Retek® Merchandising System™ 10.1.12

Operations Guide Addendum

Corporate Headquarters:

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

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

Fax:
+1 612 587 5100

European Headquarters:

Retek
110 Wigmore Street
London
W1U 3RW
United Kingdom

Switchboard:
+44 (0)20 7563 4600

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

Fax:
+44 (0)20 7563 46 10

Printed in the United States of America.

Customer Support

Customer Support hours

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

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

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

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

Australia	+1 800 555 923 (AU-Telstra) or +1 800 000 562 (AU-Optus)
France	0800 90 91 66
Hong Kong	800 96 4262
Korea	00 308 13 1342
United Kingdom	0800 917 2863
United States	+1 800 61 RETEK or 800 617 3835
Mail	Retek Customer Support Retek on the Mall 950 Nicollet Mall Minneapolis, MN 55403

When contacting Customer Support, please provide:

- Product version and program/module name.
- Functional and technical description of the problem (include business impact).
- Detailed step-by-step instructions to recreate.
- Exact error message received.
- Screen shots of each step you take.

Contents

Chapter 1 – Introduction

This addendum to the RMS 10 Operations Guide presents changes that have resulted from work completed during RMS 10.1.12 development and customer support.

Chapter 2 – RMS-Advanced Inventory Planning (AIP) interface RETL batch

Overview of modifications

Due to enhancements in the next base AIP release, RMS has been updated to better provide integration-related information that AIP needs to function. The modifications can be divided into the areas that are described in this section. Note that this release does not include any modifications surrounding the communication methods (such as the RIB) that exist from AIP to RMS.

Note: See AIP documentation for all modifications that have been made to RETL load and to transform programs.

Multi-level distribution support (warehouse changes)

Multi-tier warehouses in RMS

Currently, AIP has the concept of a primary consolidation center (2nd tier warehouse or national distribution center) and a regional distribution center (1st tier warehouse). Goods are ordered to the national distribution center and then distributed to the regional warehouses.

RMS was modified to remove the warehouse tier attribute for virtual warehouses. Going forward, the link between 1st tier and 2nd tier warehouses will be maintained in AIP. RMS no longer differentiates between 1st tier and 2nd tier distribution centers but maintains them as normal stock holding warehouses. Along with the removal of the warehouse tier attribute, the warehouse-related RETL extract has been modified to *not* differentiate between 1st and 2nd tier warehouses.

Warehouse type

RMS was modified to capture the warehouse type attribute for virtual warehouses created in RMS. The warehouse type attribute drives *no* functionality within RMS. If the AIP indicator is set to 'Y', validation ensures that a warehouse type is specified for all virtual warehouses. The warehouse type field is visible but is neither editable nor required when the AIP indicator is set to 'N'. The warehouse RETL extract has also been modified to create a new output file for the new warehouse type attribute when extracting warehouse information from RMS.

Informal and formal packs

Informal packs

Previously, the integration between RMS and AIP placed inventory at a location at the primary supplier/primary country/case pack size level. A modification was made to the RMS item structure so that the user can specify which case pack size is the primary case pack size for communicating inventory to AIP. The user has control over what is the smallest shipping unit for AIP.

For example, if item A is created with an inner size of 3 and a case size of 6, but the user identified that an each is the primary case size, inventory is sent to AIP in the manner portrayed in the table below:

Item	Pack size	Quantity
A	1	18
A	3	0
A	6	0

AIPs reconciliation processes provide the ability to send a store four units of item A if the need specifies, because 1 is a valid shipping pack size.

Taking the same example as above but for a user who selects inner as the primary case size, the inventory is sent to AIP in the manner portrayed in the table below:

Item	Pack size	Quantity
A	1	0
A	3	18
A	6	0

This processing forces AIP's reconciliation logic to always round to inners. AIP's base logic, however, allows it to send out eaches, independent of the pack size of 1 that is being sent to it. Any quantity that is not equally divisible by the selected pack size is communicated to AIP in the pack size 1 item.

RMS includes the option of specifying the pack/case size for an item. The pack/case size no longer automatically is defaulted to the pack/case size of the primary supplier. The user may choose between the option of using the 'each', 'inner', 'case', or 'pallet' as the primary case size when sending information to AIP. The option is available when creating or editing an item that uses informal packs. Pack items do not have this option. Instead, when simple packs are sent to AIP, the component item's pack/case size is sent for the pack.

In the vast majority of cases, the primary case size can be set at the parent (or grandparent in the case of 3-level items). RMS allows level 1 (in the case of simple items), level 2, and level 3 items (in the case of item-child situations), each to have their own primary case size. This allowance is made because some of the basic differentiators for items alter the orders and ordering techniques that apply. The case size that is selected at the parent level defaults to all children or grandchildren that exist below it using item default logic.

