# Purpose

In order to test forecasting, a representative set of data for product sales and receiving is required. Because this is cumbersome and difficult to create manually a process has been created to copy sales and invoice data from a production environment and load the data into a test system. This Document describes the required steps to extract the data from the production system and import it into the test system.

# Prepare the Sales Data

## Obtain the files

* Capture the XML files from a product system for the following. Three types of files are required:
  + DM\_RadXML\_Summary\_Import xxx.xml (multiple files per bu/day)
  + DM\_RadXML\_Summary\_EODFinal xxx.xml (multiple files per bu/day)
  + DM\_RadXML\_BUStatus\_Import xxx.xml (one file per bu/day)
* Note: In the actual files the “xxx” will be a random alpha numeric sequence to enforce file name uniqueness.
* The folders are stored in business unit and date folders in the BC/ESO file share server in any location. For a product or test system the structure looks like this:
* [\\emcwd1452.mgroupnet.com\ESOShare\AsynchQueueStore\spwyeso\10000001\10001164\04-22-2019\](file:///\\emcwd1452.mgroupnet.com\ESOShare\AsynchQueueStore\spwyeso\10000001\10001164\04-22-2019\)
* 10000001 is the client id.
* 10001164 is the business unit id.
* 04-22-2019 is the date.
* Copy these folders from the import export server to a working directory.

## Generate Id Replacement Mapping File for Sales Items

In order to load the sales data, the id values for sales items and employees much be changed to match the destination system.

* The first step to accomplish this is to create and file that contains the mapping values between the two systems. In the Data Extraction and Loading folder Program Files Subdirectory there is a folder named “XX - Replace Id Values in XML Upload Files”. It contains three SQL scripts.
  + Step 1 – RMI Map Extract Source DB.sql
  + Step 2 – RMI Map Insert Dest DB.sql
  + Step 3 – RMI Map Update ESO ID.sql
* Run the scripts in order.
  + Step 1 must be executed in the source BC/ESO database. It will create a table named bcssa\_custom\_integration..bc\_extract\_rmi\_bc\_eso\_map.
  + Step 2. Using the SQL Server management studio, extract the data from the table into a .sql file and name the sql file “Step 2 – RMI Map Insert Dest DB.sql”. Execute this sql in the destination BC/ESO database.
  + Step 3 must be executed in the destination BC/ESO database. It will populate a table named bc\_extract\_rmi\_bc\_eso\_map. Save this table as a comma delimited file to the Data Extraction and Loading folder “Import Files\Replace Id Values in XML Upload Files\Mapfiles\” Subdirectory using the file name BCtoESO - RMI Mapping.csv.

## Change the Id value for the Employee

Because it is not possible to map the employees directly, all source system employees will be mapped to a single destination system employee. Within the destination system find any employee id value. In the Data Extraction and Loading folder “Program Files\XX - Replace Id Values in XML Upload Files” there is a JavaScript file named ReplaceIdValuesInXMLUploadFiles. Find the line var sEmplReplaceTag = "<EmployeeID>xxx</EmployeeID>". Replace the xxx with the employee id value from the destination system. Save the file.

## Replace the Id Values

* Take the DM\_RadXML files for a single day and place the files in the “Import Files\Replace Id Values in XML Upload Files\SourceXML\10001164\04-22-2019” folder.
  + 10001164 is the business unit id.
  + 04-22-2019 is the date.
  + The business unit id and date need to match the structure of the source files.
* Two other directory structrues must be created. For example: “Import Files\Replace Id Values in XML Upload Files\SourceXML\Replaced\10001164\04-22-2019” and . “Import Files\Replace Id Values in XML Upload Files\SourceXML\Replaced\10001164\04-22-2019\Errors\”
  + 10001164 is the business unit id.
  + 04-22-2019 is the date.
  + The business unit id and date need to match the structure of the source files.
* Note: The file structure is a recommendation/exmple, any folder structure can be used if it separates the files by business unit and date.
* To process the files correctly, each location where files are placed must be directly added to the JScript code. There is a function called to processSourceXMLFiles. The first 3 parameters represent the location of the source files, the location to place the replaced files, and the location to place any error files. Do not change the remaining parameters.

