

Batch Master Design

I. Approval and Distribution

The detailed design should be approved by:

Title	Name
Design Lead	Renata Scholtz

The detailed design should be distributed to:

Title	Name
Quality Control Lead	Stacy Hall

II. Functional Area

Scheduled Item Maintenance

III. Module Affected

sitmain.pc (Scheduled Item Maintenance Main)

IV. Design Overview

This new batch program will have two different processes where the first one needs to be complete before the second one could commence.

First process will involve the tracking of addition to item lists or location lists. If there is a pre-existing link that involves item list or location list where addition took place, this program will get the latest status from previous update that have occurred in the past. If the given item/loc relationship exists then the status and status_update_date field in the ITEM_LOC table will be updated accordingly. Otherwise, a new item/loc relationship will be created. After the update/insert into ITEM_LOC table all records in SIT_EXPLODE table with update_ind of 'Y' will be updated to 'N'.

In the second process, scheduled item maintenance will take place by obtaining the data from SIT_DETAIL table. It will update the item/location status on the item_loc table as specified on the sit_detail table for given item list/location list combination. If the item/location relationship does not exist at the time of update, it will be created. At the end of the second process, old records in the SIT_DETAIL table will be deleted.

If the Batch with Online Users indicator is set to 'Y', it will bulk lock the ITEM_LOC, SIT_DETAIL, and SIT_EXPLODE records first before making the necessary updates/deletes. Information about the records that were not processed due to locking will be inserted to the BATCH_LOCK_LOG table.

The LUW for this module is location/item.

V. Stored Procedures / Shared Modules (Maintainability)

NEW_ITEM_LOC: This procedure will be used to create new item/location relationship prior to status update.

VI. Program Flow



VII. 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(). It will retrieve the btch_w_usr_ind from the SYSTEM_OPTIONS table. If the Batch with Online Users indicator is set to 'Y', sitmain.pc's records in the BATCH_LOCK_LOG will be deleted if the batch is run for the first time.

Process(): Drives the business logic for this program. There are two processes that are involved in the program.

- Call process_additions().
- Call process_maintenance().

Process_additions(): This function will process the first driving cursor that consist of a union between two select statements. First select statement should get item, location from SIT_EXPLODE table, its corresponding status from SIT_DETAIL table and '1' where status_update_date is less than vdate + 1 in SIT_DETAIL table and update_ind from SIT_EXPLODE is 'Y'. It is necessary to check that select item/location relationship exists in ITEM_LOC table. The second select statement should get be similar to the first one except that is should select '2' instead of '1' and there should be no pre-existing item/location relationship in the ITEM_LOC table for each record.

- Array-fetch the driving cursor. For each record fetched:
 - If the Batch with Online Users indicator is set to 'Y', acquire lock on ITEM_LOC and SIT_EXPLODE records to be processed. If any of the records are locked by another process, a flag will be set and information about these records will be written to the BATCH_LOCK_LOG table.
 - Call update_insert() function.
 - If the Batch with Online Users indicator is set to 'Y', update the corresponding SIT_EXPLODE records with update_ind value of 'Y' to 'N'.
- If the Batch with Online Users indicator is set to 'N', after the insert/update process to the ITEM_LOC table, update all records in SIT_EXPLODE with update_ind value of 'Y' to 'N'.

Process_maintenance(): This function will process the second driving cursor that consist of a union between two select statements. First select statement should get item, location from SIT_EXPLODE table, its corresponding status from SIT_DETAIL table and '1' where status_update_date is vdate + 1 in SIT_DETAIL table. It is necessary to check that select item/location relationship exists in ITEM_LOC table.

The second select statement should get be similar to the first one except that is should select '2' instead of '1' and there should be no pre-existing item/location relationship in the ITEM_LOC table for each record.

- Array-fetch the driving cursor. For each record fetched:
 - If the Batch with Online Users indicator is set to 'Y', acquire lock on ITEM_LOC records to be processed. If any of the records are locked by another process, a flag will be set and information about these records will be written to the BATCH_LOCK_LOG table.
 - Call update_insert() function.
- After all the records in the array are processed, acquire lock on SIT_DETAIL records to be deleted. If any of the records are locked by another process, a flag will be set and information about these records will be written to the BATCH_LOCK_LOG table. If no record is locked, call delete_sit_detail() function to delete old records.

Update_insert(): This function will loop through the arrays fetched into in process_additions and process_maintenance.

- If update indicator is '1' then copy values to new arrays for updating.
- If update indicator is '2' then, call NEW_ITEM_LOC to create the item/location relationship.
- After looping through all the indices, array update the status of item/location in ITEM_LOC table using the arrays copied to when update indicator was '1'.

Delete_sit_detail(): This function will select all itemloc_link_id in SIT_DETAIL table that has status_update_date equal to vdate + 1. For each itemloc_link_id, it will delete any record that has status_update_date that is before vdate + 1.

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



Lock_item_loc(): This function will insert the records to be processed in the item_loc_lock_temp table and then call the check_lock_item_loc() funtion.

Check_lock_item_loc(): This function will acquire lock on the item_loc records that are present in the item_loc_lock_temp table.

Lock_sit_explode(): This function will insert the records to be processed in the sit_explode_lock_temp table and then call the check_lock_sit_explode() funtion.

Check_lock_sit_explode(): This function will acquire lock on the sit_explode records that are present in the sit_explode_lock_temp table.

Lock_sit_detail(): This function will insert the records to be processed in the sit_detail_lock_temp table and then call the check_lock_sit_detail() funtion.

Check_lock_sit_detail(): This function will acquire lock on the sit_detail records that are present in the sit_detail_lock_temp table.

VIII. Input Specifications

'Table-To-Table'

Select date from:

Table Name	Column Name	Column Type	Transformation
SIT_EXPLODE	ITEM	VARCHAR2(25)	NONE
SIT_EXPLODE	LOCATION	NUMBER(10)	NONE
SIT_DETAIL	STATUS_UPDATE_DATE	DATE	NONE
SIT_DETAIL	STATUS	VARCHAR2(1)	NONE

IX. Output Specifications

'Table-To-Table'

The following table will be updated:

Table Name	Column Name	Column Type	Transformation
ITEM_LOC	STATUS	VARCHAR2(1)	'A', 'I', 'C' or 'D'
ITEM_LOC	STATUS_UPDATE_DATE	DATE	NONE
SIT_EXPLODE	UPDATE_IND	VARCHAR2(1)	'N'

X. Scheduling Considerations

This program should always run after lclrbld.pc on an ad hoc basis. More specifically the running of this program is dependant on the date in SIT_DETAIL table. If today's date + 1 is equal to a date in SIT_DETAIL table then this program should be executed.

XI. Locking Strategy

Before processing a group of records, it will be locked first with the no wait clause. If this group of records includes a row that has already been locked by another application, the whole group will be skipped, a flag will be set, information about these records will be written to the batch_lock_log table, and a non-fatal error will be written in the log file. The batch will then continue processing the next group of records.



XII. Restart/Recovery

Restart recovery for this module will be based on location. Each of the processes will have separate restart/recovery maintenance. First process must be committed before second process can begin.

XIII. Performance Considerations

XIV. Security Considerations

N/A

XV. Unit Test Considerations

XVI. Design Assumptions

XVII. Outstanding Design Issues

Issue Description	Priority	Resolution

XVIII. Appendix

Technical Design – Scheduled Item Maintenance (Retek 10.0)
Functional Design – Scheduled Item Maintenance (Retek 10.0)
PRD – Scheduled Item Maintenance (Retek 10.0)

