DSM User Guide

**Dec 15, 2014**

*This document has been prepared by Mu Sigma to outline the process for DSM Bi monthly run*

Contents

[Flow Diagram 3](#_Toc428802526)

[NOTE: The following SECTION‘s represent the different steps in the flow diagram. 4](#_Toc428802527)

[How to generate the analytical dataset 4](#_Toc428802528)

[Create folder structure 5](#_Toc428802529)

[Payment Vehicle Query 7](#_Toc428802530)

[Rundates 8](#_Toc428802531)

[Library Names 8](#_Toc428802532)

[Data pull 9](#_Toc428802533)

[NextGen Payment Details 9](#_Toc428802534)

[Next Gen Model attributes creation 11](#_Toc428802535)

[Next Gen Model Scoring 11](#_Toc428802536)

[IFM Part 1 12](#_Toc428802537)

[IFM Part 2 13](#_Toc428802538)

[Final IFM DSM Merging 14](#_Toc428802539)

[Check reconciliation status 15](#_Toc428802540)

[Run Address\_Match Code 15](#_Toc428802541)

[Run EFT Code 15](#_Toc428802542)

[Run MCO\_details code 16](#_Toc428802543)

[Run Oracle Name match code 16](#_Toc428802544)

[Run Oracle Name match SAS code 16](#_Toc428802545)

[Oracle Codes 17](#_Toc428802546)

[Data QC 17](#_Toc428802547)

[Tableau Extract 18](#_Toc428802548)

[Timeline for Bi Monthly refresh 21](#_Toc428802549)

