

Batch Detail Design

I. Functional Area

POS Download

II. Module Affected

posdnld.pc

III. Design Overview

The posdnld program is used to download pos_mods records created in the RMS to the store POS systems. This program has one output file which contains all records for all stores in a given run. This program uses the Retek standard file format FHEAD, FDETL, FTAIL.

This program needs to be modified to work in when the users can be in the system at any given time. The system_options table will be modified to introduce a new indicator named Batch with Online Users (btch_w_usr_ind). This indicator is set to 'Y' if users can be in the system at any time, otherwise it is set to 'N'.

In this program, the deletion of records done in post programs must be done after processing the records. This would eliminate the need to run the post programs in the batch schedule. The truncation of records done in prepost, specifically in posdnld_post function, will be incorporated in this batch program.

Program Flow



IV. Stored Procedures / Shared Modules (Maintainability)

pos_config_sql.check_item - Updates POS item configuration information that is downloaded to the stores by poscdnld.pc.

V. Input Specifications

All input comes from the pos_mods table. All columns of this table can be NULL with the exception of tran_type and store. Most columns should default to blank (spaces) with the exception of:

- new_price, new_multi_units, new_multi_units_retail, proportional_tare_pct and fixed_tare_value. These should default to zero (0).
- start_date, start_time and end_time. These should default to period.vdate + 1.



VI. Output Specifications

Output File

Record Name	Field Name	Field Type	Default Value	Description
File Header	File Type Record Descriptor	Char(5)	FHEAD	Identifies file record type
	File Line Identifier	Number ID(10)	Sequential number Created by program.	ID of current line being created for output file.
	File Type Definition	Char(4)	POSD	Identifies file as 'POS Download'
	File Create Date	Char(8)	Create date (vdate).	Current date, formatted to 'YYYYMMDD'.
File Detail	File Type Record Descriptor	Char(5)	FDETL	Identifies file record type
	File Line Identifier	Number ID(10)	Sequential number. Created by program.	ID of current line being created for output file.
	Location Number	Number(10)	Store	Contains the store location that has been affected by the transaction
	Update Type	Char(1)	Update type. Created by program.	Code used for client specific POS system. <ul style="list-style-type: none"> 1 - Transaction Types 1 & 2. 2 - Transaction Types 10 thru 18, 31 & 32, 50 thru 57, 59 thru 64. 3 - Transaction Types 21 & 22 4 - Transaction Types 25 & 26 0 - All other Transaction Types. These should never exist.
	Start Date	Char(8)	Start_date or vdate + 1 if NULL.	The effective date for the action determined by the transaction type of the record. Formatted to 'YYYYMMDD'.
	Time	Char(6)	Start_time, End_time or start_date.	This field will be used in conjunction with starting a promotion (Transaction Type = 31). Start time will indicate the time of day that the promotion is scheduled to start. This field will also be used in conjunction with ending a promotion (Transaction Type = 32). Any other Transaction Type will use the time from the start_date column. Formatted to 'HH24MISS'.
	Transaction Type	Number(2)	Tran_type	Indicates the type of transaction to determine what Retek action is being sent down to the stores from the Retek pos_mods table. Valid values include: <ul style="list-style-type: none"> 01 - Add new transaction level item 02 - Add new lower than transaction level item 10 - Change Short Description of existing item 11 - Change Price of an existing item 12 - Change Description of an existing item 13 - Change Department/Class/Subclass of an existing item 16 - Put Item on Clearance 17 - Change existing item's Clearance Price 18 - Remove Item from Clearance and Reset 20 - Change in VAT rate 21 - Delete existing transaction level item 22 - Delete existing lower than transaction level item 25 - Change item's status



