLINE	401408
SYSSMU1$	NO	2168	4	0	10	14214		ONLINE	2220032
SYSSMU2$	NO	1144	3	0	1	14720		ONLINE	1171456
SYSSMU3$	NO	2168	4	0	0	18724		ONLINE	2220032
SYSSMU4$	NO	3192	5	0	7	18370		ONLINE	3268608
SYSSMU5$	NO	4216	6	0	2	18023		ONLINE	4317184
SYSSMU6$	NO	4216	6	0	0	16381		ONLINE	4317184
SYSSMU7$	NO	3192	5	0	11	19113		ONLINE	3268608
SYSSMU8$	NO	2168	4	0	10	18799		ONLINE	2220032
SYSSMU9$	NO	3192	5	0	11	17183		ONLINE	3268608
SYSSMU10$	NO	4216	6	0	1	16589		ONLINE	4317184

The fields on the Rollback Information screen are explained in the Table 4-3:

Table 4–3 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

Table 4–3 (Cont.) Fields on the Rollback Information Screen

Field Name	Field Description
STATUS	Status (Online/Offline)
RRSIZE	Rollback Size

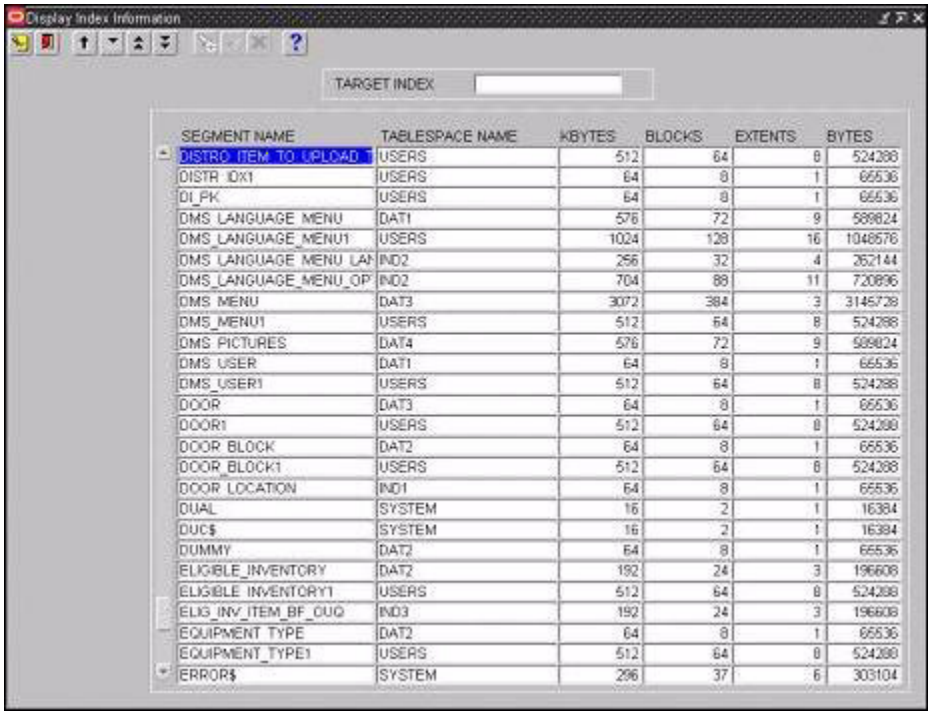
2. Click the **Refresh** button to refresh the screen and view any new information.
3. Click the **Exit** button to return to the DBA Administration menu.

Display Index Information

The user can use the Display Index Information option to display sizing information for the database indexes. The user 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.

Figure 4–5 Index Information Screen



The fields on the Index Information screen are explained in the Table 4-4:

Table 4–4 Fields 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.

Table 4–4 (Cont.) Fields on the Index Information Screen

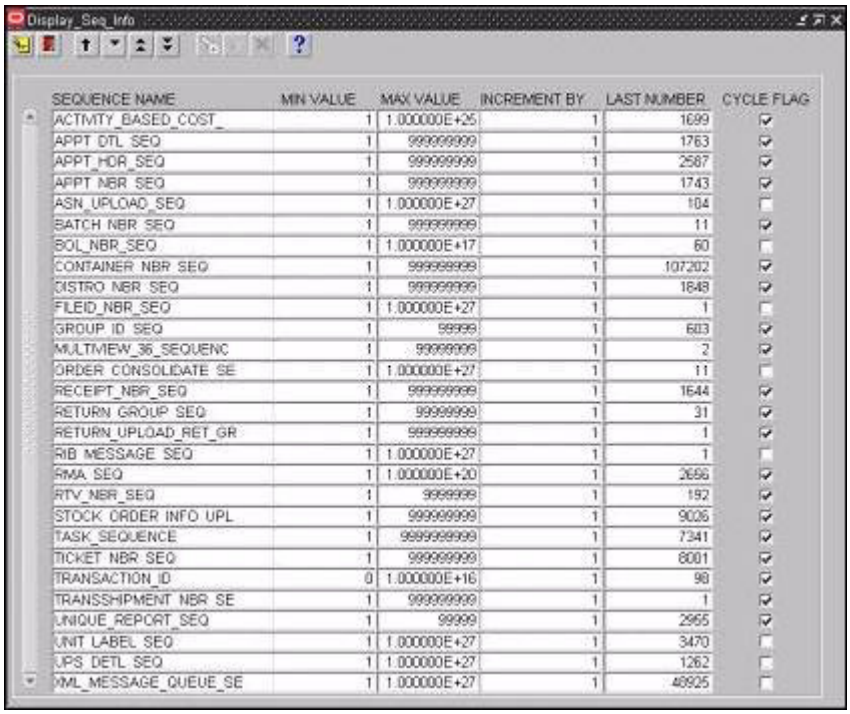
Field Name	Field Description
Kbytes	Size of index in kilobytes.
Blocks	Number of blocks the index is using (1 block = 4096 bytes).
Extents	Current number of index extents.
Bytes	Size of index in bytes.

Display Sequences Information

The user can use the Display Sequence Information option to display sizing information specific to the sequences used by the system. The user 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. The Sequences Information screen is displayed, along with the sequence information already entered into the system.

Figure 4–6 Sequences Information Screen



The fields on the Sequences Information screen are explained in the Table 4-5:

Table 4–5 Fields on 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.

Table 4–5 (Cont.) Fields on Sequences Information Screen

Field name	Field description
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.
Cycle Flag	Cycle, whether the sequence rolls back to minimum value when the maximum value is reached.

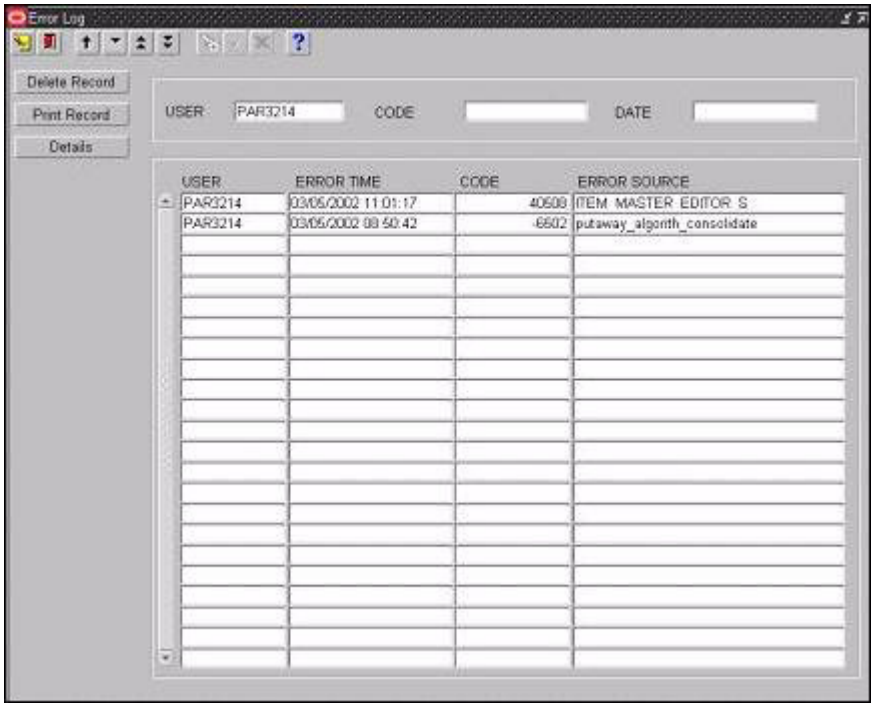
2. Click the **Refresh** button to refresh the screen and view any new information.
3. Click the **Exit** button to return to the Main menu.

Display the Error Log

Display Error Log option to display, view details, and delete logged errors. These are unanticipated errors or errors occurring in background processes. The user can also print the error log.

1. On the Monitoring and Administration menu, select the Display Error Log option.
2. Click the **Display** button. The Error Log Inquiry screen is displayed, as shown in the following:

Figure 4–7 Error Log Inquiry Screen



The fields on the Error Log Inquiry screen are explained in the Table 4-6:

Table 4–6 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

3. 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.

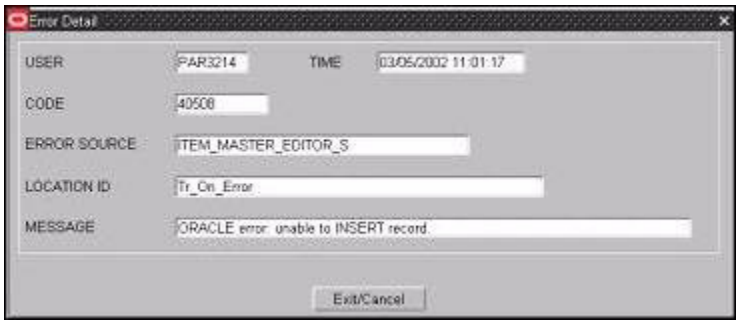
4. To display a specific error for a particular user, enter the user name in the Enter User field and the error code in the Code field.
5. To display a specific error for a specific user for a particular date, enter the user name in the Enter User field, the error code in the Code field, and the date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

> **Note:** RWMS accepts any combination of the above fields.

View Error Log Details

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. The Error Log Inquiry screen is displayed.
3. Enter the user, error code, or date 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.
4. Select the record to view in more detail, using the keypad arrow keys to move up and down the list, and then click the **Details** button. The Error Log screen is displayed.

Figure 4–8 Error Log screen



The fields on the Error Log screen are explained in the Table 4-7:

Table 4–7 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 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 to delete, using the keypad arrow keys to move up and down the list, and then click the Delete button. A message box asks to confirm the deletion. The message reads:

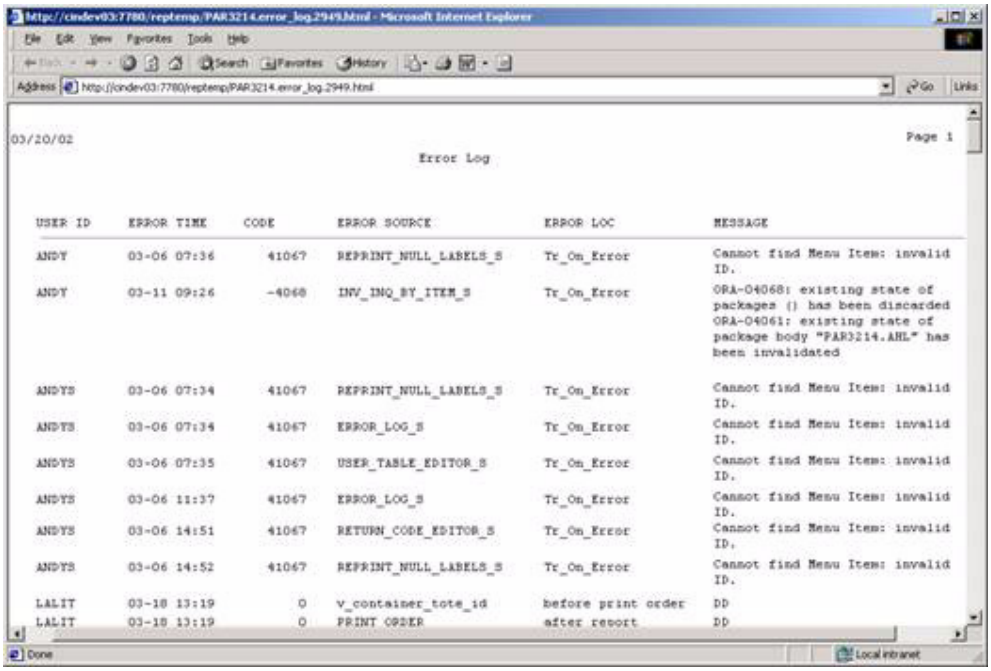
   ```
   Confirm Delete Operation (Yes/No)
   ```

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 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.