# Flow Diagram

Vehicle payment Query

Basic preparation: Libname definition and update rundates

Issuer Title

Payee employee Name match

IFM part2

IFM part1

Model Variable creation

Oracle Data Pull

**IFM Dataset**

Next Gen Payment Details

Model Scoring

Final merge

Final renaming and dropping ofcolumnscolumns

Address Match on final table

Payee Payer Name Match

Issuer Title and ENAM Flag merge

Tableau Extract creation

# NOTE: The following SECTION‘s represent the different steps in the flow diagram.

# How to generate the analytical dataset

SAS Codes

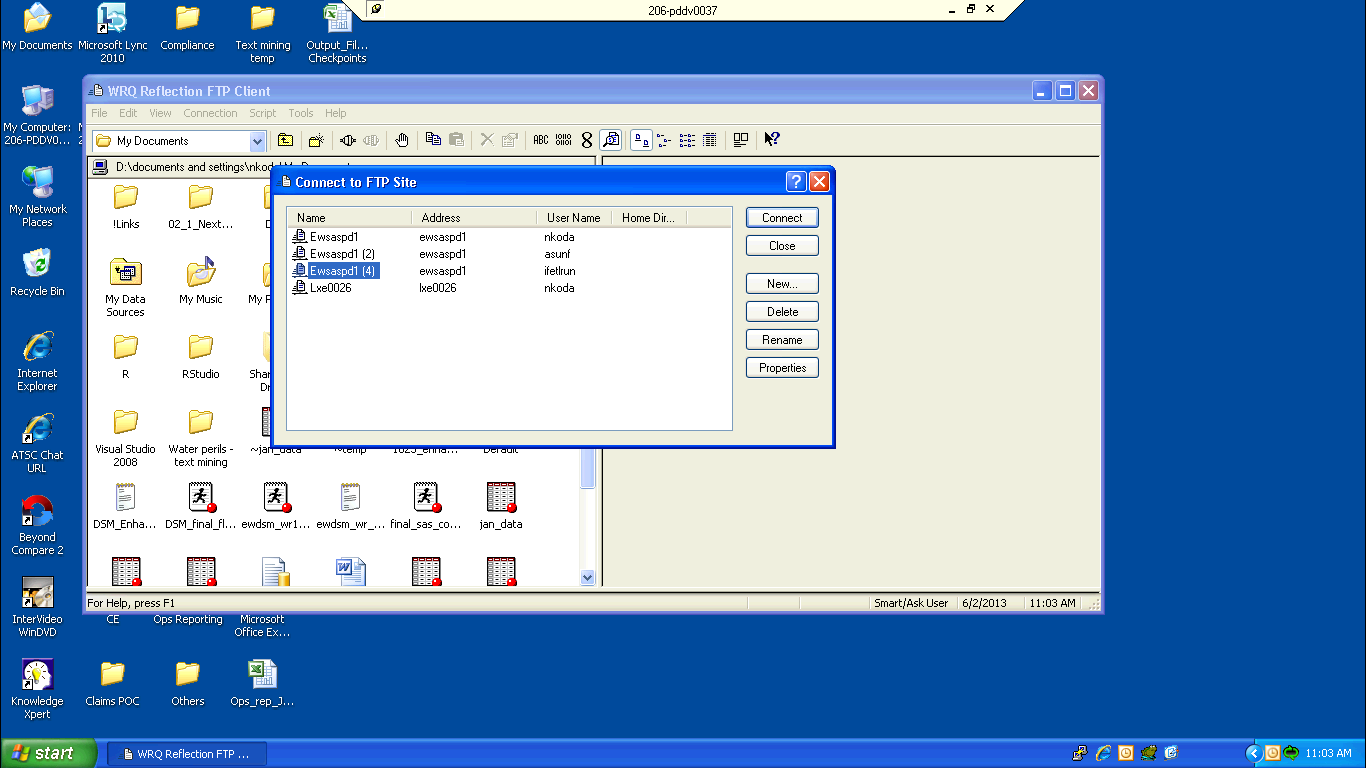
SECTION - 1

1. Connect WRQ FTP client server:

* + 1. WRQ reflection/utilities/FTP Client
    2. Username: ragav (Or username of person handling the DSM process)

NOTE: It is Rohit Agarwal’s id (non-generic) hence it may expire .So it should be taken care that the password for this id doesn’t expire prior to the Run)

* + 1. Password: aBCD1234(Unix password)
    2. Database: EWSASPD1



1. Now go to location : EWSASPD1/work/ew07/projects/dev/DSM\_Refresh\_Run

NOTE: The location where we create the folder structure may change based on the requirement

For example – If there is not enough space in ew07, then space can be created by deleting datasets from previous run’s

## ***Create folder structure***

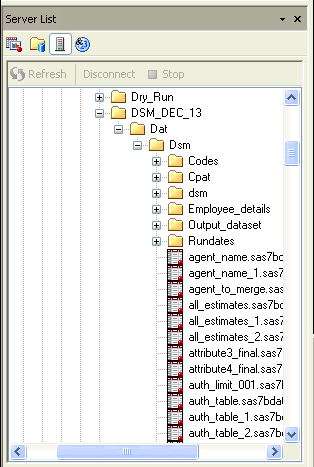
1. (Every month will have a new folder structure)

DSM\_OCT\_13

* Dat
  + Dsm
    - Codes
    - Cpat
    - Rundates
    - Output\_dataset
    - Employee\_details

**NOTE:** Copy rundates SAS datasets () from DSM\_OCT\_13 to the current rundates folder. (We can directly copy and paste from one folder to another folder here)