Formal packs

Instead of using the informal pack sizes for the primary country of an item/supplier, users can choose to use formal packs. Formal packs provide the ability to have multiple pack configurations per item and the ability to track inventory by each of these pack configurations. This processing requires a separate item (simple pack) in RMS to represent each item configuration because RMS tracks inventory at the item level.

If the item is a pack item, both the formal/informal indicator and the primary case size dropdown are disabled and set to NULL. When simple packs are sent to AIP, the extracts use the component item's formal/informal and primary case size to send information for the simple pack.

Inventory communicated to AIP in terms of primary case size

Because AIP views all inventory at the informal or formal case size, all AIP inventory extracts from RMS were changed to send inventory in terms of the primary case size. Item supplier country case sizes are now sent for items that have an informal pack setting. For formal packs, the case size is the component quantity in the pack. The extraction of the current stock on hand for AIP was changed so that the stock is sent in terms of the primary case size. Finally, all future on-order, transfer and allocation inventory is now being communicated to AIP in terms of the primary case size.

Only forecastable items sent to AIP

Previously, RMS published all items to AIP regardless of whether or not they were forecastable. This process was modified so that RMS now only sends forecasted items to AIP. Therefore, all inventory and future available inventory downloads for AIP are restricted to forecastable items only. If the item to be extracted is a simple pack, its component must be forecastable.

Visibility to warehouse-to-warehouse allocation stock movements

The existing future delivery of allocations and 'allocations in well' extracts are currently limited to only inbound allocations into stores. Because retailers may have systems in which warehouse-to-warehouse allocations are possible, the extracts have been modified to include any inbound allocations into a warehouse.

Program level modification descriptions

Rmse_wh.ksh

The warehouse RETL extract script from RMS currently produces one output for warehouses that are tier 1 (whse1.txt), and a separate output for warehouses that are tier 2 (whse2.txt). Because RMS no longer has tier information, the extract does not produce tier-specific outputs. The combined output is called rmse_aip_wh.txt, and it should not be filtered by tier.

Additionally, the warehouse RETL extract script has been modified to create a new output file called rmse_aip_wh_type.txt. This output file contains warehouses along with their associated warehouse type attribute. Rmse_wh.ksh extracts the new Warehouse Type attribute when extracting warehouse information from RMS.

Rmse_wh_cur_inventory.ksh

The warehouse current inventory RETL extract produces one output for tier 1 warehouses, and a separate output for tier 2 warehouses. Because RMS no longer has tier information, the extract does not produce tier-specific outputs. The inventory cursors no longer selects tier from the WH table. In addition, the wh_all_inv_non_zero.v dataset is not split into a file for tier 1 and 2. This is the final output.

Rmse_wh_unavail_inventory.ksh

The warehouse unavailable inventory extract currently only finds unavailable inventory for tier 1 warehouses. In both cursors that find unavailable inventory, the condition “and wh.VWH_TIER = 1” has been removed so that all warehouses, regardless of tier, have their unavailable inventory sent to AIP.

Rmse_item_supp_country.ksh

This RETL extract has been altered to include all unique pack sizes for an item. In addition, this RETL extract has been modified to only extract data for items marked as forecastable, as indicated by the item_master_foreceast_ind indicator.

Rmse_wh_cur_inventory.ksh

This RETL extract has been modified to show inventory against the primary supplier pack size that is set up on the ITEM_MASTER table. If the current inventory is not perfectly divisible by the primary supp pack size, or there is not enough to meet the primary supp pack size, then the remainder is sent as eaches. If the item is a Simple Pack, then the extract includes the valid packsku size on V_PACKSKU_QTY. It includes the component item with only an order multiple of 1.

Additionally, this RETL extract has been modified to only extract data for items marked as forecastable, as indicated by the item_master_foreceast_ind indicator.

Rmse_future_delivery_order.ksh

This RETL extract has been modified to show on-order inventory against the primary supplier pack size that is set up on ITEM_MASTER. Referencing data on ordsku, the extract sums up the "each" quantity on the order and then uses the current item/supp/country packsize and item_master primary pack size information to communicate to AIP. If the on-order inventory is not perfectly divisible by the primary supp pack size, or there is not enough to meet the primary pack size, then the remainder is sent in the standard unit of measure. Additionally, if AIP sends any future orders that are not in a valid inventory size as held in the Item-Supplier-Country table, then all inventory is put against the standard unit of measure.

If the item is a Simple Pack, then the extract communicates using the valid packsku size on V_PACKSKU_QTY. It includes the component item with only an order multiple of 1.

In addition, this RETL extract has been modified to only extract data for items marked as forecastable, as indicated by the item_master_foreceast_ind indicator.