Figure 4–9 Error Log Report



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

The fields on the Error Log Report screen are explained in the Table 4-8:

Table 4–8 Fields on Error Log Report

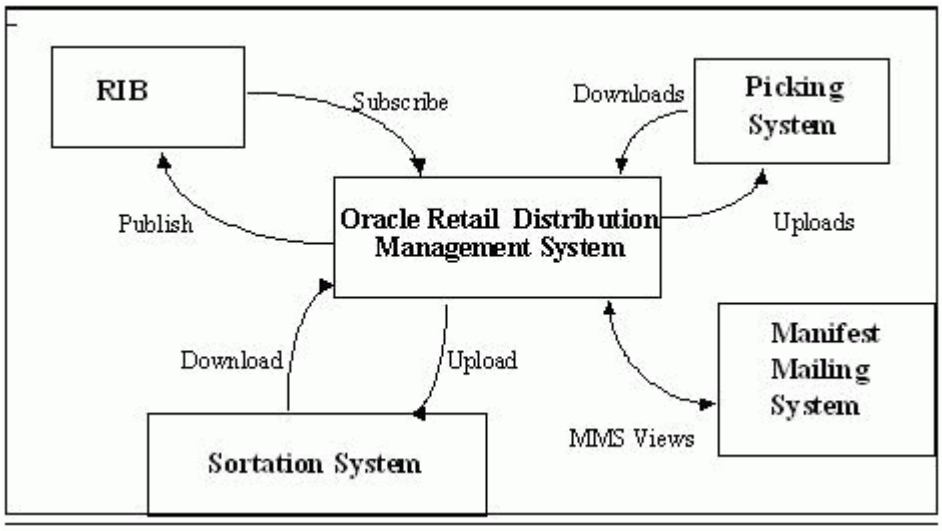
Field Name	Field Description
User ID	Identification of the user who had the error.
Error Time	Date and time the error was logged.
Code	The error code.
Error Source	Program where the error originated.
Error Loc	The location (procedure, block, and so on.) within the source program where the error occurred.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

5

RWMS RIB Components

This chapter and the next describe the various interfaces with the RWMS. These include Host System through the RIB, Picking System, Sortation System, and Manifest Mailing System (MMS) links.

Figure 5–1 Integration Points to RWMS



Acronyms and Terms used in RIB Components

The following section introduces the acronyms and terms used in RIB Components.

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. This is 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 RWMS.
- **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 RWMS, but may be used in a future release.
- **Host** – The controlling computer system. Often housed at corporate headquarters.
- **Item** – A specified part number, SKU, and so on.
- **Optional** – The field is used for information purposes and is not required.
- **Pre-distribution** – 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** – Oracle Retail Integration Bus.
- **Upload** – Any data file going out from RWMS to another system.
- **Vendor** – A supplier of in-bound goods. Each PO is assigned to a vendor.

RWMS 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 RWMS adapter recognizes that a message with the specific name with a prefix of RDMSUB (for example, 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 (for example, 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 returns an S and the error_message is null. The status code might return with an error code if the message processing failed and the error_message contains 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.

  Note: Detailed information is provided in the corresponding component section of the specific subscription.

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

  ```
  PUB_XXXX
  ```

  This procedure accepts inputs such as facility_type/dc_dest_id/pub_seq.

  It returns an Oracle Object containing the message information along with the message type (for example, 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.

  Note: Detailed information is provided in the corresponding component section of the specific subscription.

Subscription Components

The following types of subscriptions are available:

- Vendor Subscription
- Location Subscription
- Item Subscription
- UDA Subscription
- Differentiator Subscription
- Purchase Order Subscription
- Inbound Work Order Subscription

- Inbound ASN Subscription
- Stock Order Subscription
- Outbound Work Order Subscription
- Pending Returns Subscription

Vendor Subscription

Vendor messages are used by RWMS to create and maintain Vendor and Vendor Address information. Vendor messages are published by a host system.

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

This family of messages is considered to be Foundation Data. Foundation Data is used as the basis for building other data models and is routed to every RWMS 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 the Name, as well as the auditing and sampling requirements for received product.
- Address record - Address Type (for example, Billing, Shipping), Primary Indicator, and basic address information.

Vendor Message Components

The following is a description of the Vendor message components:

- Name of the Consume method: RDMSUB_VENDOR
- Name of the Process Message method: SUB_VENDOR

> **Note:** For a general description of the Consume and Process Message methods please refer to the preceding RWMS Message subscription process section of this document.

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, the structure, and the associated Rib_Object.

Table 5–1 Vendor Message Family

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

Table 5–1 (Cont.) Vendor Message Family

Message	Structure	RIB_Object Type
Vendor Address Delete (VDRD_DEL)	Address only	Rib_VendorAddrRef_Rec

Primary Vendor Tables

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

- VENDOR
- VENDOR_ADDRESS

> **Note:** Detailed descriptions of these tables are in the RWMS Data Model document.

Status and Error Messages

For status and error messages returned from the API, see the Chapter 7, "Appendix: Error Codes".

Location Subscription

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

Ship Destination 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 RWMS 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:

- Name of the Consume method: RDMSUB_LOC
- Name of the Process Message method: SUB_LOC

> **Note:** For a general description of the Consume and Process Message methods please refer to the preceding RWMS Message subscription process section of this document.

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, the structure, and the associated Rib_Object.

Table 5–2 Location Message Family

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 RWMS that hold Location data:

- SHIP_DEST

> **Note:** Detailed descriptions of these tables are in the RWMS Data Model document.

Status and Error Messages

For status and error messages returned from the API, see the Chapter 7, "Appendix: Error Codes".

Item Subscription

The Item messages are used by RWMS 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 RWMS 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 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:

- Name of the Consume method: RDMSUB_ITEMS
- Name of the Process Message method: SUB_ITEMS

> **Note:** For a general description of the Consume and Process Message methods please refer to the preceding RWMS Message subscription process section of this document.

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, the structure, and the associated Rib_Object.

Table 5–3 Item Message Family

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
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

Table 5–3 (Cont.) Item Message Family

Message	Structure	RIB_Object Type
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 RWMS that hold Item data:

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

Note: Detailed descriptions of these tables are in the RWMS Data Model document.

Status and Error Messages

For status and error messages returned from the API, see the Chapter 7, "Appendix: Error Codes".

UDA Subscription

UDA Information, known in RWMS 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 RWMS 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:

- Name of the Consume method: RDMSUB_UDAS.
- Name of the Process Message method: SUB_UDAS.

> **Note:** For a general description of the Consume and Process Message methods please refer to the preceding RWMS Message subscription process section of this document.

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, the structure, and the associated Rib_Object.

Table 5–4 UDA Message Family

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 RWMS that hold UDA data:

- ATTRIBUTES
- ATTRIBUTE_TYPE

> **Note:** Detailed descriptions of these tables are in the RWMS Data Model document.

Status and Error Messages

For status and error messages returned from the API, see the Chapter 7, "Appendix: 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 RWMS 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:

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

> **Note:** For a general description of the Consume and Process Message methods please refer to the preceding 'RWMS Message subscription process section of this document.

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, the structure, and the associated Rib_Object.

Table 5–5 Differentiator Message Family

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 RWMS that holds Differentiator data:

- DIFFERENTIATOR
- DIFFERENTIATOR_GROUP
- DIFF_GROUP_DETAIL

> **Note:** Detailed descriptions of these tables are in the RWMS Data Model document.

Status and Error Messages

For status and error messages returned from the API, see the Chapter 7, "Appendix: Error Codes".

Purchase Order Subscription

Purchase Order (PO) messages are used by RWMS 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:

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

> **Note:** For a general description of the Consume and Process Message methods please refer to the preceding RWMS Message subscription process section of this document.

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, the structure, and the associated Rib_Object.

Table 5–6 PO Message Family

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 (PODtlPhysCre)	Header and Detail	Rib_PoPhyDesc_Rec
Purchase Order Detail Modify (PODtlPhysMod)	Header and Detail	Rib_PoPhyDesc_Rec
Purchase Order Detail Delete (PODtlPhysDel)	Detail only	Rib_PoDtlRef_Rec

Primary Purchase Order Tables

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

- PO
- PO_DETAIL

Note: Detailed descriptions of these tables are in the RWMS Data Model document.

Status and Error Messages

For status and error messages returned from the API, see the Chapter 7, "Appendix: Error Codes".

Inbound Work Order Subscription

Inbound Work Order messages are used by RWMS 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:

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

Note: For a general description of the Consume and Process Message methods please refer to the preceding RWMS Message subscription process section of this document.

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, the structure, and the associated Rib_Object.

Table 5–7 Inbound Work Order Message Family

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 RWMS that hold Inbound Work Order data:

- INBOUND_WORK_ORDER

Note: Detailed descriptions of these tables are in the RWMS Data Model document.

Status and Error Messages

For status and error messages returned from the API, see the Chapter 7, "Appendix: Error Codes".

Inbound ASN Subscription

Inbound Advanced Ship Notice (ASN) messages are used by RWMS 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 ASN 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 styles 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.

- **POrecord** - Purchase Order information.
- **Carton** - Container Identifier, dimensions (for Carton Type ASNs).
- **Items** - Details about all items in the Container.

Inbound ASN Message Components

The following is a description of the BOL message components:

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

> **Note:** For a general description of the Consume and Process Message methods please refer to the preceding RWMS Message subscription process section of this document.

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, the structure, and the associated Rib_Object.

Table 5–8 ASN Message Family

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 RWMS that hold ASN data:

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

> **Note:** Detailed descriptions of these tables are in the RWMS Data Model document.

Status and Error Messages

For status and error messages returned from the API, see the Chapter 7, "Appendix: Error Codes".

Stock Order Subscription

Stock Order messages are used by RWMS 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 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:

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

> **Note:** For a general description of the Consume and Process Message methods please refer to the preceding RWMS Message subscription process section of this document.

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, the structure, and the associated Rib_Object.

Table 5–9 Stock Order Message Family

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_SoDtlRef_Rec

Primary Stock Order Tables

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

- STOCK_ORDER
- STOCK_ALLOCATION
- COMPONENT_TICKETING
- STOCK_ALLOCATION_CID

> **Note:** Detailed descriptions of these tables are in the RWMS Data Model document.

Status and Error Messages

For status and error messages returned from the API, see the Chapter 7, "Appendix: Error Codes".

Outbound Work Order Subscription

Outbound Work Order messages are used by RWMS 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: distribution, 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:

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

> **Note:** For a general description of the Consume and Process Message methods please refer to the preceding RWMS Message subscription process section of this document.

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, the structure, and the associated Rib_Object.

Table 5–10 Outbound Work Order Message Family

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 RWMS that hold Outbound Work Order data:

- OUTBOUND_WORK_ORDER

Note: Detailed descriptions of these tables are in the RWMS Data Model document.

Status and Error Messages

For status and error messages returned from the API, see the Chapter 7, "Appendix: Error Codes".

Pending Returns Subscription

Pending Return messages are used by RWMS 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, and Receipt Date.
- **Detail record** - The item and quantity.

Pending Returns Message Components

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

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

Note: For a general description of the Consume and Process Message methods please refer to the preceding RWMS Message subscription process section of this document.

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, the structure, and the associated Rib_Object.

Table 5–11 Pending Return Messages Family

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 RWMS that hold Pending Returns data:

- PENDING_RETURNS
- PENDING_RETURNS_DETAIL

Note: Detailed descriptions of these tables are in the RWMS Data Model document.

Status and Error Messages

For status and error messages returned from the API, see the Chapter 7, "Appendix: Error Codes".