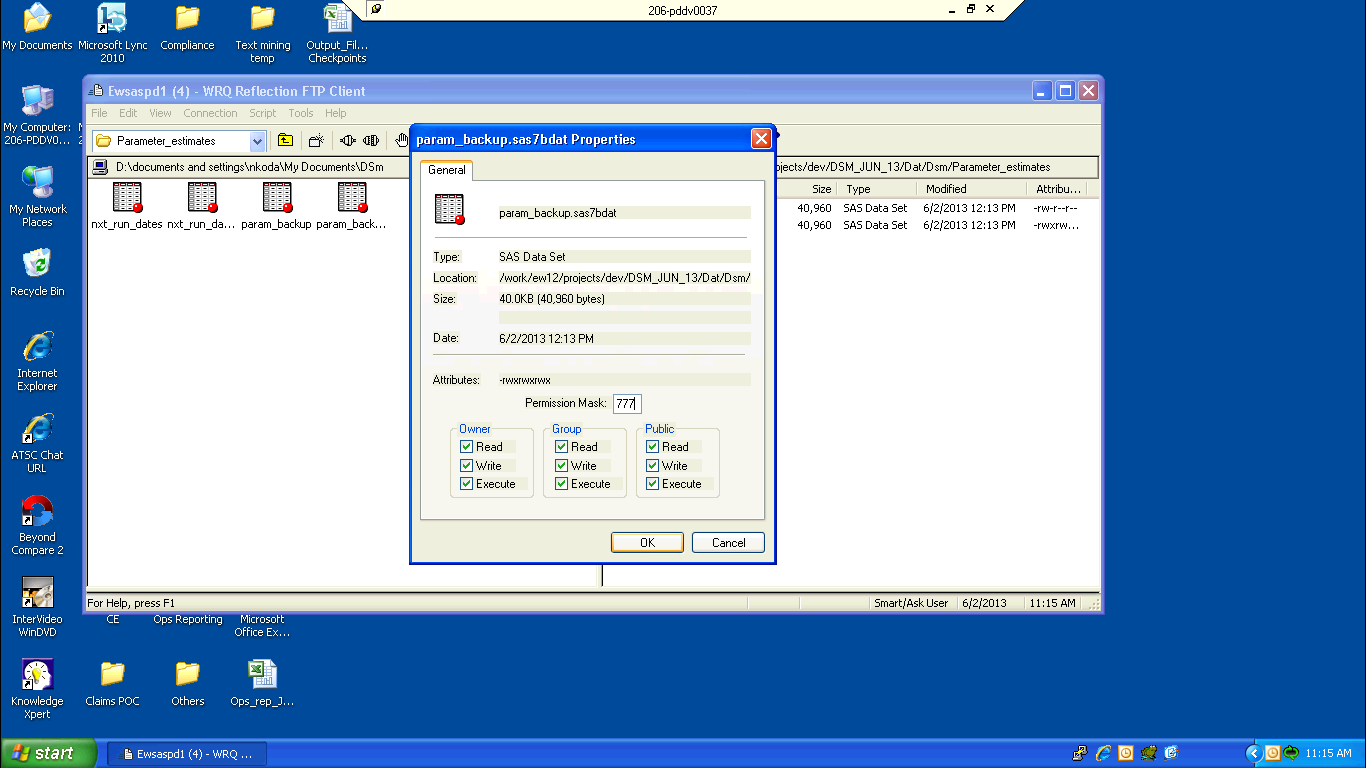
Screenshot of the folder structure in SAS –



**Checkpoints:** All the folders must be created with the exact names considering also the caps and underscore as the exact names are further used in defining libnames

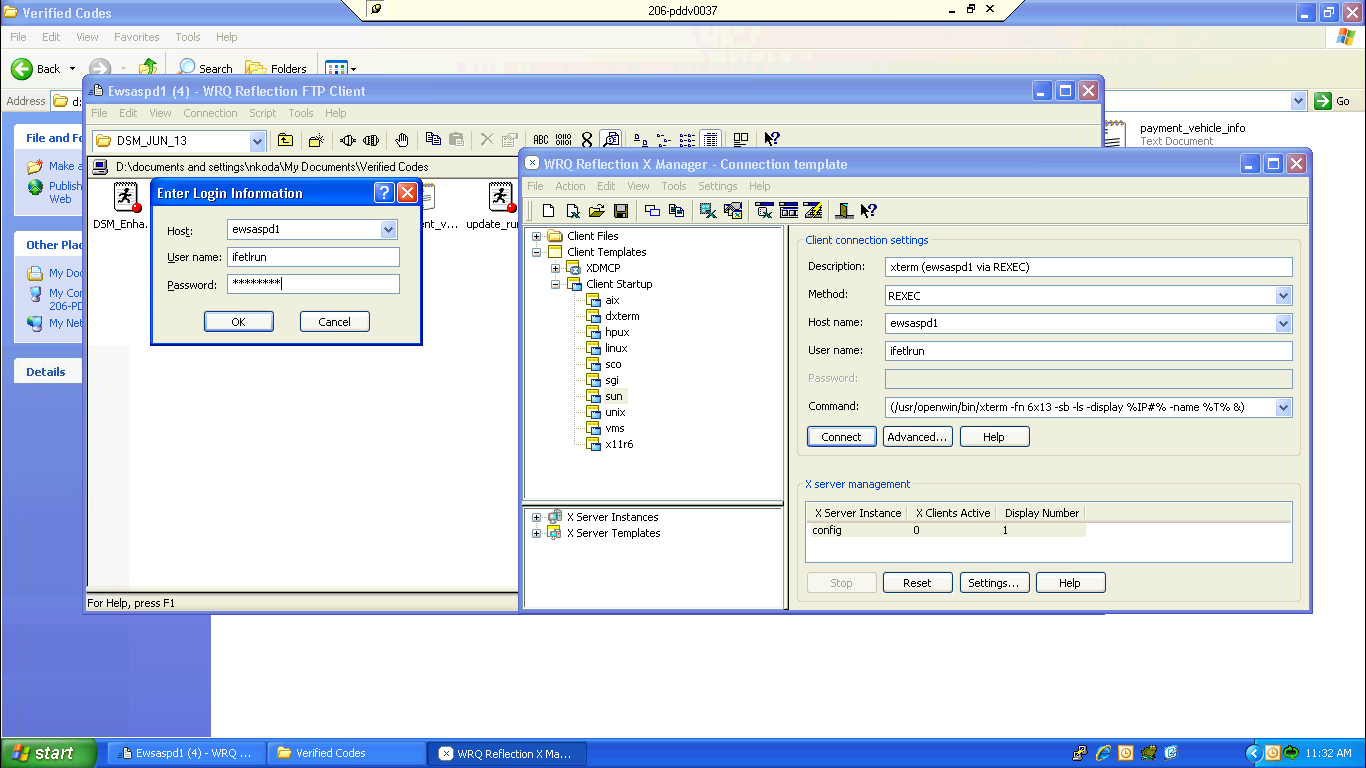
1. Assign Global access to each folder and dataset for the current run.
2. Process to assign access to each folder and file-

