****

Checkbook NYC 4.0

M/WBE & Sub Vendors

ETL Implementation Approach

version 0.1

October 24, 2014

**DOCUMENT INFORMATION AND APPROVALS**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Date** | **Author** | **Version** | Reviewer | Position | Version approved |
| 10/24/2011 | TirupatiReddy Manyam | 0.1 |  | Principal Engineer | 0.1 |

**Table of Contents**

[Background](#_Toc343697424) 4

[Scope 4](#_Toc343697425)

[Overview 4](#_Toc343697426)

[ETL Process 4](#_Toc343697428)

M/WBE [ETL Process 5](#_Toc343697428)

[Update Spending (FMS) Data 5](#_Toc343697435)

[Update Contracts (CON & MAG) Data 5](#_Toc343697436)

[Refresh Aggregate Data 6](#_Toc343697436)

Sub Vendor [ETL Process 7](#_Toc343697428)

[Process Sub Contract business type 7](#_Toc343697436)

[Process Sub Contract Details 7](#_Toc343697437)

[Process Sub Contract Spending 8](#_Toc343697438)

[Appendix A](#_Toc343697449) 10

[SV data validations](#_Toc343697451) 10

[SC data validations](#_Toc343697452) 11

[SF data validations](#_Toc343697453) 12

## Background

Checkbook NYC 4.0 is a major enhancement to Checkbook NYC 3.0 by adding M/WBE and Sub Vendors feature dashboards.

## Scope

This document provides an insight into the ETL Implementation Approach for M/WBE and Sub Vendors enhancements in Checkbook NYC 4.0. ETL process starts with the consumption of raw data, validating, processing and posting it to the transaction tables.

## Overview

Kettle, Pentaho data integration tool will be used to integrate the various ETL processes to fetch files from remote server(s), loading data to database from flat files, data cleansing etc.

Master kettle job will be created which has several transformations and invokes several other jobs. This will be started around 11 PM when the files are published by FISA. All the source files from FISA are UTF-8 complaint. Processing of the data begins only when the required files are available.

## ETL Process

ETL process starts with the directory listing of FISA server and once the required details are obtained, each of the data file is processed & changes are posted in the transaction table in the order of COA/ FMSV, Revenue/Budget/MAG, CON, FMS/PMS, SubContract, SubSpending.

Various transformations in the master job are explained in this section. If any error occurs in a transformation a DB function is executed by passing the load file identifier as a parameter to determine if ETL job can continue with the next data file or it has to abort the entire job.

* If any of the COA data file result in an error the entire job is aborted. Processed\_flag is set to E for the corresponding file and all data files which had to be consumed will be set to C for cancelled.
* For any feed other than COA only the non-processed files of the specific feed are cancelled. For example if FMS data file resulted in an error, processed\_flag is set to E for the corresponding file and all other FMS data files which had to be consumed will be set to C for cancelled.

##### 

## 

## M/WBE ETL Process

In this we will categorize the data based on vendor minority category.

##### Update Spending (FMS) data

1. In this the ETL Script will update the spending transactions and payroll summary transactions that are inserted/updated in the current data load.
2. Update payroll summary transactions minority category as “Individuals & Others”.
3. Update spending transactions with the latest minority category of the associated vendor.
4. Update spending transactions which are not categorized into any minority category and the associated contract’s agreement type code is one of ('35', '36', '39', '40', '44', '65', '68', '79', '85') to “Individuals & Others”.
5. Update spending transactions which are not categorized into any minority category and the associated contract’s award method code is one of ('07', '08', '09', '17', '18', '44', '45', '55') to “Individuals & Others”.
6. Update spending transactions which are not categorized into any minority category and vendor organization classification code is one of (1, 5, 6, 7, 8, 9, 12, 13, 14, 15, 19, 22, 23, 24, 25, 26, 27, 28, 29, 30, 31, 32, 33, 34, 35, 36) to “Individuals & Others”.
7. Update spending transactions which are not categorized into any minority category to “Non-Minority”.
8. For one time as Checkbook started getting data from 01/13/2013, update the minority category of the spending transactions whose check issued date is before 01/13/2013 using the M/WBE Application (<http://mwbereportcardnyc.com/>) data.