Rmse_future_delivery_tsf.ksh

This RETL extract has been modified to show in-transfer inventory against the primary supplier pack size that is set up on ITEM_MASTER. Referencing data on TSFDETAIL, the extract sums up the "each" quantity on the tsf and then uses the current item/supp/country packsize and ITEM_MASTER primary pack size information to communicate to AIP. If the in-transfer inventory is not perfectly divisible by the primary supp pack size, or there is not enough to meet the primary pack size, then the remainder is sent in the standard unit of measure. Additionally, if AIP sends any future transfers that are not in a valid inventory size, as held in the Item-Supplier-Country table, then all inventory is put against the standard unit of measure.

If the item is a Simple Pack, then the extract communicates using the valid packsku size on V_PACKSKU_QTY. It includes the component item with only an order multiple of 1.

In addition, this RETL extract has been modified to only extract data for items marked as forecastable, as indicated by the item_master_foreceast_ind indicator.

Rmse_tsf_in_well.ksh

This extract has been modified in the same way as rmse_future_delivery_tsf.ksh. Also, this RETL extract has been modified to only extract data for items marked as forecastable, as indicated by the item_master_foreceast_ind indicator.

Rmse_future_delivery_alloc.ksh

This RETL extract has been modified to show allocated inventory against the primary supplier pack size that is set up on ITEM_MASTER. Referencing data on ORDSKU and ALLOC_HEADER, the extract sums up the “each” quantity on the allocation and then uses the current item/supp/country packsize and item_master primary pack size information to communicate to AIP. If the allocated inventory is not perfectly divisible by the primary pack size, or there is not enough to meet the primary pack size, then the remainder is sent in the standard unit of measure. Additionally, if AIP sends any future allocations that are not in a valid inventory size as held in the Item-Supplier-Country table, then all inventory is put against the standard unit of measure.

If the item is a simple pack, the extract communicates using the valid packsku size on V_PACKSKU_QTY. It includes the component item with only an order multiple of 1.

In addition, this RETL extract has been modified to only extract data for items marked as forecastable, as indicated by the item_master_foreceast_ind indicator.

Finally, this RETL extract has been modified to extract data for inbound allocations into a warehouse, as well as inbound allocations into stores.

Rmse_alloc_in_well.ksh

This extract has been modified in the same way as rmse_future_delivery_alloc.ksh. Also, this RETL extract has been modified to only extract data for items marked as forecastable, as indicated by the item_master_foreceast_ind indicator.

Additionally, this RETL extract has been modified to extract data for ‘in well’ allocations into a warehouse, as well as ‘in well’ allocations into stores.

Rmse_aip_item_master.ksh

This RETL extract has been modified to only extract data for items marked as forecastable, as indicated by the item_master_foreceast_ind indicator.

Rmse_banded_items.ksh

This RETL extract has been modified to only extract data for items marked as forecastable, as indicated by the item_master_foreceast_ind indicator.

Rmse_item_loc_traits.ksh

This RETL extract has been modified to only extract data for items marked as forecastable, as indicated by the item_master_foreceast_ind indicator.