				<ul style="list-style-type: none"> • 26 - Change item's taxable indicator • 31 - Promotional item - Start maintenance • 32 - Promotional item - End maintenance • 50 - Change item's launch date • 51 - Change item's quantity key options • 52 - Change item's manual price entry options • 53 - Change item's deposit code • 54 - Change item's food stamp indicator • 55 - Change item's WIC indicator • 56 - Change item's proportional tare percent • 57 - Change item's fixed tare value • 58 - Change item's rewards eligible indicator • 59 - Change item's electronic marketing clubs • 60 - Change item's return policy • 61 - Change item's stop sale indicator • 62 - Change item's returnable indicator • 63 - Change item's refundable indicator • 64 - Change item's back order indicator
	Item Number ID	Char(25)	Item	This field identifies the unique alphanumeric value for the transaction level item. The ID number of a item from the Retek item_master table.
	Item Number Type	Char(6)	Item_number_type	This field identifies the type of the item number ID.
	Format ID	Char(1)	Format_id	This field identifies the type of format used if the item_number_type is 'VPLU'.
	Prefix	Number(2)	Prefix	This field identifies the prefix used if the item_number_type is 'VPLU'. In case of single digit prefix, the field will be right-justified with blank padding.
	Reference Item	Char(25)	Ref_item	This field identifies the unique alphanumeric value for an item one level below the transaction level item.
	Reference Item Number Type	Char(6)	Ref_Item_number_type	This field identifies the type of the ref item number ID.
	Reference Item Format ID	Char(1)	Ref_Format_id	This field identifies the type of format used if the ref item_number_type is 'VPLU'.
	Reference Item Prefix	Number(2)	Ref_Prefix	This field identifies the prefix used if the ref item_number_type is 'VPLU'. In case of single digit prefix, the field will be right-justified with blank padding.
	Item Short Description	Char(20)	Item_short_desc	Contains the short description associated with the item.
	Item Long Description	Char(100)	Item_long_desc	Contains the long description associated with the item.
	Department ID	Number(4)	Dept	Contains the item's associated department.
	Class ID	Number(4)	Class	Contains the item's associated class.
	Subclass ID	Number(4)	Subclass	Contains the item's associated subclass.
	New Price	Number(20)	New_price	Contains the new effective price in the selling unit of measure for an item when the transaction type identifies a change in price. Otherwise, the current retail price is used to populate this field. This field is stored in the local currency.
	New Selling UOM	Char(4)	New_selling_UOM	Contains the new selling unit of measure for an item's single-unit retail.
	New Multi Units	Number(12)	New_multi_units	Contains the new number of units sold together for multi-unit pricing. This field is only filled when a multi-unit price change is being made.



	New Multi Units Retail	Number(20)	New_multi_units_retail	Contains the new price in the selling unit of measure for units sold together for multi-unit pricing. This field is only filled when a multi-unit price change is being made. This field is stored in the local currency.
	New Multi Selling UOM	Char(4)	New_multi_selling_UOM	Contains the new selling unit of measure for an item's multi-unit retail.
	Status	Char(1)	Status	Populates if tran_type for the item is 1(new item added) or 25 (change item status) or 26 (change taxable indicator). Contains the current status of the item at the store. Valid values are: <ul style="list-style-type: none"> • A = Active • I = Inactive • D = Delete • C = Discontinued
	Taxable Indicator	Char(1)	Taxable_ind	Populates if tran_type for the item is 1 (new item added) or 25 (change item status) or 26 (change taxable indicator). Indicates whether the item is taxable at the store. Valid values are 'Y' or 'N'.
	Promotion Number	Number(10)	Promotion	This field contains the number of the promotion for which the discount originated. This field, along with the Mix Match Number or Threshold Number is used to isolate a list of items that tie together with discount information.
	Mix Match Number	Number(10)	Mix_match_no	This field contains the number of the mix and match in a promotion for which the discount originated. This field, along with the promotion, is used to isolate a list of items which tie together with the mix and match discount information.
	Mix Match Type	Char(1)	Mix_match_type	This field identifies which types of mix and match record this item belongs to. The item can either be a buy (exists on PROM_MIX_MATCH_BUY) or a get (exists on PROM_MIX_MATCH_GET) item. This field is only populated when the MIX_MATCH_NO is populated. Valid values are: <ul style="list-style-type: none"> • B - Buy • G - Get
	Threshold Number	Number(10)	Threshold_no	This field contains the number of the threshold in a promotion for which the discount originated. This field, along with the promotion, is used to isolate a list of items that tie together with discount information.
	Launch Date	Char(8)	Launch_date	Date that the item should first be sold at this location, formatted to 'YYYYMMDD'.
	Quantity Key Options	Char(6)	Qty_key_options	Determines whether the price can/should be entered manually on a POS for this item at the location. Valid values are in the code_type 'RPO'. Current values include 'R - required', 'P - Prohibited'.
	Manual Price Entry	Char(6)	Manual_price_entry	Determines whether the price can/should be entered manually on a POS for this item at the location. Valid values are in the code_type 'RPO'. Current values include 'R - required', 'P - Prohibited', and 'O - Optional'.