##### Update Contracts (CON & MAG) data

1. In this the ETL Script will update the contract transactions that are inserted/updated in the current data load.
2. Update contract transactions with the latest minority category of the associated vendor.
3. Update contract transactions which are not categorized into any minority category and the agreement type code is one of ('35', '36', '39', '40', '44', '65', '68', '79', '85') to “Individuals & Others”.
4. Update contract transactions which are not categorized into any minority category and the award method code is one of ('07', '08', '09', '17', '18', '44', '45', '55') to “Individuals & Others”.
5. Update contract transactions which are not categorized into any minority category to “Non-Minority”.
6. Below are the agreement type codes used to categorize into “Individuals & Others”.

|  |  |
| --- | --- |
| Agreement Type  Code | Agreement Type Name |
| 35 | LESSEE |
| 36 | MISCELLANEOUSPROPERTYRENTAL |
| 39 | LESSOR - REVENUE |
| 40 | LESSOR - ACCOUNTING LINES EXIST |
| 44 | PUBLICUTILITY |
| 65 | LOANS |
| 68 | FORCEACCOUNTAGREEMENT |
| 79 | WATERSHEDLANDACQUISITION |
| 85 | INTRA-AGENCYFUNDAGREEMENTS |

1. Below are the award method codes used to categorize into “Individuals & Others”.

|  |  |
| --- | --- |
| Agreement Type  Code | Agreement Type Name |
| 07 | LESSEE NEGOTIATION |
| 08 | LOAN NEGOTIATION |
| 09 | RENTAL SUBSIDY NEGOTIATION |
| 17 | GOVERNMENT TO GOVERNMENT |
| 18 | NON PROCUREMENT TRANSACTION |
| 44 | PUBLIC UTILITY |
| 45 | SMALL PURCHASE-PUBLIC UTILITY |
| 55 | NO LONGER VALID - CNV |

##### Refresh Aggregate Data

Once all the M/WBE category updates are done to spending, payroll summary and contract transactions, all M/WBE contracts and spending aggregate tables should be refreshed. ETL script will refresh table by table, first by truncating the data in the table and then inserting the aggregate data back using the queries on the respective transaction tables.

## 

## Sub Vendor ETL Process

Sub Vendors data will be processed after all COA, CON, MAG and FMS feeds are processed. The processing order of sub vendors data is Sub Vendor business types, Sub Contract details and Sub Spending feeds. Sub Vendor related aggregate tables will be refreshed after all sub vendors data is processed.

##### Process Sub Contract business type

1. The file name pattern for sub contract business type data is "AIE((G\_DLY\_SCNTRC\_)|(H\_DLY\_SCNTRC\_))BTY\_20[0-9][0-9][0-1][0-9][0-3][0-9][0-2][0-9][0-5][0-9][0-5][0-9].txt"
2. The data from sub contract business type file from FMS is loaded into staging table stg\_scntrc\_bus\_type.
3. The data loaded into stg\_scntrc\_bus\_type is validated based on the validation rules (Appendix A : SV data validations). The invalid\_flag column is updated to ‘Y’ if the record is invalid.
4. All the data in staging table (stg\_scntrc\_bus\_type) is copied to archive table (archive\_scntrc\_bus\_type) with the current load file identifier.
5. All the invalid records from staging table (stg\_scntrc\_bus\_type) are copied to invalid table (invalid\_scntrc\_bus\_type) with the current load file identifier. And those invalid records will be deleted from the staging table.
6. Set action\_flag to ‘U’ for update if the data in the staging table identified by vendor customer code, business type, minority type and status fields exist in the transaction table subcontract\_vendor\_business\_type.
7. Update the data in the transaction table subcontract\_vendor\_business\_type using staging table where action\_flag is ‘U’
8. Insert the data in the transaction table subcontract\_vendor\_business\_type for all new rows in staging table identified by action\_flag ‘I’.