// Process Files

processSourceXMLFiles(

"C:\\Test Replace Id Values in XML Files\\1000008\\05-22-2019\\",

"C:\\Test Replace Id Values in XML Files\\1000008\\Replaced\\05-22-2019\\",

"C:\\Test Replace Id Values in XML Files\\1000008\\Replaced\\05-22-2019\\Errors\\",

array\_bc, array\_eso, array\_regexp, numArrayElements);

* + Note: This function call can be copied and edited to process multiple directories at a time.
* Run the command file Step 1 - Replace Id Values.cmd in Data Extraction and Loading folder “Program Files\XX - Replace Id Values in XML Upload Files.
* After the command has been completed, new files will be generated in the “Import Files\Replace Id Values in XML Upload Files\ReplacedXML” folder. Addition files may be placed in the ErrorXML folder, these can be used for diagnostic purposes.
* Move the replaced files to the processing folder of the destination system. This will be similar to the server name and file path that the original files were copied from, however the server and business unit will likely be different. for example:
* [\\emcwt1452.mgroupnet.com\ESOShare\AsynchQueueStore\spwyeso\10000001\10004929\04-22-2019\](file:///\\emcwt1452.mgroupnet.com\ESOShare\AsynchQueueStore\spwyeso\10000001\10004929\04-22-2019\)
* Repeat this process for every business unit and day files that are required.

# Prepare the Receiving Data

## Obtain the files

* Ensure that the Data Extraction and Loading folder structure in installed in a location that has DB server access.
* In the Data Extraction and Loading folder “Program Files\XX - Receiving To Invoice” open the sql file “ReceivingToInvoiceCountForExtract.sql”. There are two values to be set. First there is a variable that indicates the number of days of invoices to extract “SET @DaysBack = -30”. The default is 30 but can be changed to any value. Second there is a list of business unit codes at the bottom of the file:

AND rsda.name IN (

'0004540')

* Add additional business unit codes to the in clause.
* Save the file.
* In the Data Extraction and Loading folder “Program Files\XX - Receiving To Invoice”, run the command file cmdStep1 - DB - To - FinalCSV\_Prod. Note, this file contains a specific server name, so it may need to be edited to a different server name.
* CSV files will be produced in the “Import Files\XX - Receiving To Invoice\FinalCSV” folder.
* In the Data Extraction and Loading folder “Program Files\XX- Receiving To Invoice”, run the command file cmdStep2 - FinalCSV - To - FinalXML.
* Final XML files will be produced. The name of each file indicates the business unit and business date. This is important for later processing.

# Processing the Files

The following steps must be executed in the correct sequence. Ideally this should be done on business units with no other prior activity.

## Set the starting business date

Each business unit must have an initial open business date. To create this, perform the following steps:

* Open the file In the Data Extraction and Loading folder “Program Files\XX - Process Sales and Receiving Test Data” named Close and Increment Business Day.sql”
* Change the values for the Date to Close and the list of business unit codes. Note, the “DayToClose” value needs to be one day prior that the required open business date:

SET @DayToClose = '2019-04-19'

INSERT @BusinessUnitList (

BusinessUnitCode,

BusinessUnitId)

SELECT name,

data\_accessor\_id

FROM rad\_sys\_data\_accessor

WHERE name IN (

'0004540')

* Save the file.

## Load Sales for the business date

The sales must be loaded one day and business unit at a time.

* Get the file names for the replaced files created in the prepare step for the business unit and open business date. This can be done by using the “dir” command with the “/w” option and directory the output to a text file. For example, “Dir \*.xml /w > files.txt” will produce a list of files in the directory.
* In order to process the files, the complete path and business unit id must be specified. There is a spreadsheet in the “XX - Process Sales and Receiving Test Data” named “Set full path for import of RAD XML files” to assist with this. Copy the file names to column B and the complete path will be calculated. The date and business unit values in the formula will also need to be changed.
* In the “XX - Process Sales and Receiving Test Data” folder there is a sql file named “Submit APE jobs for Sales Upload.sql”. Change the sql to insert the business unit id and fully qualified file path for each file that needs to be processed:

DECLARE @TableList TABLE (

SequenceNumber INT IDENTITY(1,1),

BusinessUnitId INT,

FileName NVARCHAR(255) NOT NULL)

INSERT @TableList (BusinessUnitId, FileName) SELECT 10001164,'\\emcwd1452.mgroupnet.com\ESOShare\AsynchQueueStore\spwyeso\10000001\10001164\04-22-2019\DM\_RadXML\_Summary\_Import 68409D6C-C846-490D-BF77-F1C6C400D4F3.xml'

INSERT @TableList (BusinessUnitId, FileName) SELECT 10001164,'\\emcwd1452.mgroupnet.com\ESOShare\AsynchQueueStore\spwyeso\10000001\10001164\04-22-2019\DM\_RadXML\_Summary\_Import 5EFEBABD-A82B-44E9-B25E-358007180690.xml'