Publish Components

The following types of publishing components are available:

- Inbound ASN Publish
- Appointments/Receipts Publish
- Stock Order Status Publish
- Outbound ASN Publish
- Inventory Adjustments Publish
- Customer Returns Publish

- Return to Vendor Publish
- Space Locations Publish
- Item Warehouse Publish
- Location Publish
- Equipment Publish
- Equipment Class Publish

Inbound ASN Publish

RWMS is responsible for communicating Inbound ASN Information to the Host System. Inbound ASN is defined as ASN Information originating in the RWMS System. Inbound ASNs can be Container or PO Type ASNs. PO Type ASNs detail item information is received at a unit level, not container level information. Container Type Inbound ASNs detail item information is 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 is 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 RWMS tables are populated when a record is recreated in the RWMS screens. These are the tables that stage the ASN records to be published:

- *ASN_UPLOAD*
- *ASN_ITEM_UPLOAD*
- *ASN_CONT_UPLOAD*
- *ASN_PO_UPLOAD*

Inbound ASN Message Components

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

> **Note:** For a general description of the GetNxt and message builder methods please refer to the preceding RWMS 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

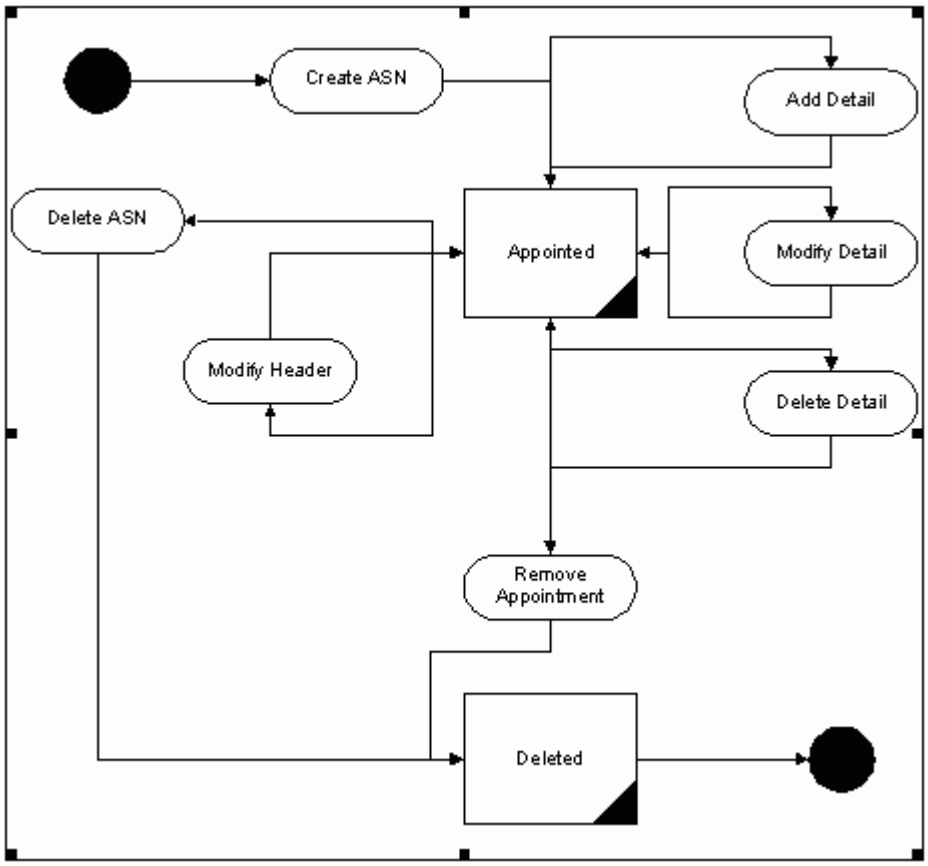
All Inbound ASN messages belong to the Inbound ASN message family. The structure of the message depends on the message type to be performed. The following table lists each message, the structure, and the associated Rib_Object.

Table 5–12 Inbound ASN Messages Family

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

Figure 5–2 Description of Activities



Create Inbound ASN Messages

1. Prerequisites: Must be an 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 an 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 Chapter 7, "Appendix: Error Codes".

Appointments/Receipts Publish

RWMS 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 is scheduled. Once scheduled, Appointment messages are communicated at the addition, modification, or deletion of a detail, a modification of the header information such as arrival time, or at Open, Close, and Deletion of the appointment.

RWMS 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 RWMS system parameter. Receipt Info Upload includes 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 RWMS tables are populated when a record is created in RWMS. These are the tables that stage the Receipts and Appointments:

- APPT_DETAIL_TO_UPLOAD
- APPT_HEADER_TO_UPLOAD
- RECEIPT_TO_UPLOAD

Receipt/Appointment Message Components

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

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

Note: For a general description of the GetNxt and message builder methods please refer to the preceding RWMS Message publication process section of this document.

Message Summary

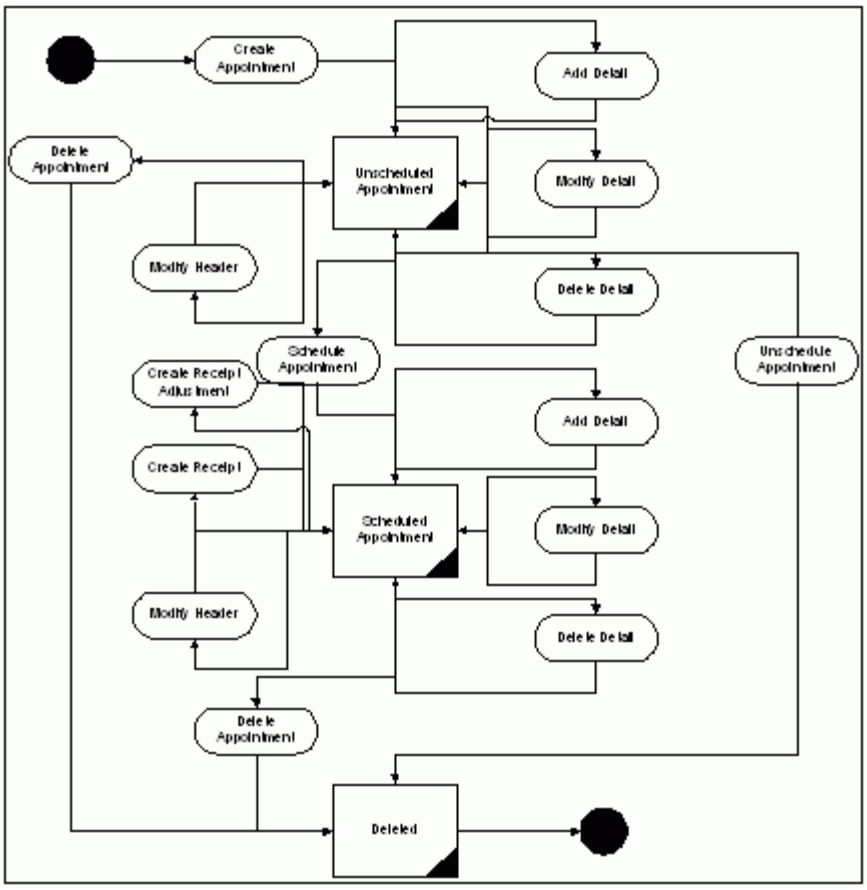
All Receipt and Appointment messages belong to the Receipt message family. The structure of the message depends on the message type to be performed. The following table lists each message, the structure, and the associated Rib_Object.

Table 5–13 Receipt Message Family

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

Figure 5–3 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 is received and the initial receipt upload is sent.
2. Activity Detail: Each container is individually checked using RWMS 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 Chapter 7, "Appendix: Error Codes".

Stock Order Status Publish

RWMS is responsible for communicating Stock Order status Information to the Host System. RWMS generates 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 distribution number, distribution type, item information, and quantities, and status.

Stock Order Status Tables

The RWMS tables are populated when a record is created in RWMS. This is the table that stages 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:

Note: For a general description of the GetNxt and message builder methods please refer to the preceding RWMS 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

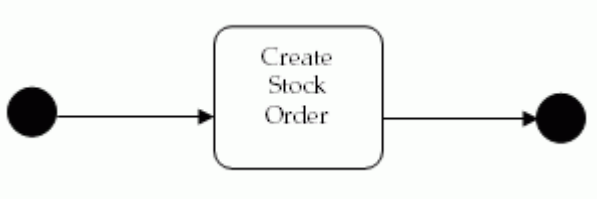
All Stock Order Status messages belong to the Stock Order Status message family. The structure of the message depends on the message type to be performed. The following table lists each message, the structure, and the associated Rib_Object.

Table 5–14 Stock Order Message Family

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

State Diagram

Figure 5–4 Description of Activities



Create Stock Order Info Messages

1. Prerequisites: Valid distribution 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 Chapter 7, "Appendix: Error Codes".

Outbound ASN Publish

RWMS 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 Distribution Information.

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

Outbound ASN Tables

The RWMS tables are populated when a record is recreated in the RWMS screens. This is the table that stages 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:

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

> **Note:** For a general description of the GetNxt and message builder methods please refer to the preceding RWMS Message publication process section of this document.

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, the structure, and the associated Rib_Object.

Table 5–15 Outbound ASN Message Family

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

State Diagram

Figure 5–5 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 Chapter 7, "Appendix: Error Codes".

Inventory Adjustments Publish

RWMS 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 (for example, from unavailable to sell, to available to sell). True inventory adjustments must always have a disposition change; however, the user 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 is included)
- Trouble (Trouble code is included)
- Distributed

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

Inventory Adjustments Tables

The RWMS tables are populated when a record is recreated in The RWMS screens. This is the table that stages 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:

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

> **Note:** For a general description of the GetNxt and message builder methods please refer to the preceding RWMS Message publication process section of this document.

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, the structure, and the associated Rib_Object.

Table 5–16 Inventory Adjustment Message Family

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

State Diagram

Figure 5–6 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 Chapter 7, "Appendix: Error Codes".

Inventory Balance Upload

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

Table 5–17 Format of Inventory Balance Record

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.

Table 5–17 (Cont.) Format of Inventory Balance Record

Field Description	Template	Description
Distributed Units	N (8) v N (4)	Units distributed includes: 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: 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 exists: container status of: I, D, M, R, T, X. For catch weight items.

Customer Returns Publish

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

RWMS provides the capability to process item level return information. Information to the host upon completion of the process includes: 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 RWMS tables are populated when a record is created in RWMS. This is the table that stages the Customer Returns records to be published:

- RETURNS_UPLOAD

Customer Returns Message Components

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

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

Note: For a general description of the GetNxt and message builder methods please refer to the preceding RWMS Message publication process section of this document.

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, the structure, and the associated Rib_Object.

Table 5–18 Customer Return Message Family

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

State Diagram

Figure 5–7 Description of Activities

Create Customer Returns

1. Prerequisites: There are no prerequisites for Customer Returns.
2. Activity Detail: There is no activity details, once the message is 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 Chapter 7, "Appendix: Error Codes".

Return to Vendor Publish

RWMS 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 RWMS tables are populated when a record is posted in RWMS. 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:

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

> **Note:** For a general description of the GetNxt and message builder methods please refer to the preceding RWMS Message publication process section of this document.

Message Summary

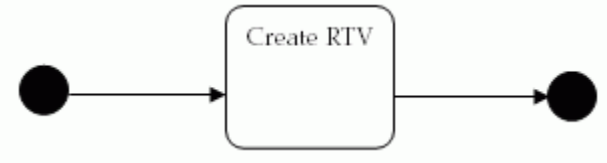
All Return to Vendor messages belong to the Return to Vendor message family. The structure of the message depends on the message type to be performed. The following table lists each message, the structure, and the associated Rib_Object.

Table 5–19 Return to Vendor Message Family

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

State Diagram

Figure 5–8 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 Chapter 7, "Appendix: Error Codes".

Streamsoft Components

Space Locations Publish

RWMS 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 RWMS. 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 (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 as 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 is SKU optimized and is assigned for SKU publishing. In this case, the location information for the picking location where the item assigned/unassigned is sent.

The information sent to the third-party system on an add or modify includes: 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, and so on. Deletion of location information includes only the DC Destination ID (from location) and the location ID for the location being deleted.

Space Location Tables

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

- LOCATION_UPLOAD

Space Location Message Components

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

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

Note: For a general description of the GetNxt and message builder methods please refer to the preceding RWMS Message publication process section of this document.

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, the structure, and the associated Rib_Object.

Table 5–20 Space Location Message Family

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 Chapter 7, "Appendix: 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 RWMS 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:

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

> **Note:** For a general description of the Consume and Process Message methods please refer to the preceding RWMS Message subscription process section of this document.

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, the structure, and the associated Rib_Object.

Table 5–21 SKUOptm Message Family

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 RWMS that hold SKU Optimization data:

- TASK_QUEUE

Note: Detailed descriptions of these tables are in the RWMS Data Model document.

Status and Error Messages

For status and error messages returned from the API, see the Chapter 7, "Appendix: Error Codes".