Go to each folder/file right click on it .Click on properties .Change the value of Permission mask to 777.The screen shot for the same is below.



1. Open WRQ REFLECTION X
2. WRQ REFLECTION X opens the UNIX terminal through which we can run our SAS codes on the UNIX server.
3. Credentials to log into are : Username –ragav(username of person handling the DSM process) and Password –aBCD1234

The screen shot below gives a better picture of how to connect to WRQ Reflection



SECTION - 2

## Payment Vehicle Query

8) Run the payment vehicle query embedded below in **oracle**.

**NOTE:** To run oracle queries open ORACLE SQL DEVELOPER in Rohit Agarwal’s machine



Time taken: 2-4 hours.

**Checkpoints:**

* Update the date conditions in the query according to the current **Run Dates**.
* Example of run dates for October run will be: 01/10/11 to 30/09/13**. (Last 2 years from date of run).**
* We run the entire model for the last 24 months data, i.e. we score all the payments in the last two years from the current run date .Hence we need to pull all the data including the vehicle information which we are pulling in this code for last 24 months in every run.

**Code Description:** This code pulls the data for the various vehicles involved in the claims process for this period. Mainly the columns pulled are vehicle model, vehicle model year, vehicle make name at the payment detail level.

* Query to check number of rows and min ,max date in oracle table:

Select count(\*) from dsm\_pymnt\_vehicle\_info\_final

DESIRED RESULT ~ 27.5M -28.5M .We generally get approximately 1-1.5M records monthly.

* Ensure the schema in which the query is run is **ragav(person handling DSM process)** as of now.

## Rundates

9) Run the **update\_rundates** code on SAS server

***NOTE:*** *All the codes including this one are embedded in this document and need to be copied to the Codes folder in the folder structure before the run*

Update the date conditions in the query according to the current **Run Dates**. The date, month and

Year needs to be updated in the query according to the run dates manually.

For example: run dates for October run will be 01/10/11 to 30/09/13**. (Last 2 years from date of run)**



**Code description**: It updates the rundates in SAS dataset to current rundates to be used.

Checkpoint: Ensure that the output dataset contains the required run dates

For example: The run dates for the DSM October 2013 run should be 1/10/11 to 30/09/13.

Time taken: 2 minutes.

## Library Names

10. Run the 00\_LibNames.sas

Code description: Contains the definition of all the libnames defined in all other codes as this file is called in each piece of code**. Change the Lib paths as per the current run**



**Checkpoint:**

* In the output we get the from\_date and to\_date in integer form and

then convert them to date format in excel by adding date-01/01/1960 to these integers.

