# Purpose

In order workaround existing issues with the JDA item import process and to discover issues within the cleansed item spreadsheets, a process has been created to reconcile the spreadsheet contents to the data that was imported into the ESO database. This process will identify the issues and repair when possible.

# Process Steps

## Extract relevant columns from the spreadsheets and import it into the database

After each spreadsheet is transformed into an xml import, a final step should be executed to extract retail strategy and manufacturer inform directly from the spreadsheets. This process creates a SQL script that will insert the data into a worktable in the ESO. Here is an example of single statement for an item.

INSERT bc\_extract\_repair\_manufacturer\_and\_strategy (fileName, itemXrefId, manufacturer, itemStrategy, rmiXrefId, rmiMerchGroup)

SELECT 'Retail Item-601 General Merchan\_01.xlsx','1116651',NULL,'General Merchandise','1116651-1','Grocery'

After all the sql files are generated, the files can be easily merged into a single sql script using the copy command from a command prompt window. For example:

COPY \*.SQL AllFiles.SQL

In the case executed the AllFiles.SQL script on the DB server where the data was imported. Before running the file, drop and recreate the worktable using the script named “01 -ItemRepairCreateTable.sql” located in the program files\retail item directory. This will ensure that no data from prior imports are in the data.

## Resolve the id values and check the data

After the import has been completed, the bc\_extract\_repair\_manufacturer\_and\_strategy will be populated. The next step is to run the script named “02 – ItemRepairResolveIdValues.sql”. This script will check items and the retail strategy hierarchy:

### Items & Retail Items

The script attempts to match the item external id values from the spreadsheets to what is in the database. If the script does not find a match for all items in the worktable, the missing items will be displayed. If all items are matched, two possible messages will be displayed “Retail Items Verified” with a count of the number of retail items verified and “Not Sold Items Verified” with a count of the number of not sold items verified.

The script attempts to match the retail modified item external id values from the spreadsheets to what is in the database. If the script does not find a match for all items in the worktable, the missing retail modified items will be displayed. If all items are matched, a possible message will be displayed “RIM Items Verified” with a count of the number of retail items verified.

If there are items listed as missing, it typically indicates an error in the spreadsheet data. The most likely resolution is to resolve the issue and report the item. It may be helpful to query the mail table searching on the file name for the missing items.

### Retail Strategy Hierarchy

The script attempts to match the retail strategy (merch group table), merchandise group (merch group member table) and retail level (retail level table). If the script does not find a match for all items in the worktable, the missing items will be displayed. If all items are matched, two possible messages will be displayed “Retail Items Verified” with a count of the number of retail items verified and “Not Sold Items Verified” with a count of the number of not sold items verified.

The script will return any missing merch groups or merch group members. If either of these are missing, the information should be added via the user interface. There is no need to re-import the items.

### Processing Method

This script is wrapped in a transaction, by default the script rolls back the transaction at the end. Although it is not necessary to process it this way, the ideal behind this is to resolve all the errors before committing the changes to the worktable. It will also work by committing the transaction each time. The key point is to keep running the script until all the items and strategies are resolved and nothing is missing. Only after this, should the next step be processed.

## Repair Strategies

After the resolve id step has been completed, the next step is to run the script named “03 - ItemRepairStrategy.sql”. The prior step determined the actual strategy and merchandise group according to the spreadsheet and it ensured that the spreadsheet values are in the database. Th repair script will synchronize the id values based upon the spreadsheet with what is in the database.

It will return 3 results.

* 999, Changed Strategies
* 999, Changed Merch Groups
* 999, Changed Retail Levels

Note: 999 indicates the number of changes, it is ok if a 0 is returned.

## Repair Manufacturers

This step will ensure that the manufacturer that is in the spreadsheet matches what is in the database. It will also create any manufacturers that are in the spreadsheet but not in the database. The script is named “04 - ItemRepairManufacturer.sql.”

## Review the Results

The script “05 - ItemRepairGetResults.sql” can be used to view the complete list of what changes were made. It does not modify any data.