Item Warehouse Publish

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

RWMS 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.

- Modification or deletion of optimized and published item/SKU (item has sku_optimized and sku_opt_published flags set to Y).
- Modification or deletion of optimized and published SKU supplier information.
- Modification or deletion of optimized and published SKU supplier country information.
- Modification or deletion of optimized and published SKU supplier country DIM information.
- Creation, modification, or deletion of optimized and published SKU association to a forward picking/case picking location.
- First association of optimized SKU to an appointment.
- First association of optimized SKU to an 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 includes only the DC Destination ID (from location) and the item ID for the item being deleted.

Status and Error Messages

For status and error messages returned from the API, see the Chapter 7, "Appendix: Error Codes".

Item Tables

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

- ITEM_MASTER_UPLOAD

Item Warehouse Message Components

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

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

> **Note:** For a general description of the GetNxt and message builder methods please refer to the preceding RWMS Message publication process section of this document.

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, the structure, and the associated Rib_Object.

Table 5–22 Item Warehouse Message Family

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

Oracle Retail Labor Management (RLM) Components

Transaction Publish

RWMS 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 is added or modified in the RWMS system.

Transaction Tables

The RWMS tables are populated when a record is recreated in the RWMS screens. This is the table that stages the Transaction records to be published:

- TRANSACTION_UPLOAD

Warehouse Transaction Message Components

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

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

> **Note:** For a general description of the GetNxt and message builder methods please refer to the preceding RWMS Message publication process section of this document.

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, the structure, and the associated Rib_Object.

Table 5–23 Warehouse Transaction Message Family

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 Chapter 7, "Appendix: Error Codes".

Location Publish

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

Location messages are communicated to the Host once the Location is added, modified, or deleted out of the RWMS system.

Location Tables

The RWMS tables are populated when a record is recreated in the RWMS screens. This is the table that stages the Location records to be published:

- LMS_LOCATION_UPLOAD

Location Message Components

The following is a description of the Location message components:

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

> **Note:** For a general description of the GetNxt and message builder methods please refer to the preceding RWMS Message publication process section of this document.

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, the structure, and the associated Rib_Object.

Table 5–24 Warehouse Space Location Message Family

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

Table 5–24 Warehouse Space Location Message Family

Message	Structure	RIB_Object Type
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 Chapter 7, "Appendix: Error Codes".

Equipment Publish

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

Equipment messages are communicated to the Labor Management System once the equipment is added, modified, or deleted out of The RWMS system.

Equipment Tables

The RWMS tables are populated when a record is recreated in The RWMS screens. This is the table that stages the Equipment records to be published:

- EQUIPMENT_UPLOAD

Equipment Message Components

The following is a description of the Equipment message components:

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

Note: For a general description of the GetNxt and message builder methods please refer to the preceding RWMS Message publication process section of this document.

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, the structure, and the associated Rib_Object.

Table 5–25 Warehouse Equipment Message Family

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 Chapter 7, "Appendix: Error Codes".

Equipment Class Publish

The RWMS 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 is added, modified, or deleted out of The RWMS system.

Equipment Class Tables

The RWMS tables are populated when a record is recreated in The RWMS screens. This is the table that stages 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:

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

> **Note:** For a general description of the GetNxt and message builder methods please refer to the preceding RWMS Message publication process section of this document.

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, the structure, and the associated Rib_Object.

Table 5–26 Warehouse Equipment Class Message Family

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 Chapter 7, "Appendix: Error Codes".

6

Subsystem Interfaces

This chapter consists of the following:

Batch File Formats

All batch files passed between an outside system and RWMS 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:

Table 6–1 Different Types of Fields

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.
YYYYMMDDHHMI	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.

Table 6–2 Store Orders

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 to be packed

Inbound Carton Download

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

Table 6–3 Carton Content

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 RWMS concerning contents of a picked container, the associated wave number and the outbound destination ID.

Table 6–4 Notification

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 is either RWMS or, as in standalone operations, some other application. Data is exchanged through text files. With text file data exchange, PBL is not 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. The files are listed in the order in which they are run because each download may depend upon a previous one.

Table 6–5 Files and Directories

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 RWMS and to allow the PBL to work standing alone, without RWMS. PBL download errors are recorded in the local RWMS error log, and are not uploaded to the host. The user 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 RWMS. They contain PBL built containers and the items and quantities in them that are 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 RWMS for this download is started by PBL. The script reads the files, loads the data into RWMS, and adds the container information to RWMS.

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

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

Table 6–6 Destination Container Download

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:

Table 6–7 Detail File

Field Description	Template	Description
Transaction Date/Time	YYYYMMDD HH24MISS	Date and time this record was created (future_use).

Table 6–7 (Cont.) Detail File

Field Description	Template	Description
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 RWMS.
- Container already exists in RWMS.
- 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 RWMS. Therefore, pick directive records are 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 RWMS for this download starts by PBL. The script reads the file, loads the data into RWMS, and updates the picking information in RWMS as required.

The format for the Distribution Item Download is as follows:

Table 6–8 Distribution Item Download

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 an 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.

Table 6–8 (Cont.) Distribution Item Download

Field Description	Template	Description
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 RWMS when there is a difference between the quantity sent on the Source Container Upload and the actual quantity picked. RWMS validates the data in the file and sends the information in an Inventory Adjustment Upload to the host system. This is the only action RWMS takes on this; no change in RWMS data occurs. After PBL creates the files, they are copied to the download directory by PBL. Then PBL starts the script in RWMS, or this download. The script reads the files, validates the data, and inserts the information into the Inventory Adjustment Upload table in RWMS 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:

Table 6–9 Inventory Adjustment Download

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 RWMS (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 RWMS from the host, or performed in the Destination Editor screen. Whenever an adjustment is performed, all shipping destinations that RWMS 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:

Table 6–10 Ship Destination Upload

Field Description	Template	Description
Transaction Date/Time	YYYYMMDD HHMISS	Date and time this record was created.
Record Type	A	Record type in PBL. Always an '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, are 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 FPL 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:

Table 6–11 Distro Item Upload

Field Description	Template	Description
Transaction Date/Time	YYYYMMDD HHMISS	Date and time this record was created.

Table 6–11 (Cont.) Distro Item Upload

Field Description	Template	Description
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 distribution/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 are needed by PBL. This file is built after each distribution run. Records are sorted by facility number, company number, distribution/order number, item, distribution/order creation time stamp, and container ID.

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

Table 6–12 Expected Source Container Upload

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 distribution/order was created.
Container ID	X (20)	Identifier for the container.
Requested Qty	N (8) v N (4)	Unit quantity that is 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 is 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:

Table 6–13 Source Container Upload

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 RWMS, the Oracle Retail Distribution Management Sortation module now sends container divert instruction messages to the sortation system to control the flow of containers on the conveyor.

Each message that RWMS 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 RWMS. However, subsequent messages are 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 RWMS of any container diverts that occur on the conveyor system. Depending on the type of divert that has taken place, RWMS 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.

Table 6–14 Files and Directories

Interface Name	Script Name	File Name
Container Divert Download	sorter_dnld.sh	sorter_dnld.dat

Table 6–14 (Cont.) Files and Directories

Interface Name	Script Name	File Name
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, RWMS performs an auto-receiving function. If the container is diverted to a processing area, RWMS updates the location of the container. When a divert to a shipping lane is sent, RWMS adds the container to the manifest for the trailer if one is available. For more details, refer to the RWMS Shipping Module in the *Oracle Retail Distribution Management User Guide*.

The Container Divert Message Download file has the following format:

Table 6–15 Container Divert Message Download

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

RWMS 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, RWMS only informs 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 RWMS.

When the receiving allocation module creates a container, RWMS calculates a pallet group identifier in order to provide a palletization operator a quickly recognized code that helps to group cartons together on pallets. RWMS 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:

Table 6–16 Container Divert Instruction Message Upload

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 MMS uses the merchandise carton ID to form an Oracle Data Base Compliant (ODBC) query into the RWMS 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 RWMS, is first taken to a shipping station. Each shipping station is a PC running a MMS with interfaces to a user interface terminal and often to a scale.

The label applied by the RWMS 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:

Table 6–17 Manifest Mailing System

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

MMS Views

The RWMS data base, which the MMS 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 MMScan change the value of the CARRIER_SERVICE_CODE, even if the field is not blank.

RWMS downloads the SHIP_TO_ADDRESS with the stock order. If this field is blank in the order download, the information is supplied by the SHIP_DEST table. The MMS 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:

Table 6–18 MMS Container

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
DL Comment	X (30)	Download comment that is 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:

Table 6–19 Container Item View

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)

MMS Tables

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

Manifest Mailing System Manifest Table

RWMS 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, RWMS looks up the corresponding BOL number in the ORACLE RETAIL DISTRIBUTION MANAGEMENT MANIFEST table. If the BOL does not exist, RWMS 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. RWMS 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:

Table 6–20 MMS_MANIFEST Table Format

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: OPEN: a new grouping of containers is started. SHIPPED: shipment is released.

Table 6–20 (Cont.) MMS_MANIFEST Table Format

Field Description	Template	Description
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 MMS writes records to the MMS_CONTAINER table. RWMS 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:

Table 6–21 MMS_CONTAINER Table

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

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

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

Triggers

ship_carton_trg

This new trigger notifies Kewill when the status of outbound cartons is updated to 'S'hipped.

create_sorter_instructions_trg

When the Kewill interface is enabled in RWMS, this trigger calls the socket interface package on the downloads from RWMS to Kewill.

Packages

label_info_received

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

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 is updated on the outbound carton record in RWMS.

ship_lane_upload

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

Tables

Cont_Ship_Label

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

Table 6–22 Cont_Ship_Label

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

Table 6–22 (Cont.) Cont_Ship_Label

label_type	VARCHAR2(4)	N	N	Type of label, for example 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

RWMS interfaces with Rapistan through socket interfaces. RWMS still generates directives based on logical destination IDs associated to locations setup in RWMS.

For RWMS generated directives (message sent to the control system), a trigger calls 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. RWMS determines the message type and call the appropriate procedure.

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

Triggers

create_sorter_instruction_trg

When the Rapistan interface is enabled in RWMS, the trigger calls the socket interface package on the message transfer from RWMS to Rapistan.

appt_rec_dir_trig

When the Rapistan interface is enabled in RWMS, the trigger writes receiving directive records for ASN appointments when the appointment status is updated to PEND.

cont_dest_trg.sql

Send the destination ID (carrier service) to Rapistan if the destination 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 inserts 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 inserts 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 inserts records into the sorter_intake table to process by the *process_diverts_a.sql* script.

pack_wave_release_upload.osp

Receive pack wave release confirmations from the control system. It calls the new *unit_sorter_directive.osp* stored procedure to send unit sortation information to the control system for the pack wave that releases by the control system.

unit_control_sorter_upload.osp

Receive unit sorter confirmations from the control system. It updates 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 RWMS, including inbound, outbound, and movements within the facility.

Table 6–23 Table for Container Transactions

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	RWMS container identifier

Table 6–23 (Cont.) Table for Container Transactions

Field Name	Field Type	Primary Key?	Req?	Description
logical_dest_id	X (4)	N	Y	Logical destination ID that relates to a location within RWMS.
divert_type	X (1)	N		
divert_ts	DATETIME	N	N	Date/time stamp
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.
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 RWMS for the third 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 is loaded onto trailers shipped that day. This information is then used in RWMS to determine the order in which outbound containers are picked so that they are 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 third party routing system.

route_data_upload.pkg

Procedures used to read and process route information returned from the third 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 third party routing system.

route_data_upload.sh and route_data_upload.sql

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

Download Transactions

RWMS sends a file containing estimated weights and cubes for the ship destinations by stock order number for which picking and shipping occurs 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:

Table 6–24 Format of Route Data File

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 destination ID in the stock order.
Total_weight	N(9)V(3)	N	N	Total weight for destination 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 third party routing package are placed in a directory named by the UNIX environment variable UPLOAD_DIR. The files are named with any set of numbers or characters as a prefix, but must end with the following character strings:

Table 6–25 File Format

File	File Naming Conventions
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 (for example, 001_route_dest.dat, 002_route_dest.dat). RWMS processes spool data from the routing files into corresponding upload tables, then each file processed is renamed with an extension of .baknnn, 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 processed at the same time.