Result of this code gives the run dates in integer form .Copy the integer values to excel and add them to date (01/01/1960) will give the run dates in date format which should match the dates manually entered.

* For example :

For December run dates in integer format are 18962 to 19692 which convert to 01/12/11 to 30/11/13 by adding 01/01/1960 to integer format in excel.

Excel command – 18962 + 01/01/1960 Result – 01/12/11.

SECTION - 3

## Data pull

11) Run the DSM data pull code

Code description: This code pulls all the data and tables from oracle which will be required for the DSM run.

We can run the DSM\_datapull\_v1, DSM\_datapull\_v1\_2 codes in parallel. DSM\_datapull\_v1 and DSM\_datapull\_v1\_2 codes are the codes to pull ewt\_performer\_ng table from Oracle which is split into two tables due to the huge size of the table. DSM\_datapull\_v1\_3 code is run in the end to append the two performer tables. After this code, the intermediate tables for performer tables can be deleted. After this, the DSM\_datapull code can be run



SECTION - 4

## NextGen Payment Details

12) Run 02\_NG\_Pymnt.sas

Code description: Most of the data pull for Next Gen payments (amount, date, and check number) occurs in this code.



**Checkpoints:**

* This is one of the most important and longest running data pull for DSM

**NOTE** – 70% of the total data pull which is payment level happens in this code .It is being pulled from different tables which lead to several merges...

* Time taken: 20-24 hours.

**NOTE** – We can monitor the code through the UNIX platform where we run the codes by writing the command - ***ps***

If the code has failed/completed the command gives - ***codename done***

In case the code has failed we need to check the log file generated in SAS and debug the error, and run the code again once the problem has been solved.

* Check the min and max of the column d\_issue\_dt in the final table created (dsm\_base\_pymnts)

SAS CODE to find the minimum and maximum date of check issue

%include"/work/ew07/projects/dev/DSM\_Refresh\_Run/DSM\_DEC\_13/Dat/Dsm/Codes/00\_LibNames.sas ";

Proc sql;

Select max (d\_issue\_dt), min (d\_issue\_dt) from data.dsm\_base\_pymnts;

Quit;

DESIRED RESULT - 01/10/11, 30/09/13(for October 2013 extract)

* No of records ~19M
* Take 4-5 payment Ids and compare it with previous extracts for the same payments ids.

**For example –**

The file below contains the data pulled from one of the previous run final table for 4 random payment id’s and also the data for the same payment id’s from the current run table .This way we can do a column wise comparison for the payment id’s which we have pulled.



* If there is any discrepancy in previous and current extract there are two options.
  + If variable for which discrepancy exists is being used in subsequent codes, this discrepancy has to be fixed before running the subsequent

**NOTE**: Solution to any discrepancy will depend on the type of discrepancy in that particular variable. One case could be that the null count for that column is very high .In such a case back track the variable and find its base table , now check whether the nulls were there in the table before the data pull or they crept into the data after the pull. If the data did not have nulls in the base table then we are sure that some manipulation during the code has led to nulls in that column. Now we need to find the point in the code where these nulls crept into the data. And make the required changes in the code.

* + If variable for which discrepancy exists is not being used in subsequent codes, go ahead with the subsequent codes and debug the discrepancy in parallel

NOTE – This will save us time with respect to running of codes.

The above QC step will take ~ 2 hours

SECTION - 5

## Next Gen Model attributes creation

13) Run 03\_NG\_model.sas

Code description: Most of the derived variables used for modeling are getting created here.



Time taken: 12 hours

**Checkpoints:**

* Take 4-5 payment Ids and compare it with previous extracts for the same payments ids.

**For example –**

The file below contains the data pulled from one of the previous run final table for 4 random payment id’s and also the data for the same payment id’s from the current run table .This way we can do a column wise comparison for the payment id’s which we have pulled.



SECTION - 6

## Next Gen Model Scoring

14) Run 04\_NG\_model\_scoring.sas

Code description: Beta coefficients of regression model are embedded in this code. These coefficients are predefined. DSM Model scoring is performed in this code.