INSERT @TableList (BusinessUnitId, FileName) SELECT 10001164,'\\emcwd1452.mgroupnet.com\ESOShare\AsynchQueueStore\spwyeso\10000001\10001164\04-22-2019\DM\_RadXML\_Summary\_Import 51B766C6-798F-47D3-AC34-079D39D2DCA1.xml'

INSERT @TableList (BusinessUnitId, FileName) SELECT 10001164,'\\emcwd1452.mgroupnet.com\ESOShare\AsynchQueueStore\spwyeso\10000001\10001164\04-22-2019\DM\_RadXML\_BUStatus\_Import 07085A9-0FBC-84BD-9B0E-6FED99BDB51F.xml'

* Save the file.
* Open the file in a query window in the destination database and execution the sql. This process will take as long as 10 minutes to complete. Verify that item sales have been loading by running the Sales Mix report in the destination BC/ESO system. If the sales mix does not reflect new data, check the APE queued job viewer for errors.

## Load Invoices for the business date

In order to have accurate forecast test data receiving documents are needed. In order to produce these, invoices are extracted from the source BC/ESO system and imported into the destination BC/ESO system. There is a flag setting that will automatically create receiving documents based upon invoices, this can be used to create receiving via the invoice import.

Perform the following steps:

* In the data extraction and loading folder there is a “XX - Set Create Receiving From Invoice to True” there is a file named “Set Create Receiving From Invoice to True.sql”. Execute this file in the destination BC/ESO database. This will save the current value so it can be restored later after all files have been processed. (This only needs to be done one time until all files are processed).
* Take the Final invoice XML files produced in the prepare step for the open business day and process the files using the standard BC/ESO file import process for invoices.
* Verify that draft receiving documents were created for the business units using the BC/ESO user interface. If no files are present check the APE queued job viewer and BC/ESO email for errors.
* In the “XX - Process Sales and Receiving Test Data” folder there is a sql file named “05 - Insert Missing Cost Rows.sql”. Open the file and change the business date value and set the business unit codes to process.

SET @BusinessDate = '2019-04-20'

…

FROM rad\_sys\_data\_accessor

WHERE name IN (

'0004540')

* Execute the script
* Typically, this script will return zero rows affected or only a few rows affected.

## Resolve Missing Cost Issue

The end of day will not close properly if there are depletion items that are missing costs. This step will resolve the issue.

* In the “XX - Process Sales and Receiving Test Data” folder there is a sql file named “Resolved Missing Costs. Sql”. Open the file and change the day to close date value and set the business unit codes to process.

SET @DayToClose = '2019-04-20'

…

FROM rad\_sys\_data\_accessor

WHERE name IN (

'0004540')

## Process the End of Day

After Sales and Receiving Document have been loaded, the eod of day can be processed.

* In the “XX - Process Sales and Receiving Test Data” folder there is a sql file named “Post Inventory Day.sql”. Open the file and change the day to close date value and set the business unit codes to process.

SET @DayToClose = '2019-04-20'

…

FROM rad\_sys\_data\_accessor

WHERE name IN (

'0004540')

* Execute the script and check the Queued Job viewer for errors. The jobs will take several minutes to complete. The job will not change the business day.

## Move to the next Business Day

After the end of day jobs have been executed, the business day must be closed:

* Open the file In the Data Extraction and Loading folder “Program Files\XX - Process Sales and Receiving Test Data” named Close and Increment Business Day.sql”
* Change the values for the Date to Close and the list of business unit codes. Note, the “DayToClose” value needs to be one day prior that the required open business date:

SET @DayToClose = '2019-04-19'

INSERT @BusinessUnitList (

BusinessUnitCode,

BusinessUnitId)

SELECT name,

data\_accessor\_id

FROM rad\_sys\_data\_accessor

WHERE name IN (

'0004540')

* Save the file.
* Open the file in a query window in the destination database and execution the sql.

## Complete the processing

Repeat all the steps for each business unit and day except for the steps to create the initial adjustment and to set the flag to create receiving documents from invoices. After all files are processed, perform the step to restore the create receiving flag:

* In the data extraction and loading folder there is a “XX - Reset Create Receiving From Invoice Flag” there is a file named “Reset Create Receiving From Invoice Flag.sql”. Execute this file in the destination BC/ESO database. This will restore the prior after all files have been processed.

# Configuring the Forecast

These instructions were not needed, however the screen shots for the existing configuration are provided for reference.