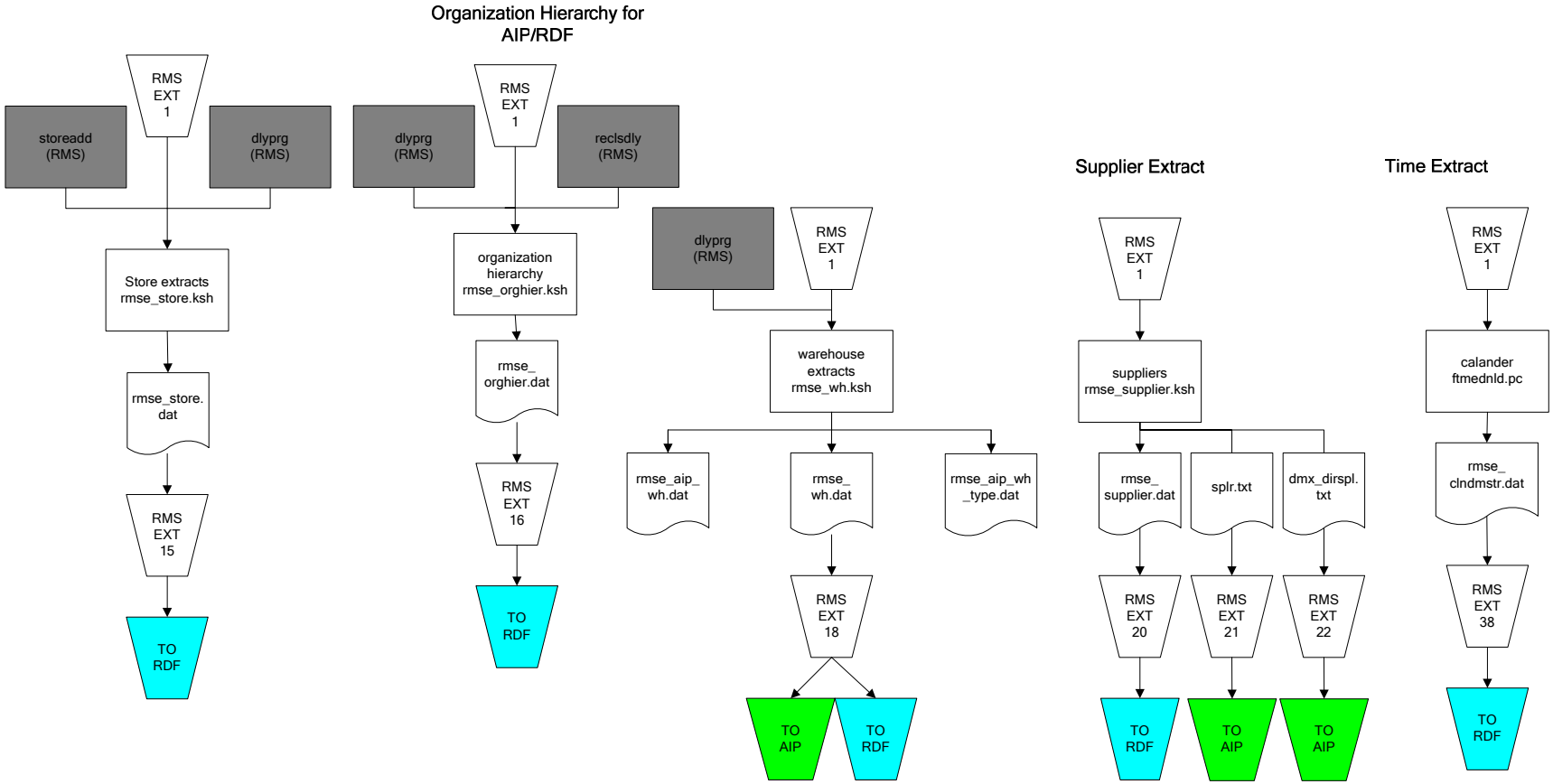
Rmse_item_retail.ksh

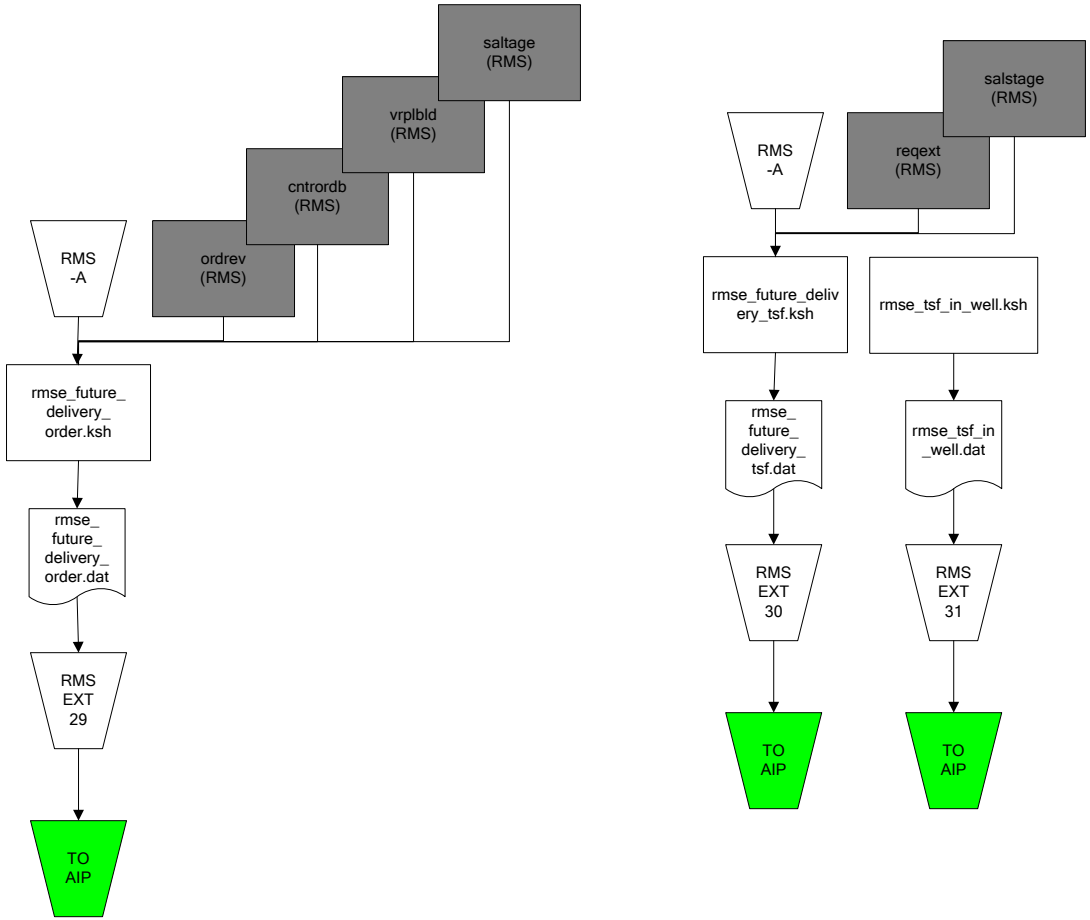
This RETL extract has been modified to only extract data for items marked as forecastable, as indicated by the item_master_foreceast_ind indicator.