Time taken: 10 hours

**Checkpoints :**

* Ensure all payments have a valid model score that is < 1 (which in turn is used to assign them a valid percentile score

**NOTE**: Model score is the probability of a particular payment being fraud, so it needs to be less than 1.

* Take 4-5 payment Ids and compare it with previous extracts for the same payments

**NOTE:** All the fields for the same payment ids should be same apart from the model score, percentile score and details related to employees .Reason -

* SAS code to validate model\_score :

%include"/work/ew07/projects/dev/DSM\_Refresh\_Run/DSM\_DEC\_13/Dat/Dsm/Codes/00\_LibNames.sas ";

Proc sql;

Select count (\*) from cpat.dsm where model\_score < ‘0’ or model\_score > ‘1’;

Quit;

DESIRED RESULT - There should be 0 such entries. This query gives the count of payments which have a model score greater than 1 .And any probability value cannot be greater than 1 hence this count should be zero.

SECTION - 7

## IFM Part 1

15) We can run the IFM codes just after the IFM (Legacy) Dataset is made available. IFM data set is generally available by 3rd working day of the month.

NOTE – This dataset is provided by Prabha in the dash boarding team .In case it is not available reach out to her. To make sure the dataset is correct the steps below should be followed.

SAS Code to check the min and max date in the given IFM dataset -

Proc sql;

Select max (d\_issue\_dt), min (d\_issue\_dt) from internal.icf\_p2\_&lastmonthday.\_2yr;

Quit;

DESIRED RESULT - 01/10/11, 30/09/13(for October 2013 extract)

Run 01\_IFM\_Part1.sas



Time taken: 10-12 hours

**Checkpoints:**

* Take 4-5 payment Ids from the final table of this code and compare it with previous extracts for the same. Example to this has been provided above.
* Ensure the tables which are finally used for merging with the DSM data are populated in the desired library. And to ensure that, we need to go to that particular location as mentioned in the code.

SECTION - 8

## IFM Part 2

16. Run 02\_IFM\_Part2.sas



Time taken: 8-10 hours

**Checkpoints:**

* Take 4-5 payment Ids from the final table of this code and compare it with previous extracts for the same
* Ensure the tables which are finally used for merging with the DSM data are populated in the desired library. And to ensure that, we need to go to that particular location as mentioned in the code

SECTION - 9

## Final IFM DSM Merging

17.05\_Merge.sas

Code Description: Final merging of IFM and DSM data happens in this code. This code can be run only when all the above codes have run successfully. In this code all the data pulled is merged to form a dataset of the magnitude of around 18 million rows and around 120 columns .Further this huge dataset is joined to the IFM data as well as some derived columns are also merged into it. This pushes the time taken by the dataset



Time taken: 20-24 hours.

**Checkpoints:**

* This is one of the most important and longest running data pull for DSM
* Take 4-5 payment Ids and compare it with previous extracts for the same payments Id. If there is any discrepancy in previous and current extract there are two options
  + If variable for which discrepancy exists is being used in subsequent codes, this discrepancy has to be fixed before running the subsequent codes
  + If variable for which discrepancy exists is not being used in subsequent codes, go ahead and run subsequent codes and debug the discrepancy in parallel

NOTE – The concept behind is to save time by debugging the problem in parallel before that particular variable is used again.

* The above QC step will take ~ 2 hours
* SAS CODE TO CHECK NUMBER OF ROWS -
* %include"/work/ew07/projects/dev/DSM\_Refresh\_Run/DSM\_DEC\_13/Dat/Dsm/Codes/00\_LibNames.sas ";

Proc sql;

Select count (\*) from cpat.final\_output\_dec2013;

Quit;

DESIRED RESULT ~18M

NOTE – This number can vary from 17.5 to 18.5M

## Check reconciliation status

18) Code description: This code creates columns for check reconciliation status and cat codes.

**NOTE:** This column is representing the current status of the check issued to make that particular payment. The different values which this column can hold are

-Recoded

-Warehoused

-Cancelled

-Cleared

-Pending

This helps the client to differentiate between the payments marked with high fraud probability on the basis of the status of the check issued for that particular payment .Client will give more importance to the payments in which the status of check is Cleared .And give almost no importance to payments where the check is Cancelled .



Time taken: 4-5 hours

SECTION - 10

## Run Address\_Match Code

19) This code is run to convert the addresses in the output table to a standardized address to make it easier to compare the addresses for the address match flag.