Distro Route Upload

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

Table 6–26 Distribution Route Data File

Field Name	Field Type	Primary Key?	Req?	Description
Facility_id	X(2)	N	Y	Facility identifier
Transaction_ts	YYYYMMDDHH24MI	N	N	Date time stamp
Distro_nbr	X (10)	N	Y	Stock order identifier
Carrier_code	X(4)	N	Y	Carrier identifier
Service_code	X(6)	N	Y	Service identifier
Route	X(10)	N	Y	Route identifier
Ship_date	YYYYMMDDHH24MI	N	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
Distro_route_UDA5	X(10)	N	N	User defined attribute
Dest_id	X(10)	N	N	Store or DC dest id
Item_id	X(25)	N	N	Item id/code

Carrier Service Route Upload

The carrier service route data file contains carrier service route combinations are stored in the RWMS carrier_service_route table. Carrier code and route must be valid entries in the carrier and route tables in RWMS respectively.

Table 6–27 Carrier Service Route Upload

Field Name	Field Type	Primary Key?	Req?	Description
Facility_id	X(2)	Y	Y	Facility identifier
Transaction_ts	YYYYMMDDHH24MI	N	N	Date time stamp
Carrier_code	X(4)	Y	Y	Carrier identifier
Service_code	X(6)	Y	Y	Service identifier
Route	X(10)	Y	Y	Route identifier

Table 6–27 (Cont.) Carrier Service Route Upload

Field Name	Field Type	Primary Key?	Req?	Description
Location_id	X(12)	N	N	Optional location ID where containers for this carrier service route are staged for loading.

Route Date Upload

The route date upload file contains routes and the order in which they are picked and loaded for a given ship date. If a route coming from the third party routing system is not already defined in the RWMS route table, an entry in this table is created for it.

Table 6–28 Route Date Upload File

Field Name	Field Type	Primary Key?	Req?	Description
Facility_id	X(2)	Y	Y	Facility identifier
Transaction_ts	YYYYMMDDHH24MI	N	N	Date time stamp
Route	X(10)	Y	Y	Route identifier
Ship_date	YYYYMMDDHH24MI	Y	Y	Shipping date for which route sequence applies.
Route_sequence	N(3)	N	N	Sequence in which the route is loaded with other routes for the same day.

Route Dest Upload

The route destination upload file contains all of the ship destinations for a given route and the order in which they are loaded onto a trailer. The route and destination ID values must be valid in The RWMS route and ship_dest tables respectively.

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

Table 6–29 Route Destination Upload File

Field Name	Field Type	Primary Key?	Req?	Description
Facility_id	X(2)	Y	Y	Facility identifier
Transaction_ts	YYYYMMDDHH24MI	N	N	Date time stamp
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 destination ID are loaded with other ship destinations in the same route.

7

Appendix: Error Codes

RWMS maintains many API interfaces to support the downloading of specific data used in the application. These interfaces provide both generic and specific return values to the RIB as each message is consumed. The appendices contain the status codes and error messages returned by the API interfaces.

The first table contains the entire list of return values with the status code in numeric order. The second table contains the same list, with the error messages in alphabetic order.

Detail of Procedures

Each CONSUME procedure returns both a status code and an error message. When the message is consumed successfully, the status code returns an S, and the error message is NULL.

Appendix: Error Codes

All error codes are contained in the RWMSRIB_ERROR package.

Error Codes in Numerical Order

The following table includes error codes in numerical order.

Table 7–1 Error Codes in Numerical Order

Error Message	Status Code	Description
k_GENERAL_SUCCESS	S	A message was consumed.
k_GENERAL_API_FAILURE	E	A fatal error occurred in the procedure.
k_GENERAL_INVALID_FACILITY	102	Facility does not exist in the transshipment_setup table.
k_GENERAL_INVALID_ROUTING	103	An invalid destination was passed.
k_GENERAL_PRE_ADDL_PROCESSING	104	An error occurred in pre-additional processing.
k_GENERAL_FACILITY_NOT_FOUND	105	Facility does not exist in the transshipment_setup table.
k_VENDOR_DELETE_ERROR	200	Error while deleting a vendor.
k_VENDOR_OVERWRIT_STATUS_ ERROR	201	Errors while retrieving overwrite status values.

Table 7–1 Error Codes in Numerical Order

Error Message	Status Code	Description
k_VENDOR_MISSING_ERROR	202	Cannot delete or modify the record because it does not exist.
k_VENDOR_FOUND_ERROR	203	Cannot create record, record already exists.
k_VENDOR_ADDR_IND_UPDATE_ ERROR	204	An error occurred updating the primary address indicator.
k_VENDOR_PRIMARY_KEY_ERROR	205	Primary Key Violation.
k_VENDOR_ADDR_PRIM_KEY_ERROR	206	Primary Key Violation.
k_VENDOR_FK_ERROR	207	Referential Integrity error.
k_VENDOR_ADDR_TYPE_UPD_ERROR	208	Unable to update the Vendor Address description.
k_VENDOR_STATE_FK_ERROR	209	Referential Integrity error.
k_VENDOR_ADDR_MISSING_ERROR	210	Cannot delete or modify the record because it does not exist.
k_VENDOR_ADDR_FOUND_ERROR	211	Cannot create record, record already exists.
k_DIFF_ID_MISSING_ERROR	250	Cannot delete or modify the record because it does not exist.
k_DIFF_INSERT_ERROR	251	Diff ID is already being used as a Diff Group ID.
k_DIFF_PRIMARY_KEY_ERROR	252	Primary Key Violation.
k_DIFF_OTHER_ERROR	253	Unanticipated Differentiator error.
k_DIFF_ID_FOUND_ERROR	254	Cannot create record, record already exists.
k_DIFF_GRP_ID_MISSING_ERROR	300	Cannot delete or modify the record because it does not exist.
k_DIFF_GROUP_INSERT_ERROR	301	A Diff Group already exists.
k_DIFF_GROUP_PRIMARY_KEY_ERROR	302	Primary Key Violation.
k_DIFF_GROUP_OTHER_ERROR	303	An error occurred evaluating the Diff Group ID.
k_DIFF_DELETE_DIFF_GROUP_ERROR	304	An error occurred deleting from the Diff Group Detail table.
k_DIFF_GRP_ID_FOUND_ERROR	306	Cannot create record, record already exists.
k_DIFF_GRP_DET_MISSING_ERROR	350	Cannot delete or modify the record because it does not exist.
k_DIFF_GRP_DET_FOUND_ERROR	351	Cannot create record, record already exists.
k_DIFF_GROUP_DET_PRI_KEY_ERROR	352	Primary Key Violation.
k_DIFF_GROUP_DET_DG_FK_ERROR	353	Referential Integrity error.
k_DIFF_GROUP_DET_DI_FK_ERROR	354	Referential Integrity error.
k_ITEM_UPC_IND_UPDATE_ERROR	400	An error occurred in updating the primary UPC indicator in item_ upc.

Table 7–1 Error Codes in Numerical Order

Error Message	Status Code	Description
k_ITEM_SUPP_IND_UPDATE_ERROR	401	An error occurred updating the item_supplier or item_master tables.
k_ITEM_COUNTRY_IND_UPD_ERROR	402	Error updating item_supp_ country primary country indicator.
k_ITEM_OVERWRITE_STATUS_ERROR	403	Error when checking if a column can be overwritten.
k_APPT_DETAIL_POD_FK	404	Referential Integrity error.
k_ITEM_VENDOR_REF_ERROR	405	Foreign key error for vendor when creating an item.
k_ITEMM_PK_ERROR	406	Primary Key Violation.
k_ITEMM_UOM_FK_ERROR	407	Referential Integrity error.
k_ITEMM_VDR_FK_ERROR	408	Referential Integrity error.
k_ITEMM_CT_FK_ERROR	409	Referential Integrity error.
k_ITEMM_UPS_FK_ERROR	410	Referential Integrity error.
k_ITEMM_WC_FK_ERROR	411	Referential Integrity error.
k_ITEMUPC_PK_ERROR	412	Primary Key Violation.
k_ITEMUPC_IM_FK_ERROR	413	Referential Integrity error.
k_ITEMUPC_UNIQUE_ERROR	414	Unique error on item_upc UPC number when modifying a record.
k_ITEM_DIFF_PK_ERROR	415	Primary Key Violation.
k_ITEM_SUPP_PK_ERROR	416	Primary Key Violation.
k_ITEM_SUPP_IM_FK_ERROR	417	Referential Integrity error.
k_ITEM_SUPP_VDR_FK_ERROR	418	Referential Integrity error.
k_ISC_PK_ERROR	419	Primary Key Violation.
k_ISC_IS_FK_ERROR	420	Referential Integrity error.
k_ISCD_PK_ERROR	421	Primary Key Violation.
k_ISCD_ISC_FK_ERROR	422	Referential Integrity error.
k_ITEMA_PK_ERROR	423	Primary Key Violation.
k_ITEMA_ATTR_FK_ERROR	424	Referential Integrity error.
k_ITEMA_IM_FK_ERROR	425	Referential Integrity error.
k_PICKFROMLOC_ITEMM_FK_ERROR	426	Referential Integrity error.
k_PICKFROMLOC_LOC_FK_ERROR	427	Referential Integrity error.
k_STOCKA_ITEMM_FK_ERROR	428	Referential Integrity error.
k_ITEMCP_IM_FK_ERROR	429	Referential Integrity error.
k_ITEMCP_CURR_FK_ERROR	430	Referential Integrity error.
k_ITEM_SCD_OTHER_ERROR	431	Error while updating the dimensions of an item.
k_ITEM_CREATE_OTHER_ERROR	432	Unanticipated error while creating an item.

Table 7–1 Error Codes in Numerical Order

Error Message	Status Code	Description
k_ITEM_SUPP_MOD_OTHER_ERROR	433	Unanticipated error while modifying an item.
k_ITEM_SUPP_ITEM_CRE_OTH_ERROR	434	Unanticipated error while modifying the item.
k_ITEM_MOD_MSG_OTHER_ERROR	435	Unanticipated error when modifying an item, item differentiator.
k_ITEM_MISSING_ERROR	436	Cannot delete or modify the record because it does not exist.
k_ITEM_FOUND_ERROR	437	Error when item is created. Item already exists.
k_ITEM_SUPPLIER_MISSING_ERROR	438	Cannot delete or modify the record because it does not exist.
k_ITEM_SUPPLIER_FOUND_ERROR	439	Cannot create record, record already exists.
k_ITEM_SUPP_CTR_MISSING_ERROR	440	Cannot delete or modify the record because it does not exist.
k_ITEM_SUPP_CTR_FOUND_ERROR	441	Cannot create record, record already exists.
k_ITEM_SUPP_CTR_DIM_MISS_ERROR	442	Cannot delete or modify the record because it does not exist.
k_ITEM_SUPP_CTR_DIM_FND_ERROR	443	Cannot create record, record already exists.
k_ITEM_UPC_MISSING_ERROR	444	Cannot delete or modify the record because it does not exist.
k_ITEM_UPC_FOUND_ERROR	445	Cannot create record, record already exists.
k_ITEM_UDA_MISSING_ERROR	446	Cannot delete or modify the record because it does not exist.
k_ITEM_UDA_FOUND_ERROR	447	Cannot create record, record already exists.
k_ITEM_BOM_MISSING_ERROR	448	Cannot delete or modify the record because it does not exist.
k_ITEM_BOM_FOUND_ERROR	449	Cannot create record, record already exists.
k_ITEM_DIFF_FOUND_ERROR	451	Cannot create record, record already exists.
k_ITEM_ATTR_DEFAULTS_ERROR	452	Error when creating item and processing the attributes.
k_ITEMBOM_COMP_MSTR_SAME_ ERROR	453	MasterItemID is the same as the ComponentItemID.
k_ITEMBOM_CHECKING_ERROR	454	Error occurred checks to see if the MasterItemID is the same as the ComponentItemID.
k_ITEM_CLASS_ERROR	457	Error applying Item Download class.