##### Process Sub Contract details

1. The file name pattern for sub contract details data is "AIE((G\_DLY\_SCNTRC\_)|(H\_DLY\_SCNTRC\_))DET\_20[0-9][0-9][0-1][0-9][0-3][0-9][0-2][0-9][0-5][0-9][0-5][0-9].txt"
2. The data from sub contract details file from FMS is loaded into staging table stg\_scntrc\_details.
3. The data loaded into stg\_scntrc\_details is validated based on the validation rules (Appendix A : SC data validations). The invalid\_flag column is updated to ‘Y’ if the record is invalid.
4. All the data in staging table (stg\_scntrc\_details) is copied to archive table (archive\_scntrc\_details) with the current load file identifier.
5. All the invalid records from staging table (stg\_scntrc\_details) are copied to invalid table (invalid\_scntrc\_details) with the current load file identifier. And those invalid records will be deleted from the staging table.
6. Update source updated date (doc\_appl\_last\_dt) column to current date to all the records in the staging table.
7. Update registration date column in the staging table based on the prime contract registration date which means all the sub contracts of a specific prime contract will have the same registration date.
8. FK fields such as document\_code\_id, agency\_id, are updated in the staging table stg\_scntrc\_details from the corresponding reference tables for the columns namely doc\_cd, doc\_dept\_cd.
9. Identify the vendors to be updated based on scntrc\_vend\_cd column and update the vendor information in transaction table subvendor. And for all the new vendors insert the data in transaction table subvendor.
10. Identify the records to be updated in sub contract transaction table subcontract\_details using the fields doc\_id, doc\_dept\_cd, doc\_cd, scntrc\_id and scntrc\_vers\_no and update the action\_flag to ‘U’ for update.
11. Update all the records in the transaction table subcontract\_details with the data in staging table identified by action\_flag ‘U’ .
12. Insert all the records in the transaction table subcontract\_details with the data in staging table identified by action\_flag ‘I’ .
13. Update original contract amount of all versions of a sub contract with the current amount of the original version of the sub contract.
14. Update Award method code, agreement type code, award category code fields using the data from prime contract.
15. Update original\_version\_flag and latest\_flag based on scntrc\_vers\_no field. The sub contract with least scntrc\_vers\_no will have original\_version\_flag as ‘Y’ and the sub contract with highest scntrc\_vers\_no will have latest\_flag as ‘Y’.

##### Process Sub Contract Spending

1. The file name pattern for sub contract spending data is "AIE((G\_DLY\_SCNTRC\_)|(H\_DLY\_SCNTRC\_))PMT\_20[0-9][0-9][0-1][0-9][0-3][0-9][0-2][0-9][0-5][0-9][0-5][0-9].txt"
2. The data from sub contract details file from FMS is loaded into staging table stg\_scntrc\_pymt.
3. The data loaded into stg\_scntrc\_pymt is validated based on the validation rules (Appendix A : SF data validations). The invalid\_flag column is updated to ‘Y’ if the record is invalid.
4. All the data in staging table (stg\_scntrc\_pymt) is copied to archive table (archive\_scntrc\_pymt) with the current load file identifier.
5. All the invalid records from staging table (stg\_scntrc\_pymt) are copied to invalid table (invalid\_scntrc\_pymt) with the current load file identifier. And those invalid records will be deleted from the staging table.
6. FK fields such as document\_code\_id, agency\_id, are updated in the staging table stg\_scntrc\_pymt from the corresponding reference tables for the columns namely doc\_cd, doc\_dept\_cd.
7. Associate the sub contract identifier to sub spending transaction based on columns con\_document\_code, con\_agency\_code, con\_document\_id and con\_sub\_contract\_id.
8. Identify the vendors to be updated based on scntrc\_vend\_cd column and update the vendor information in transaction table subvendor. And for all the new vendors insert the data in transaction table subvendor.
9. Associate the spending transaction to the above subvendor identifier based on scntrc\_vend\_cd column.
10. Identify the records to be updated based on the columns doc\_cd, doc\_dept\_cd, doc\_id, scntrc\_id and scntrc\_pymt\_id and update action\_flag to ‘U’ for update.
11. Update the spending transactions in transaction table subcontract\_spending using the staging table stg\_scntrc\_pymt for which action\_flag is ‘U’.
12. Insert the spending transactions in transaction table subcontract\_spending using the staging table stg\_scntrc\_pymt for which action\_flag is ‘I’.
13. Update the minority category of the sub spending transaction with the latest minority category of associated sub vendor.

## 

## Appendix A

#### SV data validations

|  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **SubVendor** | | | | | | | | | | |
| **Type of feed** | | Daily – Incremental | | | | | | | | |
| **File Name Pattern** | | AIE((G\_DLY\_SCNTRC\_)|(H\_DLY\_SCNTRC\_))BTY\_20[0-9][0-9][0-1][0-9][0-3][0-9][0-2][0-9][0-5][0-9][0-5][0-9].txt (AIEG\_DLY\_SCNTRC\_BTY\_20141101010152.txt) | | | | | | | | |
| **Number of files to be processed** | | Multiple files processed in the order in which it was created in FTP server. | | | | | | | | |
| **Tables** | | | | | | | | | | |
| **Entity Name** | **Entity Identifier** | | **Unique elements** | **Staging Table** | | **Transaction Table** | | | **History Table** | |
| Sub vendor business type | NA | | doc\_cd  doc\_dept\_cd  doc\_id  scntrc\_id  scntrc\_vend\_cd  bus\_typ | etl. stg\_scntrc\_bus\_type | | subcontract\_vendor\_business\_type | | | NA | |
| **Validation Rules** | | | | | | | | | | |
| **Entity Name** | **Missing Key Values** | | **Duplicate** | | **Invalid/Inconsistent Values** | **Missing Parent/Child Entity** | **Associated to invalid entities** | **Multiple records** | | **Inter-Load duplicates** |
| Sub vendor business type | 1.doc\_cd  2.doc\_dept\_cd  3.doc\_id  4. scntrc\_id  5. scntrc\_vend\_cd  6. bus\_typ  7. cert\_strt\_dt | | 1.doc\_cd  2.doc\_dept\_cd  3.doc\_id  4. scntrc\_id  5. scntrc\_vend\_cd  6. bus\_typ  7. min\_typ | | 1. Invalid business type ( Not in ref\_business\_type )  2. Invalid business type status( Not in ref\_business\_type\_status) |  |  |  | |  |