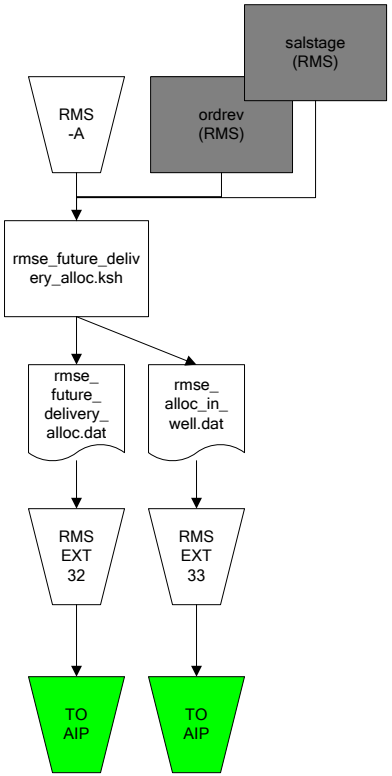
Rmse_store_cur_inventory.ksh

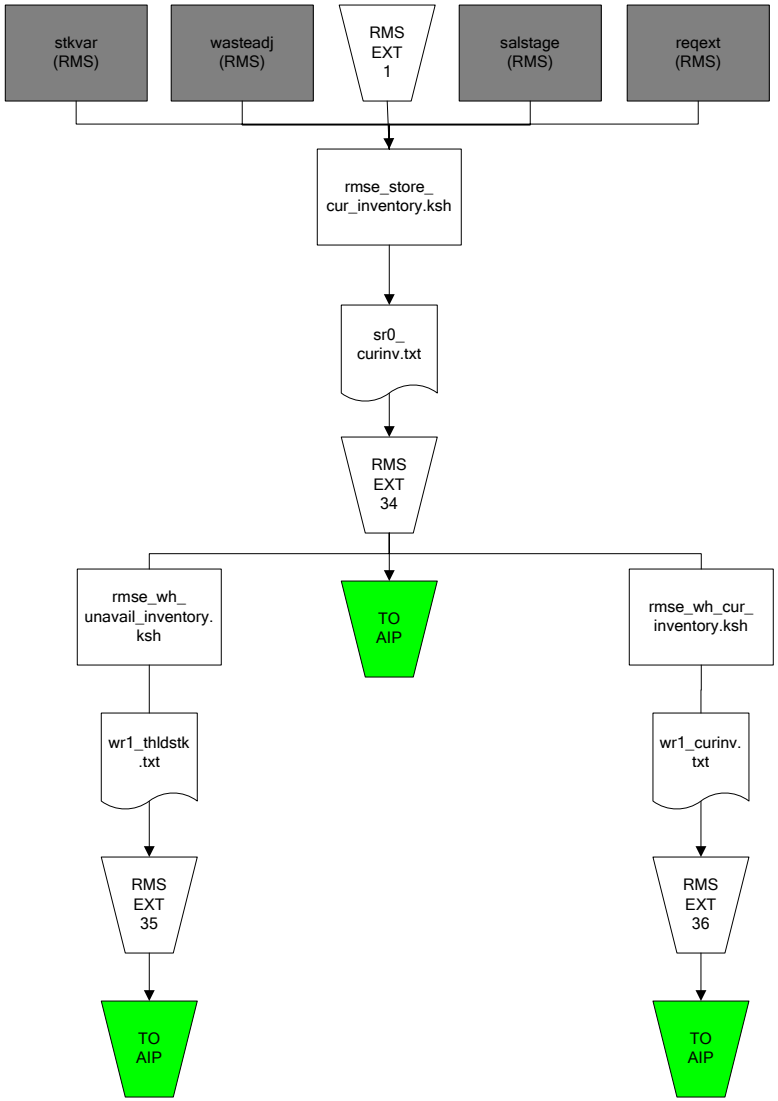
This RETL extract has been modified to only extract data for items marked as forecastable, as indicated by the item_master_foreceast_ind indicator.