	Deposit Code	Char(6)	Deposit_code	Indicates whether a deposit is associated with this item at the location. Valid values are in the code_type 'DEPO'. Additional values may be added or removed as needed. Deposits are not subtracted from the retail of an item uploaded to RMS, etc. This kind of processing is the responsibility of the client and should occur before sales are sent to any Retek application.
	Food Stamp Indicator	Char(1)	Food_stamp_ind	Indicates whether the item is approved for food stamps at the location.
	WIC Indicator	Char(1)	Wic_ind	Indicates whether the item is approved for WIC at the location.
	Proportional Tare Percent	Number(12)	Proportional_tare_pct	Holds the value associated of the packaging in items sold by weight at the location. The proportional tare is the proportion of the total weight of a unit of an item that is packaging (i.e. if the tare item is bulk candy, this is the proportional of the total weight of one piece of candy that is the candy wrapper). The only processing RMS does involving the proportional tare percent is downloading it to the POS.
	Fixed Tare Value	Number(12)	Fixed_tare_value	Holds the value associated of the packaging in items sold by weight at the location. Fixed tare is the tare of the packaging used to (i.e. if the tare item is bulk candy, this is weight of the bag and twist tie). The only processing RMS does involving the fixed tare value is downloading it to the POS. Fixed tare is not subtracted from items sold by weight when sales are uploaded to RMS, etc. This kind of processing is the responsibility of the client and should occur before sales are sent to any Retek application.
	Fixed Tare UOM	Char(4)	Fixed_tare_uom	Holds the unit of measure value associated with the tare value. The only processing RMS does involving the proportional tare value and UOM is downloading it to the POS. This kind of processing is the responsibility of the client and should occur before sales are sent to any Retek application.
	Reward Eligible Indicator	Char(1)	Reward_eligible_ind	Holds whether the item is legally valid for various types of bonus point/award programs at the location.
	Elective Marketing Clubs	Char(6)	Elect_mtk_clubs	Holds the code that represents the marketing clubs to which the item belongs at the location. Valid values can belong to the code_type 'MTKC'. Additional values can be added or removed from the code type as needed
	Return Policy	Char(6)	Return_pocily	Holds the return policy for the item at the location. Valid values for this field belong to the code_type 'RETP'.
	Stop Sale Indicator	Char(1)	Stop_sale_ind	Indicates that sale of the item should be stopped immediately at the location (i.e. in case of recall etc).
	Returnable Indicator	Char(1)	Returnable_ind	Indicates that the item is returnable at the location when equal to 'Y'es. Indicates that the item is not returnable at the location when equal to 'N'o.
	Refundable Indicator	Char(1)	Refundable_ind	Indicates that the item is refundable at the location when equal to 'Y'es. Indicates that the item is not refundable at the location when equal to 'N'o.
	Back Order Indicator	Char(1)	Back_order_ind	Indicates that the item is back orderable at the location when equal to 'Y'. Indicates that the item is not back orderable when equal to 'N'o.
	Vat Code	Char(6)		Indicates the VAT code used with this item.
	Vat Rate	Number(20,10)		Indicates the VAT rate associated with this item and VAT code.



