

Retek Supplier Cost Changes Update [sccect]

Design Overview

The Supplier Cost Change Extract (sccect) 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.

If the Batch with Online Users indicator is set to 'Y', it will bulk lock the records that need to be modified before calling the package that performs the necessary updates. Information about the records that were not processed due to locking will be inserted to the batch_lock_log table.

Tables Affected:

TABLE	INDEX	SELECT	INSERT	UPDATE	DELETE
COST_SUSP_SUP_DETAIL	No	Yes	No	No	No
COST_SUSP_SUP_HEAD	No	Yes	No	Yes	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	Yes	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_L OC	No	Yes	No	Yes	No
ITEM_SUPP_COUNTRY_B RACKET_COST	No	Yes	Yes	Yes	No
ITEM_MASTER	No	Yes	No	No	No
PACKITEM	No	Yes	No	No	No
COST_CHANGE_LOCK_T EMP	No	Yes	Yes	No	Yes
BATCH_LOCK_LOG	No	Yes	Yes	No	Yes
ITEM_EXP_DETAIL	No	Yes	No	Yes	No
ITEM HTS ASSESS	No	Yes	No	Yes	No
ORDLOC_EXP	No	Yes	No	Yes	No
ORDSKU HTS ASSESS	No	Yes	No	Yes	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, then the item cost on the order will be updated. If the item is a component item in a buyer pack, the pack cost 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 ran 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, its cost will be updated with the new base cost. If the item is in a buyer pack, the pack will be updated with

the new cost 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, its cost will be updated with the new base cost. If the item is in a buyer pack, the pack will be updated with the new cost 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, its cost will be updated with the new base cost. If the item is in a buyer pack, the pack will be updated with the new cost base cost of the item. If the item has children, the child items will be processed in the same manner.

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

init() – fetch SYSTEM_OPTIONS.BTCH_W_USR_IND into a global variable. Deletes record from the batch_lock_log table. It also checks if there are previous locks in case of restart recovery.

process() – main processing function:

while(1) – main loop

```
{
    if BTCH_W_USR_IND = 'Y'
        call check_locks ()
    if (BTCH_W_USR_IND = 'N') or (BTCH_W_USR_IND = 'Y' and lock
        does not exist)
        call COST_EXTRACT_SQL.UPDATE_COSTS ()
}
```

check_locks() - check locks by joining the temporary table and target table.

I/O Specification

N/A

Technical Issues

N/A