Modified flow diagrams




saltage
(RMS)
vrplbld
(RMS)
cntrordb
(RMS)
ordrev
(RMS)
RMS
-A
rmse_future_
delivery_
order.ksh
rmse_
future_
delivery_
order.dat
RMS
EXT
29
TO
AIP
salstage
(RMS)
reqext
(RMS)
RMS
-A
rmse_future_deliv
ery_tsf.ksh
rmse_tsf_in_well.ksh
rmse_
future_
delivery_
tsf.dat
rmse_tsf_in
_well.dat
RMS
EXT
30
RMS
EXT
31
TO
AIP
TO
AIP


salstage
(RMS)
RMS
-A
ordrev
(RMS)
rmse_future_deliv
ery_alloc.ksh
rmse_
future_
delivery_
alloc.dat
rmse_
alloc_in_
well.dat
RMS
EXT
32
RMS
EXT
33
TO
AIP
TO
AIP


stkvar
(RMS)
wasteadj
(RMS)
RMS
EXT
1
salstage
(RMS)
reqext
(RMS)
rmse_store_
cur_inventory.ksh
sr0_
curinv.txt
RMS
EXT
34
rmse_wh_
unavail_inventory.
ksh
TO
AIP
rmse_wh_cur_
inventory.ksh
wr1_thldstk
.txt
wr1_curinv.
txt
RMS
EXT
35
RMS
EXT
36
TO
AIP
TO
AIP

Modified file layouts

Data file name: rmse_aip_wh_type.txt

Business rules:

- This file holds multiple warehouses along with their warehouse types.

Field name	Data type	Length	Required	Source
warehouse	Number	20	TRUE	WH.WH
wh_type	Varchar2	6	FALSE	WH.WH_TYPE

Data file name: rmse_aip_wh.txt

Business rules:

- This file holds multiple warehouses along with their warehouse names.

Field name	Data type	Length	Required	Source
warehouse_chamber	VARCHAR2	20	TRUE	WH.WH
warehouse_chamber_description	VARCHAR2	40	TRUE	WH.WH_NAME
warehouse	NUMBER	20	TRUE	WH.WH
warehouse_description	VARCHAR2	40	TRUE	WH.WH_NAME

Data file name: rmse_alloc_in_well.txt

Business rules:

- This file holds allocations that are currently coming in to a location and the transfers for the allocation that have been shipped.
- If the proposed release date has passed, the day field is the current date plus the expected transit times.
- If the proposed release date is in the future, the day field is the proposed release date.

Field name	Data type	Length	Required	Source
transit_times.transit_time + period.vdate OR alloc_header.release_date	Varchar2	9	TRUE	TRANSIT_ TIMES, PERIOD, ALLOC_ HEADER
to_loc	Number	20	TRUE	ALLOC_ DETAIL
item	Varchar2	20	TRUE	ITEM_ MASTER, V_ PACKSKU_ QTY (based on pack_ind on item_master)
alloc_reserve_qty	Number	8	TRUE	ALLOC_ DETAIL
order_multiple	Number	4	TRUE	ITEM_SUPP_ COUNTRY (based upon primary case size on item_master)

Data file name: rmse_tsf_in_well.txt

Business rules:

- This file holds the quantity of an item that exists on transfers that have shipments created for them but have not been received at the location yet.

Field name	Data type	Length	Required	Source
day	Varchar2	9	TRUE	Calculated
loc	Number	20	TRUE	TSFHEAD.FROM_LOC
item	Varchar2	20	TRUE	TSFDETAIL.ITEM
order_multiple	Number	4	TRUE	ITEM_SUPPLIER (based upon primary case size on item_master)
tsf_reserve_qty	Number	8	TRUE	TSFDETAIL.TSF_QTY - TSFDETAIL.SHIP_QTY

Chapter 3 – Pro*C batch designs

Retek supplier cost changes update [SCCEXT]

Design overview

The Supplier Cost Change Extract (sccext) module selects supplier cost change records, which are set to go into effect tomorrow and updates the RMS item/supplier table with the new cost. The item/location tables are also updated with the new cost if the cost change is a base cost change (supplier is the primary supplier for the item). In addition to the updates to the unit costs on the item/location tables when the base cost changes, records are also written for price history, supplier history and transaction-level stock ledger (if using standard cost). The costs on approved orders may also be updated if the recalculate order indicator is set to Yes for the item/supplier combination. The Pre/Post program for multi-threadable programs should be run after this program to update the status of the cost change to Extracted.