	Class Vat Indicator	Char(1)		Indicates whether or not the class VAT indicator is on or off for the class that this item exists in.
File Trailer	File Type Record Descriptor	Char(5)	FTAIL	Identifies file record type
	File Line Identifier	Number ID(10)	Sequential number. Created by program.	ID of current line being created for output file.
	File Record Counter	Number ID(10)	Number of FDETL records. Created by program.	Number of records/transactions processed in current file (only records between head & tail)

VII. Function Level Description

init - This function initializes restart/recovery for this program. It also retrieves system variables (period.vdate and vdate + 1) and system options (btch_w_usr_ind), opens the output file and write the FHEAD record.

process - This function drives the processing of the program. It calls size_arrays() function to size the arrays used in this program and also, when done, it calls free_arrays() to release any memory it has been allocated. The driving cursor is opened and fetched here, which retrieves all the records from pos_mods where the pos_mods.store value is greater than zero. If the Transaction Type is 31, then the time field returned by the cursor should be the start time, else if the Transaction Type is 32, then the time field should be the end time. If the Transaction Type is something else or if either the start time or end time is NULL, blanks should be used.

Once the records are fetched, it is then inserted to a global temporary table to check for bulk locks on all records fetched. If the records fetched is locked, it will fetch the next logical unit of work, otherwise it processes the fetched records. While processing, if the Transaction Type of the record fetched is 1 or 21 then pos_config_check() is called. The write_rec() function is called to perform processing on all records fetched. Restart/Recovery and committing of records is also performed here. Before committing the records, it will check for system_options.btch_w_usr_ind. If it is 'Y', it will delete the fetched records in the pos_mods table otherwise it will not delete records from the pos_mods table.

final - This function will finish restart/recovery logic, write the FTAIL record and close the output.

size_arrays - This function initializes the size of the array used for the driving cursor fetch the size of the restart max counter on restart_control.

free_arrays - This function frees the array allocated in size_arrays.

write_rec - This function will prepare records for insert into the output file and write them as FDETL records. The Transaction Type will determine the Update Type. If the Transaction Type is 1, 25 or 26 then the status and taxable_ind columns must be outputted, otherwise these should remain blank.

pos_config_check - This function will call the package pos_config_sql.check_item(). If the Transaction type is 1, then a status of 'A' will be passed in. If the Transaction Type is 21, then a status of 'D' will be passed in. This function body should be commented out for A&P Phase 1a.

check_lock - This function will check bulk locks by joining the pos_mods table and the global temporary table by their row id with for update nowait clause.

VIII. Scheduling Considerations

Processing Cycle: PHASE 4 (daily)

Scheduling Diagram: This program is run towards the end of the batch run when all pos_mods records have been created for the transaction day.



Pre-Processing: N/A

Post-Processing: prepost.pc - posdnld_post() – records in POS_MODS are truncated if system_options.btch_w_usr_ind is 'N'.

Threading Scheme: v_restart_store

IX. Locking Strategy

This program has been modified for bulk locking strategy for pos_mods table. Before firing a delete, the records will be bulk locked with update for nowait clause. If batch is trying to acquire a lock that has been locked by some other application, the batch will skip the logical unit of work and continue processing other records.

X. Restart/Recovery

Restart/recovery for this program is set up at the store/item or item level. Threading is done by store using the v_restart_store view to thread properly.

XI. Performance Considerations

Both table and file restart/recovery must be used.

The commit_max_ctr field should be set to prevent excessive rollback space usage, and to reduce the overhead of file I/O. The recommended commit counter setting is 10000 records (subject to change based on experimentation).

XII. Security Considerations

Price changes for all stores are stored in a Unix file with the processes default permissions (umask). Care should be exercised so that this file cannot be tampered with.

XIII. Design Assumptions

Data columns required by a particular Transaction Type are filled in and correct.

XIV. Outstanding Design Issues

Issue Description	Priority	Issue Log Updated?
The columns pos_config_item.item and pos_merch_criteria.sku are of type number and have a length of 8. These columns are updated and referenced by the pos_config_sql.check_item() package function. These tables are then used by poscdnld.pc.	H	

XV. Appendix

None.