#### SC data validations

|  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **SubContract** | | | | | | | | | | |
| **Type of feed** | | Daily – Incremental | | | | | | | | |
| **File Name Pattern** | | AIE((G\_DLY\_SCNTRC\_)|(H\_DLY\_SCNTRC\_))DET\_20[0-9][0-9][0-1][0-9][0-3][0-9][0-2][0-9][0-5][0-9][0-5][0-9].txt (AIEG\_DLY\_SCNTRC\_DET\_20141101010152.txt) | | | | | | | | |
| **Number of files to be processed** | | Multiple files processed in the order in which it was created in FTP server. | | | | | | | | |
| **Tables** | | | | | | | | | | |
| **Entity Name** | **Entity Identifier** | | **Unique elements** | **Staging Table** | | **Transaction Table** | | | **History Table** | |
| Sub vendor Contracts | NA | | doc\_cd  doc\_dept\_cd  doc\_id  scntrc\_id  scntrc\_vers\_no | etl. stg\_scntrc\_details | | subcontract\_details  sub\_agreement\_snapshot  sub\_agreement\_snapshot\_cy | | | NA | |
| **Validation Rules** | | | | | | | | | | |
| **Entity Name** | **Missing Key Values** | | **Duplicate** | | **Invalid/Inconsistent Values** | **Missing Parent/Child Entity** | **Associated to invalid entities** | **Multiple records** | | **Inter-Load duplicates** |
| Sub vendor Contracts | 1.doc\_cd  2.doc\_dept\_cd  3.doc\_id  4. scntrc\_id  5. scntrc\_vers\_no  6. scntrc\_strt\_dt  7. scntrc\_max\_am | | 1.doc\_cd  2.doc\_dept\_cd  3.doc\_id  4. scntrc\_id  5. scntrc\_vers\_no | |  |  |  |  | |  |

#### SF data validations

|  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **SubSpending** | | | | | | | | | | |
| **Type of feed** | | Daily – Incremental | | | | | | | | |
| **File Name Pattern** | | AIE((G\_DLY\_SCNTRC\_)|(H\_DLY\_SCNTRC\_))PMT\_20[0-9][0-9][0-1][0-9][0-3][0-9][0-2][0-9][0-5][0-9][0-5][0-9].txt (AIEG\_DLY\_SCNTRC\_PMT\_20141101010152.txt) | | | | | | | | |
| **Number of files to be processed** | | Multiple files processed in the order in which it was created in FTP server. | | | | | | | | |
| **Tables** | | | | | | | | | | |
| **Entity Name** | **Entity Identifier** | | **Unique elements** | **Staging Table** | | **Transaction Table** | | | **History Table** | |
| Sub vendor Spending | NA | | doc\_cd  doc\_dept\_cd  doc\_id  scntrc\_id  scntrc\_pymt\_id | etl. stg\_scntrc\_pymt | | subcontract\_spending  subcontract\_spending\_details | | | NA | |
| **Validation Rules** | | | | | | | | | | |
| **Entity Name** | **Missing Key Values** | | **Duplicate** | | **Invalid/Inconsistent Values** | **Missing Parent/Child Entity** | **Associated to invalid entities** | **Multiple records** | | **Inter-Load duplicates** |
| Sub vendor Spending | 1.doc\_cd  2.doc\_dept\_cd  3.doc\_id  4. scntrc\_id  5. scntrc\_pymt\_id  6. scntrc\_pymt\_dt  7. scntrc\_pymt\_am | | 1.doc\_cd  2.doc\_dept\_cd  3.doc\_id  4. scntrc\_id  5. scntrc\_pymt\_id | |  |  |  |  | |  |