Tables affected:

TABLE	INDEX	SELECT	INSERT	UPDATE	DELETE
COST_SUSP_SUP_DETAIL	No	Yes	No	No	No
COST_SUSP_SUP_HEAD	No	Yes	No	No	No
DEAL_CALC_QUEUE	No	Yes	Yes	Yes	No
DEAL_SKU_TEMP	No	No	Yes	No	No
PERIOD	No	Yes	No	No	No
PRICE_HIST	No	No	Yes	No	No
ITEM_SUPPLIER	No	Yes	No	Yes	No
SUPS	No	Yes	No	No	No
SYSTEM_OPTIONS	No	Yes	No	No	No
V_RESTART_COST_CHG	No	Yes	No	No	No
COST_SUSP_SUP_DETAIL_LOC	No	Tes	No	No	No
ORDLOC	No	Yes	No	Yes	No
ORDHEAD	No	Yes	No	No	No
ITEM_SUPP_COUNTRY	No	Yes	No	Yes	No
ITEM_SUPP_COUNTRY_LOC	No	Yes	No	Yes	No
ITEM_SUPP_COUNTRY_BRACKET_COST	No	Yes	Yes	Yes	No
ITEM_MASTER	No	Yes	No	No	No
PACKITEM	No	Yes	No	No	No

Scheduling constraints

Processing Cycle:	Ad Hoc (run as required since not impacted by inventory position)
Scheduling Diagram:	N/A
Pre-Processing:	N/A
Post-Processing:	sccext_post() – records in cost_susp_sup_head are updated to set status = 'E' where status = 'A' and active_date = (select vdate + 1 from period);
Threading Scheme:	COST_CHANGE (V_restart_cost_chg)

Restart recovery

```
EXEC SQL DECLARE c_head CURSOR FOR
    SELECT h.cost_change,
           h.reason,

      FROM v_restart_cost_chg rv,
cost_susp_sup_head h
     WHERE h.status = 'A'
       AND h.active_date = to_date(:tomorrow, 'DDMMYYYY')

       AND rv.driver_value = h.cost_change
       AND rv.driver_name  = :ora_restart_driver_name
       AND rv.thread_val   = :ora_restart_thread_val
       AND rv.num_threads  = :ora_restart_num_threads
       AND (h.cost_change > NVL(:ora_restart_cost_chg, -999) OR
               (h.cost_change = :ora_restart_cost_chg))
  ORDER BY h.cost_change,
```

Program flow

N/A

Shared modules

COST_EXTRACT_SQL.UPDATE_COSTS – This function will update costs in RMS based on the cost change and cost change reason passed in (all processing will be handled in this package). If the cost change reason is 1 or 2 for the cost change, then the brackets will be inserted onto ITEM_SUPP_COUNTRY_BRACKET_COST. The unit costs on ITEM_SUPP_COUNTRY_LOC will be updated with the default bracket cost. If the location is the primary location, update the unit cost on ITEM_SUPP_COUNTRY. If the item is on an approved order, which is not associated to any contract and none of the item in the order has been received fully or partially, then the item unit_cost and unit_cost_init on the order will be updated. If the item is a component item in a buyer pack, the pack unit_cost and unit_cost_init will be updated using the new component cost. If the item has children, the same processing as described above will occur for all children items. If it is run in a multi-channel environment, only virtual locations will receive the new brackets. For any other cost change reason:

- If bracket are to be updated with new costs, ITEM_SUPP_COUNTRY and ITEM_SUPP_COUNTRY_LOC will be updated with the default bracket costs. If the item is on an approved order, which is not associated to any contract and none of the item in the order has been received fully or partially, its unit_cost and unit_cost_init will be updated with the new base cost. If the item is in a buyer pack, the pack will be updated with the new base cost of the item. If the item has children, the child items will be processed in the same manner.
- If location costs are only updated, ITEM_SUPP_COUNTRY_LOC will be updated with the new location costs. The unit cost will be updated on ITEM_SUPP_COUNTRY if the location updated is the primary location. If the item is on an approved order, which is not associated to any contract and none of the item in the order has been received fully or partially, its unit_cost and unit_cost_init will be updated with the new base cost. If the item is in a buyer pack, the pack will be updated with the new base cost of the item. If the item has children, the child items will be processed in the same manner.
- If the item is costed at the item level only, then ITEM_SUPP_COUNTRY will be updated with the new unit cost. This cost will be updated to all locations on ITEM_SUPP_COUNTRY_LOC. If the item is on an approved order, which is not associated to any contract and none of the item in the order has been received fully or partially, its unit_cost and unit_cost_init will be updated with the new base cost. If the item is in a buyer pack, the pack will be updated with the new base cost of the item. If the item has children, the child items will be processed in the same manner.

When the unit_cost and unit_cost_init for an approved order is updated with the new base cost of the item, a record for that order will be inserted into the deal_calc_queue table with recalc_ind ='N'.

If these three bullets occur in a multi-channel environment and the location is a warehouse, only the virtual warehouse locations will be updated.