Table 7–1 Error Codes in Numerical Order

Error Message	Status Code	Description
k_PO_VENDOR_REF_ERROR	503	Error on the vendor foreign key reference when creating a record.
k_PO_PK_ERROR	504	Primary Key Violation.
k_PO_DETAIL_PK_ERROR	505	Primary Key Violation.
k_PO_DETAIL_PO_ERROR	506	Error on the foreign key reference to PO from PO details when creating a PO detail.
k_PO_DETAIL_ITEM_ERROR	507	Error on the foreign key reference to item from PO details when creating a PO detail.
k_PO_RECEIPT_FK_ERROR	508	Referential Integrity error.
k_PO_CRE_MOD_OTHER_ERROR	509	Unanticipated error while creating a PO.
k_PO_FOUND_ERROR	512	Cannot create record, record already exists.
k_PO_MISSING_ERROR	513	Cannot delete or modify the record because it does not exist.
k_PO_DETAIL_FOUND_ERROR	514	Cannot create record, record already exists.
k_PO_DETAIL_MISSING_ERROR	515	Cannot delete or modify the record because it does not exist.
k_PO_DELIVER_DATE_ERROR	516	Error on the deliver date while creating a PO.
k_PO_CNTRY_VENDOR_ITEM_ERROR	517	PO record exists, cannot delete Vendor.
k_PO_OTHER_ERROR	521	Error validating PO detail required fields.
k_PO_STATUS_CANNOT_CLOSE_ERROR	524	Error trying to Close a PO where an unreceived appointment exists.
k_PO_STATUS_CHECKING_ERROR	525	Error checking appointment information.
k_ATTRIBUTE_FOUND_ERROR	554	Cannot create record, record already exists.
k_ATTRIBUTE_MISSING_ERROR	555	Cannot delete or modify the record because it does not exist.
k_ATTRIBUTE_OTHER_ERROR	556	Unanticipated error occurred while validating the key.
k_ATTRIBUTE_TYPE_PK_ERROR	559	Primary Key Violation.
k_ATTRIBUTE_TYPE_FK_ERROR	560	Referential Integrity error.
k_ATTRIBUTE_DETAIL_FOUND_ERROR	600	Attribute record already exists.
k_ATTRIBUTE_DETAIL_MISS_ERROR	601	Cannot delete or modify the record because it does not exist.
k_ATTRIBUTE_DETAIL_PK_ERROR	603	Primary Key Violation.
k_ATTRIBUTE_DET_ITEM_FK_ERROR	604	Referential Integrity error.
k_ATTRIBUTE_DET_WIP_FK_ERROR	605	Referential Integrity error.

Table 7–1 Error Codes in Numerical Order

Error Message	Status Code	Description
k_SHIP_OVERWRITE_STATUS_ERROR	650	Error retrieving ship destination overwrite status information.
k_SHIP_DEST_TYPE_NOT_VAL_ERROR	651	An error occurred checking the ship destination type.
k_SHIP_OWNING_DC_ERROR	652	Owning DC not found.
k_SHIP_DEST_FOUND_ERROR	653	Cannot create record, record already exists.
k_SHIP_DEST_MISSING_ERROR	654	Cannot delete or modify the record because it does not exist.
k_SHIP_DEST_REF_CONT_TYP_ERROR	655	Referential Integrity error.
k_SHIP_DEST_REF_UNIT_PCK_ERROR	656	Referential Integrity error.
k_SHIP_DEST_REF_ST_RDM_ERROR	657	Referential Integrity error.
k_SHIP_DEST_REF_PRIM_KEY_ERROR	658	Primary Key Violation.
k_SHIP_DEST_REF_DEF_CSR_ERROR	659	Referential Integrity error.
k_SHIP_DEST_REF_EXP_CSR_ERROR	660	Referential Integrity error.
k_SHIP_DEST_REF_CUR_CODE_ERROR	661	Referential Integrity error.
k_SHIP_DEST_ROUTE_ERROR	662	Referential Integrity error.
k_SHIP_INVALID_DEF_CSR_ERROR	663	Default_Service_Code, Default_ Carrier_Code and/or
Default_Route values are NULL.		
k_SHIP_INVALID_EXP_CSR_ERROR	664	Expedite_Service_Code, Expedite_ Carrier_Code and/or
Expedite_Route values are NULL.		
k_SHIP_MLD_FK_ERROR	665	Referential Integrity error.
k_SHIP_DEST_SEQ_NBR_ERROR	668	Error validating sequence number.
k_ASN_PK_ERROR	700	Primary Key Violation.
k_ASN_ITEM_PK_ERROR	701	Primary Key Violation.
k_ASN_ITEM_ASN_FK_ERROR	702	Referential Integrity error.
k_ASN_ITEM_SD_FK_ERROR	703	Referential Integrity error.
k_ASN_ITEM_POD_FK_ERROR	704	Referential Integrity error.
k_ASN_MISSING_ERROR	705	Cannot delete or modify the record because it does not exist.
k_ASN_FOUND_ERROR	706	Cannot create record, record already exists.
k_ASN_HDR_MOD_OTHER_ERROR	707	Unanticipated error while checking ASN type.
k_ASN_HDR_DEL_OTHER_ERROR	708	Unanticipated Error while checking appointment status for a delete.
k_ASN_ITEM_INVALID_QTY_ERROR	709	Change in unit quantity is less than zero.

Table 7–1 Error Codes in Numerical Order

Error Message	Status Code	Description
k_ASN_ITEM_APPT_ERROR	710	Error, cannot deleted an already appointed ASN.
k_ASN_CNTR_MOD_OTHER_ERROR	711	Unanticipated error while attempting to modify the container.
k_ASN_CNTR_ITEM_CRE_OTH_ERROR	712	Unanticipated error while setting the distributed unit quantity.
k_ASN_CNTR_ITEM_MOD_OTH_ERROR	713	Unanticipated error while setting the distributed unit quantity.
k_ASN_MOD_WRONG_ASN_ERROR	714	Error, ASN type is incorrect.
k_ASN_DEL_NOT_ALLOWED_ERROR	715	Error, status is not received.
k_ASN_PO_DTL_COUNTRY_ERROR	716	Error, country not found in item supplier country.
k_ASN_APPOINT_OPEN_ERROR	717	Error, status in not received or null.
k_ASN_DATA_TRACKING_ERROR	718	Error occurred in ASN processing procedure.
k_ASN_NOZERO_APPT_ASNITM_ ERROR	719	Error while deleting ASN item records.
k_VERIFY_ASN_PO_DATE_ERROR	720	Error, the deliver not after date is invalid.
k_VERIFY_ASN_PO_OTHER_ERROR	721	Error occurred verifying ASN Purchase Order.
k_VERIFY_ASN_ITEM_OTHER_ERROR	722	Unanticipated error occurred verifying ASN item record.
k_VERIFY_ASN_OTHER_ERROR	723	Unanticipated error occurred verifying the ASN record.
k_VERIFY_CONTAINER_OTHER_ERROR	724	Unanticipated error occurred verifying the container record.
k_VERIFY_CNTR_ITEM_OTHER_ERROR	725	Unanticipated error occurred verifying the container item record.
k_ASN_POST_ITEM_CRE_OTH_ERROR	726	Unanticipated error occurred in the post processing of ASN items.
k_ASN_POST_WIP_CODE_ERROR	727	Error occurred in the combine WIP codes procedure.
k_ASN_PO_INVALID_PO_ERROR	728	The PO is invalid for this ASN.
k_ASN_INVALID_STORE_DATE_ERROR	730	The In Store Date is invalid.
k_ASN_CONT_INVALID_ASN_ERROR	735	The ASN is incorrect for this container.
k_SA_IN_DISTRIBUTION_ERROR	829	Stock Allocation is in distribution.
k_SO_IN_DISTRIBUTION_ERROR	830	Stock Order has an Allocation in distribution.
k_STOCK_REQUIRED_PO_ERROR	831	Missing PO number.

Table 7–1 Error Codes in Numerical Order

Error Message	Status Code	Description
k_SA_NOT_FOUND_DELETE_ERROR	832	Cannot create the Stock Order Info Upload record, Stock Allocation record does not exist.
k_CONT_PK_ERROR	750	Primary Key Violation.
k_CONT_SD_FK_ERROR	751	Referential Integrity error.
k_CONT_LOC_FK1_ERROR	752	Referential Integrity error.
k_CONT_LOC_FK2_ERROR	753	Referential Integrity error.
k_CONT_CT_FK_ERROR	754	Referential Integrity error.
k_CONT_LOC_FK3_ERROR	755	Referential Integrity error.
k_CONT_ITEM_PK_ERROR	756	Primary Key Violation.
k_CONT_ITEM_CONT_FK_ERROR	757	Referential Integrity error.
k_CONT_ITEM_IM_FK_ERROR	758	Referential Integrity error.
k_CONT_MISSING_ERROR	759	Cannot delete or modify the record because it does not exist.
k_CONT_FOUND_ERROR	760	Cannot create record, record already exists.
k_CONT_ITEM_MISSING_ERROR	761	Cannot delete or modify the record because it does not exist.
k_CONT_ITEM_FOUND_ERROR	762	Cannot create record, record already exists.
k_STOCKO_PK_ERROR	800	Primary Key Violation.
k_STOCKO_STATE_FK1_ERROR	801	Referential Integrity error.
k_STOCKO_STATE_FK2_ERROR	802	Referential Integrity error.
k_STOCKO_PO_FK_ERROR	803	Referential Integrity error.
k_STOCK_NULL_SHIP_FIELD_ERROR	804	Error in ship field(s) when creating a stock order. One of the field(s) is null.
k_STOCK_PICK_DATE_ERROR	805	Error in the pick date when creating a stock order.
k_STOCK_MOD_CRE_OTHER_ERROR	806	Unanticipated error while modifying a stock order.
k_STOCK_ORDER_CRE_OTHER_ERROR	807	Unanticipated error while creating a stock order.
k_STOCKA_UP_REF1_ERROR	808	Referential Integrity error.
k_STOCKA_STOCKO_FK_ERROR	809	Referential Integrity error.
k_STOCKA_SHIPD_FK_ERROR	810	Referential Integrity error.
k_STOCK_INV_CARR_SERV_ERROR	811	Invalid carrier service when creating a stock order.
k_STOCK_INV_CARRIER_ERROR	812	Invalid carrier when creating a stock order.
k_STOCK_INV_CSR_ERROR	813	Invalid carrier service route when creating a stock order.

Table 7–1 Error Codes in Numerical Order

Error Message	Status Code	Description
k_STOCK_ORDER_MISSING_ERROR	814	Cannot delete or modify the record because it does not exist.
k_STOCK_ORDER_FOUND_ERROR	815	Cannot create record, record already exists.
k_STOCK_ALLOC_MISSING_ERROR	816	Cannot delete or modify the record because it does not exist.
k_STOCK_ALLOC_FOUND_ERROR	817	Cannot create record, record already exists.
k_STOCK_MODIFY_OTHER_ERROR	818	Unanticipated error while modifying a stock order.
k_STOCK_DISTRIBUTION_ERROR	819	Invalid stock distribution error while modifying a stock order.
k_STOCK_UNCARTONIZE_ERROR	820	Error during the uncartonize process when stock orders are modified. The error occurs during the call to the perform cartonization package.
k_STOCK_ALLOC_CRE_OTHER_ERROR	821	Unanticipated error in the standard UOM when creating a stock order.
k_ST_ALLOC_CRE_MOD_OTHER_ ERROR	822	Unanticipated error while creating a stock order.
k_ST_ALLOC_INSTORE_DATE_ERROR	823	In store date error while creating a stock order.
k_ST_ALLOC_MODIFY_OTHER_ERROR	824	Unanticipated error while modifying a stock allocation.
k_STOCK_DELETE_OTHER_ERROR	825	Unanticipated error while generating a SD stock order info status message when a stock allocation is deleted.
k_STOCK_CARTONIZATION_ERROR	826	Error in cartonization during the call to perform cartonization when a stock order is created.
k_SA_NOT_FOUND_DELETE_ERROR	832	Cannot create record, record already exists.
k_BOM_PK_ERROR	850	Primary Key Violation.
k_BOM_IM_FK1_ERROR	851	Referential Integrity error.
k_BOM_IM_FK2_ERROR	852	Referential Integrity error.
k_INBOUND_WO_PK_ERROR	900	Primary Key Violation.
k_INBOUND_WO_PO_FK_ERROR	901	Referential Integrity error.
k_INBOUND_WO_SD_FK_ERROR	902	Referential Integrity error.
k_INBOUND_WO_WIP_FK_ERROR	903	Referential Integrity error.
k_INBOUND_WO_FOUND_ERROR	904	Cannot create record, record already exists.
k_INBOUND_WO_MISSING_ERROR	905	Cannot delete or modify the record because it does not exist.