Function level description

```
process() – main processing function

while(1) – main loop
    {
        call COST_EXTRACT_SQL.UPDATE_COSTS ()


     }
```

I/O specification

N/A

Technical issues

N/A

Order deal discount [ORDDSCNT.PC/.H]

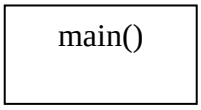
Design overview

This new batch program will fetch orders that need to be recalculated for cost from DEAL_CALC_QUEUE. Using the dealordlib shared library, it will update the unit cost on the argument order along with potentially populating ORDLOC_DISCOUNT, and ORDHEAD_DISCOUNT. This program is a batch-end wrapper for the library. For an in-depth description of what happens inside the library, please refer to dealordlib Master Batch Library Design document.

Stored procedures/shared modules (maintainability)

Dealordlib.pc/.h

Program flow



Function level description

Main(): Standard Retek main function. Validates input parameters, calls init, process and final. Logs appropriate message.

Init(): Standard Retek init function. Calls retek_init(), and the library's lib_init().

Process(): Drives the rest of the program:

- Call size_order_array(), size_deal_array(), size_disct_bld_array() and size_disc_item_array() from the dealordlib library.
- Array-fetch the driving cursor.
- Call check_lock_and_validate() from the dealordlib library.

- For each fetched record:
  - Call reset_order() from the dealordlib library.
  - Call get_deals() from the dealordlib library.
  - Call get_deal_ct() from the dealordlib library to see how many deals were found. If none, move to next order.
  - Call make_ordloc_build() from the dealordlib library.
  - Call buyer_packs_detected() from the dealordlib library to see if make_ordloc_build() found any buyer packs on the ORDLOC table. If number returned is higher than 0, write a warning.
  - If the order is approved, call get_otb_cost() to get the pre-deals OTB cost.
  - Call find_items_for_deal() to create a combination data between deals and items for this order (deal + item deal applies to).
  - Call calculate_cost() to calculate new costs.
  - If the order is approved and there were transaction level deals found for the order, call process_txn_discount() to apply those deals. Also call get_otb_cost() to get the post-deals OTB cost. Records will be insert into table DISC_OTB_APPLY and later picked up by batch program discotbapply.pc to make sure the change in costs is communicated to OTB. Also call insert_rev_orders() from the library to make sure rev_orders table is properly inserted if costs changed on an approved order.
  - Call free_item_cost_array() from the library.
  - Call delete_ordloc_disc_bld() to clean the ORDLOC_DISCOUNT_BUILD staging table.
- Call cleanup_discount_tables() to delete unnecessary data.
- Call delete_dcq() to delete the DEAL_CALC_QUEUE record that was just processed.
- Call retek_force_commit() to commit work.

Final():

Performs restart/recovery close logic. Calls retek_close().

Input specifications

'Table-to-table'

Select data from:

Table Name	Column Name	Column Type	Transformation
DEAL_CALC_QUEUE	ORDER_NO	NUMBER(8)	NONE
DEAL_CALC_QUEUE	RECALC_ALL_IND	VARCHAR2(1)	NONE
DEAL_CALC_QUEUE	OVERRIDE_MANUAL_IND	VARCHAR2(1)	NONE
DEAL_CALC_QUEUE	ORDER_APPR_IND	VARCHAR2(1)	NONE
ORDHEAD	SUPPLIER	NUMBER(10)	NONE
ORDHEAD	IMPORT_ORDER_IND	VARCHAR2(1)	NONE

Table Name	Column Name	Column Type	Transformation
ORDHEAD	IMPORT_COUNTRY_ID	VARCHAR2(3)	NONE
ORDHEAD	NOT_BEFORE_DATE	DATE	NONE
ORDHEAD	CURRENCY_CODE	VARCHAR2(3)	NONE
ORDHEAD	STATUS	VARCHAR2(1)	NONE
ORDHEAD	EXCHANGE_RATE	NUMBER(20,10)	NONE
SUPS	CURRENCY_CODE	VARCHAR2(3)	NONE
SUPS	BRACKET_COSTING_IND	VARCHAR2(1)	NONE

Output specifications

'Table-To-Table'

See dealordlib Master Batch Library Design.

Scheduling considerations

This program should run after ditinsrt. It should run twice in the nightly batch run; first time in Phase 3 just before sccext is run and second time in Phase 4 before discotbapply, dealcls or dealprg in the deals batch cycle.

Locking strategy

See dealordlib Master Batch Library Design.

Restart/recovery

This program has inherent restartability, since records are deleted from deal_calc_queue as they are processed.

Recommended maximum commit counter: --- (low)

Performance considerations

N/A

Security considerations

N/A

Unit test considerations

See dealordlib Batch Library UTP.

Design assumptions

See dealordlib Master Batch Library Design.

Outstanding design issues

N/A