Table 7–1 Error Codes in Numerical Order

Error Message	Status Code	Description
k_INBOUND_DEST_XML_OTHER_ ERROR	906	Unanticipated
k_PENDING_RETURN_PK_ERROR	950	Primary Key Violation.
k_PENDING_RET_DET_PK_ERROR	952	Primary Key Violation.
k_PENDRET_DET_FK_PR_ERROR	953	Referential Integrity error.
k_PENDING_RETURN_FOUND_ERROR	954	Cannot create record, record already exists.
k_PENDING_RETURN_MISSING_ERROR	955	Cannot delete or modify the record because it does not exist.
k_PENDRET_DEST_XML_OTHER_ ERROR	958	Unanticipated
k_PENDRET_ITEM_FK_ERROR	959	Referential Integrity error.
k_OUTBOUND_DEST_XML_OTH_ ERROR	1001	Unanticipated
k_OBWO_STOCK_ALLOC_FK_ERROR	1002	Referential Integrity error.
k_OBWO_WIP_FK_ERROR	1003	Referential Integrity error.
k_OUTBOUND_WO_MISSING_ERROR	1004	Cannot delete or modify the record because it does not exist.
k_OUTBOUND_WO_FOUND_ERROR	1005	Cannot create record, record already exists.
k_SKU_ADD_OTHER_ERROR	1100	Unanticipated
k_SKU_MISSING_ERROR	1101	Cannot delete or modify the record because it does not exist.
k_SKU_FOUND_ERROR	1102	Task_Queue record already exists.
k_TASK_QUEUE_ACT_CODE_FK_ERROR	1103	Referential Integrity error.
k_TASK_QUEUE_WAVE_FK_ERROR	1104	Referential Integrity error.
k_TASK_QUEUE_ITEM_FK_ERROR	1105	Referential Integrity error.
k_STATE_NOT_FOUND_ERROR	10501	Cannot create record, record already exists.
k_COUNTRY_NOT_FOUND_ERROR	10502	Cannot create record, record already exists.
k_CURRENCY_NOT_FOUND_ERROR	10503	Cannot create record, record already exists.
k_CONTAINER_NOT_FOUND_ERROR	10504	Cannot create record, record already exists.
k_ITEM_UPDATE_COLUMNS_ERROR	10505	Cannot create record, record already exists.
k_PICK_DIR_SO_FK_ERROR	10506	Referential Integrity error.
k_COMP_TICK_STOCKO_ERROR	10507	Referential Integrity error.
k_KIT_BUILD_REF3_ERROR	10508	Referential Integrity error.
k_SORTED_ALLOC_REF1_ERROR	10510	Referential Integrity error.
k_DISTRO_IS_REF1_ERROR	10511	Referential Integrity error.

Table 7–1 Error Codes in Numerical Order

Error Message	Status Code	Description
k_STOCKA_CID_REF1	10512	Referential Integrity error.

Error Codes in Alphabetical Order

The following table includes error codes in alphabetical order.

Table 7–2 Error Codes in Alphabetical Order

Error Message	Status Code	Description
k_APPT_DETAIL_POD_FK	404	Referential Integrity error.
k_ASN_APPOINT_OPEN_ERROR	717	Error, status in not received or null.
k_ASN_CNTR_ITEM_CRE_OTH_ERROR	712	Unanticipated error while setting the distributed unit quantity.
k_ASN_CNTR_ITEM_MOD_OTH_ERROR	713	Unanticipated error while setting the distributed unit quantity.
k_ASN_CNTR_MOD_OTHER_ERROR	711	Unanticipated error while attempting to modify the container.
k_ASN_CONT_INVALID_ASN_ERROR	735	The ASN is incorrect for this container.
k_ASN_DATA_TRACKING_ERROR	718	Error occurred in ASN processing procedure.
k_ASN_DEL_NOT_ALLOWED_ERROR	715	Error, status is not received.
k_ASN_FOUND_ERROR	706	Cannot create record, record already exists.
k_ASN_HDR_DEL_OTHER_ERROR	708	Unanticipated Error while checking appointment status for a delete.
k_ASN_HDR_MOD_OTHER_ERROR	707	Unanticipated error while checking ASN type.
k_ASN_INVALID_STORE_DATE_ERROR	730	The In Store Date is invalid.
k_ASN_ITEM_APPT_ERROR	710	Error, cannot deleted an already appointed ASN.
k_ASN_ITEM_ASN_FK_ERROR	702	Referential Integrity error.
k_ASN_ITEM_INVALID_QTY_ERROR	709	Change in unit quantity is less than zero.
k_ASN_ITEM_PK_ERROR	701	Primary Key Violation.
k_ASN_ITEM_POD_FK_ERROR	704	Referential Integrity error.
k_ASN_ITEM_SD_FK_ERROR	703	Referential Integrity error.
k_ASN_MISSING_ERROR	705	Cannot delete or modify the record because it does not exist.
k_ASN_MOD_WRONG_ASN_ERROR	714	Error, ASN type is incorrect.
k_ASN_NOZERO_APPT_ASNITM_ ERROR	719	Error while deleting ASN item records.
k_ASN_PK_ERROR	700	Primary Key Violation.

Table 7–2 Error Codes in Alphabetical Order

Error Message	Status Code	Description
k_ASN_PO_DTL_COUNTRY_ERROR	716	Error, country not found in item supplier country.
k_ASN_PO_INVALID_PO_ERROR	728	The PO is invalid for this ASN.
k_ASN_POST_ITEM_CRE_OTH_ERROR	726	Unanticipated error occurred in the post processing of ASN items.
k_ASN_POST_WIP_CODE_ERROR	727	Error occurred in the combine WIP codes procedure.
k_ATTRIBUTE_DET_ITEM_FK_ERROR	604	Referential Integrity error.
k_ATTRIBUTE_DET_WIP_FK_ERROR	605	Referential Integrity error.
k_ATTRIBUTE_DETAIL_FOUND_ERROR	600	Attribute record already exists.
k_ATTRIBUTE_DETAIL_MISS_ERROR	601	Cannot delete or modify the record because it does not exist.
k_ATTRIBUTE_DETAIL_PK_ERROR	603	Primary Key Violation.
k_ATTRIBUTE_FOUND_ERROR	554	Cannot create record, record already exists.
k_ATTRIBUTE_MISSING_ERROR	555	Cannot delete or modify the record because it does not exist.
k_ATTRIBUTE_OTHER_ERROR	556	Unanticipated error occurred while validating the key.
k_ATTRIBUTE_TYPE_FK_ERROR	560	Referential Integrity error.
k_ATTRIBUTE_TYPE_PK_ERROR	559	Primary Key Violation.
k_BOM_IM_FK1_ERROR	851	Referential Integrity error.
k_BOM_IM_FK2_ERROR	852	Referential Integrity error.
k_BOM_PK_ERROR	850	Primary Key Violation.
k_CONT_CT_FK_ERROR	754	Referential Integrity error.
k_CONT_FOUND_ERROR	760	Cannot create record, record already exists.
k_CONT_ITEM_CONT_FK_ERROR	757	Referential Integrity error.
k_CONT_ITEM_FOUND_ERROR	762	Cannot create record, record already exists.
k_CONT_ITEM_IM_FK_ERROR	758	Referential Integrity error.
k_CONT_ITEM_MISSING_ERROR	761	Cannot delete or modify the record because it does not exist.
k_CONT_ITEM_PK_ERROR	756	Primary Key Violation.
k_CONT_LOC_FK1_ERROR	752	Referential Integrity error.
k_CONT_LOC_FK2_ERROR	753	Referential Integrity error.
k_CONT_LOC_FK3_ERROR	755	Referential Integrity error.
k_CONT_MISSING_ERROR	759	Cannot delete or modify the record because it does not exist.
k_CONT_PK_ERROR	750	Primary Key Violation.
k_CONT_SD_FK_ERROR	751	Referential Integrity error.

Table 7–2 Error Codes in Alphabetical Order

Error Message	Status Code	Description
k_DIFF_DELETE_DIFF_GROUP_ERROR	304	An error occurred deleting from the Diff Group Detail table.
k_DIFF_GROUP_DET_DG_FK_ERROR	353	Referential Integrity error.
k_DIFF_GROUP_DET_DI_FK_ERROR	354	Referential Integrity error.
k_DIFF_GROUP_DET_PRI_KEY_ERROR	352	Primary Key Violation.
k_DIFF_GROUP_INSERT_ERROR	301	A Diff Group already exists.
k_DIFF_GROUP_OTHER_ERROR	303	An error occurred evaluating the Diff Group ID.
k_DIFF_GROUP_PRIMARY_KEY_ERROR	302	Primary Key Violation.
k_GENERAL_SUCCESS	S	A message was consumed.
k_DIFF_GRP_DET_FOUND_ERROR	351	Cannot create record, record already exists.
k_DIFF_GRP_DET_MISSING_ERROR	350	Cannot delete or modify the record because it does not exist.
k_DIFF_GRP_ID_FOUND_ERROR	306	Cannot create record, record already exists.
k_DIFF_GRP_ID_MISSING_ERROR	300	Cannot delete or modify the record because it does not exist.
k_DIFF_ID_FOUND_ERROR	254	Cannot create record, record already exists.
k_DIFF_ID_MISSING_ERROR	250	Cannot delete or modify the record because it does not exist.
k_DIFF_INSERT_ERROR	251	Diff ID is already being used as a Diff Group ID.
k_DIFF_OTHER_ERROR	253	Unanticipated Differentiator error.
k_DIFF_PRIMARY_KEY_ERROR	252	Primary Key Violation.
k_GENERAL_API_FAILURE	E	A fatal error occurred in the procedure.
k_GENERAL_FACILITY_NOT_FOUND	105	Facility does not exist in the transshipment_setup table.
k_GENERAL_INVALID_FACILITY	102	Facility does not exist in the transshipment_setup table.
k_GENERAL_INVALID_ROUTING	103	An invalid destination was passed.
k_GENERAL_PRE_ADDL_PROCESSING	104	An error occurred in pre-additional processing.
k_INBOUND_DEST_XML_OTHER_ ERROR	906	Unanticipated
k_INBOUND_WO_FOUND_ERROR	904	Cannot create record, record already exists.
k_INBOUND_WO_MISSING_ERROR	905	Cannot delete or modify the record because it does not exist.
k_INBOUND_WO_PK_ERROR	900	Primary Key Violation.
k_INBOUND_WO_PO_FK_ERROR	901	Referential Integrity error.

Table 7–2 Error Codes in Alphabetical Order

Error Message	Status Code	Description
k_INBOUND_WO_SD_FK_ERROR	902	Referential Integrity error.
k_INBOUND_WO_WIP_FK_ERROR	903	Referential Integrity error.
k_ISC_IS_FK_ERROR	420	Referential Integrity error.
k_ISC_PK_ERROR	419	Primary Key Violation.
k_ISCD_ISC_FK_ERROR	422	Referential Integrity error.
k_ISCD_PK_ERROR	421	Primary Key Violation.
k_ITEM_ATTR_DEFAULTS_ERROR	452	Error when creating item and processing the attributes.
k_ITEM_BOM_FOUND_ERROR	449	Cannot create record, record already exists.
k_ITEM_BOM_MISSING_ERROR	448	Cannot delete or modify the record because it does not exist.
k_ITEM_CLASS_ERROR	457	Error applying Item Download class.
k_ITEM_COUNTRY_IND_UPD_ERROR	402	Error updating item_supp_ country primary country indicator.
k_ITEM_CREATE_OTHER_ERROR	432	Unanticipated error while creating an item.
k_ITEM_DIFF_FOUND_ERROR	451	Cannot create record, record already exists.
k_ITEM_DIFF_PK_ERROR	415	Primary Key Violation.
k_ITEM_FOUND_ERROR	437	Error when item is created. Item already exists.
k_ITEM_MISSING_ERROR	436	Cannot delete or modify the record because it does not exist.
k_ITEM_MOD_MSG_OTHER_ERROR	435	Unanticipated error when modifying an item, item differentiator.
k_ITEM_OVERWRITE_STATUS_ERROR	403	Error when checking if a column can be overwritten.
k_ITEM_SCD_OTHER_ERROR	431	Error while updating the dimensions of an item.
k_ITEM_SUPP_CTR_DIM_FND_ERROR	443	Cannot create record, record already exists.
k_ITEM_SUPP_CTR_DIM_MISS_ERROR	442	Cannot delete or modify the record because it does not exist.
k_ITEM_SUPP_CTR_FOUND_ERROR	441	Cannot create record, record already exists.
k_ITEM_SUPP_CTR_MISSING_ERROR	440	Cannot delete or modify the record because it does not exist.
k_ITEM_SUPP_IM_FK_ERROR	417	Referential Integrity error.
k_ITEM_SUPP_IND_UPDATE_ERROR	401	An error occurred updating the item_supplier or item_master tables.

Table 7–2 Error Codes in Alphabetical Order

Error Message	Status Code	Description
k_ITEM_SUPP_ITEM_CRE_OTH_ERROR	434	Unanticipated error while modifying the item.
k_ITEM_SUPP_MOD_OTHER_ERROR	433	Unanticipated error while modifying an item.
k_ITEM_SUPP_PK_ERROR	416	Primary Key Violation.
k_ITEM_SUPP_VDR_FK_ERROR	418	Referential Integrity error.
k_ITEM_SUPPLIER_FOUND_ERROR	439	Cannot create record, record already exists.
k_ITEM_SUPPLIER_MISSING_ERROR	438	Cannot delete or modify the record because it does not exist.
k_ITEM_UDA_FOUND_ERROR	447	Cannot create record, record already exists.
k_ITEM_UDA_MISSING_ERROR	446	Cannot delete or modify the record because it does not exist.
k_ITEM_UPC_FOUND_ERROR	445	Cannot create record, record already exists.
k_ITEM_UPC_IND_UPDATE_ERROR	400	An error occurred in updating the primary UPC indicator in item_ upc.
k_ITEM_UPC_MISSING_ERROR	444	Cannot delete or modify the record because it does not exist.
k_ITEM_VENDOR_REF_ERROR	405	Foreign key error for vendor when creating an item.
k_ITEMA_ATTR_FK_ERROR	424	Referential Integrity error.
k_ITEMA_IM_FK_ERROR	425	Referential Integrity error.
k_ITEMA_PK_ERROR	423	Primary Key Violation.
k_ITEMBOM_CHECKING_ERROR	454	Error occurred checks to see if the MasterItemID is the same as the ComponentItemID.
k_ITEMBOM_COMP_MSTR_SAME_ ERROR	453	MasterItemID is the same as the ComponentItemID.
k_ITEMCP_CURR_FK_ERROR	430	Referential Integrity error.
k_ITEMCP_IM_FK_ERROR	429	Referential Integrity error.
k_ITEMM_CT_FK_ERROR	409	Referential Integrity error.
k_ITEMM_PK_ERROR	406	Primary Key Violation.
k_ITEMM_UOM_FK_ERROR	407	Referential Integrity error.
k_ITEMM_UPS_FK_ERROR	410	Referential Integrity error.
k_ITEMM_VDR_FK_ERROR	408	Referential Integrity error.
k_ITEMM_WC_FK_ERROR	411	Referential Integrity error.
k_ITEMUPC_IM_FK_ERROR	413	Referential Integrity error.
k_ITEMUPC_PK_ERROR	412	Primary Key Violation.
k_ITEMUPC_UNIQUE_ERROR	414	Unique error on item_upc UPC number when modifying a record.

Table 7–2 Error Codes in Alphabetical Order

Error Message	Status Code	Description
k_OBWO_STOCK_ALLOC_FK_ERROR	1002	Referential Integrity error.
k_OUTBOUND_DEST_XML_OTH_ ERROR	1001	Unanticipated
k_PENDING_RET_DET_PK_ERROR	952	Primary Key Violation.
k_PENDING_RETURN_FOUND_ERROR	954	Cannot create record, record already exists.
k_PENDING_RETURN_MISSING_ERROR	955	Cannot delete or modify the record because it does not exist.
k_PENDING_RETURN_PK_ERROR	950	Primary Key Violation.
k_PENDRET_DEST_XML_OTHER_ ERROR	958	Unanticipated
k_PENDRET_DET_FK_PR_ERROR	953	Referential Integrity error.
k_PENDRET_ITEM_FK_ERROR	959	Referential Integrity error.
k_PICKFROMLOC_ITEMM_FK_ERROR	426	Referential Integrity error.
k_PICKFROMLOC_LOC_FK_ERROR	427	Referential Integrity error.
k_PO_CNTRY_VENDOR_ITEM_ERROR	517	PO record exists, cannot delete Vendor.
k_PO_CRE_MOD_OTHER_ERROR	509	Unanticipated error while creating a PO.
k_PO_DELIVER_DATE_ERROR	516	Error on the deliver date while creating a PO.
k_PO_DETAIL_FOUND_ERROR	514	Cannot create record, record already exists.
k_PO_DETAIL_ITEM_ERROR	507	Error on the foreign key reference to item from PO details when creating a PO detail.
k_PO_DETAIL_MISSING_ERROR	515	Cannot delete or modify the record because it does not exist.
k_PO_DETAIL_PK_ERROR	505	Primary Key Violation.
k_PO_DETAIL_PO_ERROR	506	Error on the foreign key reference to PO from PO details when creating a PO detail.
k_PO_FOUND_ERROR	512	Cannot create record, record already exists.
k_PO_MISSING_ERROR	513	Cannot delete or modify the record because it does not exist.
k_PO_OTHER_ERROR	521	Error validating PO detail required fields.
k_PO_PK_ERROR	504	Primary Key Violation.
k_PO_RECEIPT_FK_ERROR	508	Referential Integrity error.
k_PO_STATUS_CANNOT_CLOSE_ERROR	524	Error trying to Close a PO where an unreceived appointment exists.
k_PO_STATUS_CHECKING_ERROR	525	Error checking appointment information.

Table 7–2 Error Codes in Alphabetical Order

Error Message	Status Code	Description
k_PO_VENDOR_REF_ERROR	503	Error on the vendor foreign key reference when creating a record.
k_SA_IN_DISTRIBUTION_ERROR	829	Stock Allocation is in distribution.
k_SA_NOT_FOUND_DELETE_ERROR	832	Cannot create the Stock Order Info Upload record, Stock Allocation record does not exist.
k_SA_NOT_FOUND_DELETE_ERROR	832	Cannot create record, record already exists.
k_SHIP_DEST_FOUND_ERROR	653	Cannot create record, record already exists.
k_SHIP_DEST_MISSING_ERROR	654	Cannot delete or modify the record because it does not exist.
k_SHIP_DEST_REF_CONT_TYP_ERROR	655	Referential Integrity error.
k_SHIP_DEST_REF_CUR_CODE_ERROR	661	Referential Integrity error.
k_SHIP_DEST_REF_DEF_CSR_ERROR	659	Referential Integrity error.
k_SHIP_DEST_REF_EXP_CSR_ERROR	660	Referential Integrity error.
k_SHIP_DEST_REF_PRIM_KEY_ERROR	658	Primary Key Violation.
k_SHIP_DEST_REF_ST_RDM_ERROR	657	Referential Integrity error.
k_SHIP_DEST_REF_UNIT_PCK_ERROR	656	Referential Integrity error.
k_SHIP_DEST_ROUTE_ERROR	662	Referential Integrity error.
k_SHIP_DEST_SEQ_NBR_ERROR	668	Error validating sequence number.
k_SHIP_DEST_TYPE_NOT_VAL_ERROR	651	An error occurred checking the ship destination type.
k_SHIP_INVALID_DEF_CSR_ERROR	663	Default_Service_Code, Default_ Carrier_Code and/or Default_ Route values are NULL.
k_SHIP_INVALID_EXP_CSR_ERROR	664	Expedite_Service_Code, Expedite_ Carrier_Code and/or Expedite_ Route values are NULL.
k_SHIP_MLD_FK_ERROR	665	Referential Integrity error.
k_SHIP_OVERWRITE_STATUS_ERROR	650	Error retrieving ship destination overwrite status information.
k_SHIP_OWNING_DC_ERROR	652	Owning DC not found.
k_SO_IN_DISTRIBUTION_ERROR	830	Stock Order has an Allocation in distribution.
k_ST_ALLOC_CRE_MOD_OTHER_ ERROR	822	Unanticipated error while creating a stock order.
k_ST_ALLOC_INSTORE_DATE_ERROR	823	In store date error while creating a stock order.
k_ST_ALLOC_MODIFY_OTHER_ERROR	824	Unanticipated error while modifying a stock allocation.
k_STOCK_ALLOC_CRE_OTHER_ERROR	821	Unanticipated error in the standard UOM when creating a stock order.

Table 7–2 Error Codes in Alphabetical Order

Error Message	Status Code	Description
k_STOCK_ALLOC_FOUND_ERROR	817	Cannot create record, record already exists.
k_STOCK_ALLOC_MISSING_ERROR	816	Cannot delete or modify the record because it does not exist.
k_STOCK_CARTONIZATION_ERROR	826	Error in cartonization during the call to perform cartonization when a stock order is created.
k_STOCK_DELETE_OTHER_ERROR	825	Unanticipated error while generating a SD stock order info status message when a stock allocation is deleted.
k_STOCK_DISTRIBUTION_ERROR	819	Invalid stock distribution error while modifying a stock order.
k_STOCK_INV_CARR_SERV_ERROR	811	Invalid carrier service when creating a stock order.
k_STOCK_INV_CARRIER_ERROR	812	Invalid carrier when creating a stock order.
k_STOCK_INV_CSR_ERROR	813	Invalid carrier service route when creating a stock order.
k_STOCK_MOD_CRE_OTHER_ERROR	806	Unanticipated error while modifying a stock order.
k_STOCK_MODIFY_OTHER_ERROR	818	Unanticipated error while modifying a stock order.
k_STOCK_NULL_SHIP_FIELD_ERROR	804	Error in ship field(s) when creating a stock order. One of the field(s) is null.
k_STOCK_ORDER_CRE_OTHER_ERROR	807	Unanticipated error while creating a stock order.
k_STOCK_ORDER_FOUND_ERROR	815	Cannot create record, record already exists.
k_STOCK_ORDER_MISSING_ERROR	814	Cannot delete or modify the record because it does not exist.
k_STOCK_PICK_DATE_ERROR	805	Error in the pick date when creating a stock order.
k_STOCK_REQUIRED_PO_ERROR	831	Missing PO number.
k_STOCK_UNCARTONIZE_ERROR	820	Error during the uncartonize process when stock orders are modified. The error occurs during the call to the perform cartonization package.
k_STOCKA_ITEMM_FK_ERROR	428	Referential Integrity error.
k_STOCKA_SHIPD_FK_ERROR	810	Referential Integrity error.
k_STOCKA_STOCKO_FK_ERROR	809	Referential Integrity error.
k_STOCKA_UP_REF1_ERROR	808	Referential Integrity error.
k_STOCKO_PK_ERROR	800	Primary Key Violation.
k_STOCKO_PO_FK_ERROR	803	Referential Integrity error.

Table 7–2 Error Codes in Alphabetical Order

Error Message	Status Code	Description
k_STOCKO_STATE_FK1_ERROR	801	Referential Integrity error.
k_STOCKO_STATE_FK2_ERROR	802	Referential Integrity error.
k_VENDOR_ADDR_FOUND_ERROR	211	Cannot create record, record already exists.
k_VENDOR_ADDR_IND_UPDATE_ ERROR	204	An error occurred updating the primary address indicator.
k_VENDOR_ADDR_MISSING_ERROR	210	Cannot delete or modify the record because it does not exist.
k_VENDOR_ADDR_PRIM_KEY_ERROR	206	Primary Key Violation.
k_VENDOR_ADDR_TYPE_UPD_ERROR	208	Unable to update the Vendor Address description.
k_VENDOR_DELETE_ERROR	200	Error while deleting a vendor.
k_VENDOR_FK_ERROR	207	Referential Integrity error.
k_VENDOR_FOUND_ERROR	203	Cannot create record, record already exists.
k_VENDOR_MISSING_ERROR	202	Cannot delete or modify the record because it does not exist.
k_VENDOR_OVERWRIT_STATUS_ ERROR	201	Error while retrieving overwrite status values.
k_VENDOR_PRIMARY_KEY_ERROR	205	Primary Key Violation.
k_VENDOR_STATE_FK_ERROR	209	Referential Integrity error.
k_VERIFY_ASN_ITEM_OTHER_ERROR	722	Unanticipated error occurred verifying ASN item record.
k_VERIFY_ASN_OTHER_ERROR	723	Unanticipated error occurred verifying the ASN record.
k_VERIFY_ASN_PO_DATE_ERROR	720	Error, the deliver not after date is invalid.
k_VERIFY_ASN_PO_OTHER_ERROR	721	Error occurred verifying ASN Purchase Order.
k_VERIFY_CNTR_ITEM_OTHER_ERROR	725	Unanticipated error occurred verifying the container item record.
k_VERIFY_CONTAINER_OTHER_ERROR	724	Unanticipated error occurred verifying the container record.

Index

Q

R

S

T

U

V

W