Time taken: 30 minutes-1 hour

## Run EFT Code

20) This code is used to encrypt and prepare a table for all the EFT payments in the DSM timeframe.



Time taken: 30 minutes-1 hour

## Run MCO\_details code

21) This code is used to update the correct MCO details for all the payments



Time taken: 30 minutes – 1 hour

## Run Oracle Name match code

22) This code is used to create the initial payee payer name match tables.



Time taken: 2 – 4 hours

## Run Oracle Name match SAS code

22) This code is used to create the final payee payer name match tables. The reason for running this part of the code in SAS is due to the heavy load this code places on the Oracle servers



Time taken: 30 minutes - 1 hour

SECTION - 11

### Oracle Codes

The flow of codes in oracle is given below along with the oracle codes that are used in the DSM process



SECTION - 12

# Data QC

Note:

Issue log: Issue log will be maintained, revised and consulted before/during run. New issues will be appended and the solution for respective defects will be updated regularly.

**Basic Preparation stage:**

* The availability of space should be checked before the run, it is recommended that 600-700 GB free space should be present in the directory
* The time period considered for every run is 24 months (2 years), dates used in queries should also be checked manually before firing the codes
* Log should be checked to verify that the automated date update query is running properly and the relevant month name is present in the log file
* The date in run date folder codes are present in SAS format, should be verified that they are correct
* Location should be checked whether the folders are created or not at the specified location
* Check the access on the created files and folders, it should be accessible by everyone

**Data fetch and scoring:**

* Count distinct payment IDs in the base table and compare it with the average payment counts from previous runs
* Repeat the above exercise on Legacy base dataset (legacy database)
* Once the payments are scored, payment IDs and their counts should be checked in the scored dataset

Merge (NextGen + Legacy data):

* It is extremely important that at this stage the consolidated dataset should be checked for duplicates

Creation of derived variables:

**Final Merge:**

* The final dataset must be checked for duplicates and few top scored payments should be verified with database information/NG application

**Basic checks before extract creation:**

* Ensure all payments have a valid model score that is < 1 (which in turn is used to assign them a valid percentile score).
* Ensure the volume of payments for each flag in the ENAM extract appears reasonable / similar to the prior ENAM extract
* Ensure all column names reflect the business names and are consistent for every extract
* Ensure all the extracts are created in Tableau 6 , as the workbooks client utilizes do not work well in other versions of Tableau.
* For all the column null count

Few general codes which can be used for QC of data:

* Code to count rows in a table in oracle
  + **Select count(\*) from table name**
* Code to find the minimum and maximum dates for which the data is pulled
  + **Select max(d\_issue\_dt) , min(d\_issue\_dt) from table name**
* Code to find the number of nulls in multiple columns in a table (oracle query)
  + **SELECT sum(case when PAYMENT\_AMOUNT is null then 1 else 0 end) as PAYMENT\_AMOUNT,**

**sum(case when N\_PYMNT\_ID is null then 1 else 0 end) as N\_PYMNT\_ID ,**

**sum(case when CHECK\_NUMBER is null then 1 else 0 end) as CHECK\_NUMBER,**

**sum(case when ALPHA\_ID is null then 1 else 0 end) as ALPHA\_ID**

**From tablename**

The excel sheet below gives a comparison and also the range in which the null count per column has been in the last 4 bimonthly run’s.



# Tableau Extract

25) Final Extract creation using tableau.

Description: Five extracts need to be created

* 1.) Full Extract - Final table is used to create this extract without any extra conditions 2.)Mini Extract - Final table is used with conditions including either of the three below

Mentioned columns (flags) must be ‘YES’. {Columns: PAYEE\_PAYER\_NAME\_MATCH, PAYEE\_EMPLOYEE\_NAME\_MATCH, PAYEE\_EMPLOYEE\_ADDRESS\_MATCH}

3.)“Mini” –Full Extract – Final table is used to create this extract, client has provided a set of columns which should be present in this extract ,the query below contains all those columns and it needs to be used as the extraction query in tableau to create the extract.



4) The EFT extract is prepared to show those details for when the number of EFT payments to a bank account is greater than 5



5.) Another extract is prepared to show the details when the number of payments to a particular address is greater than 5 and 90% or more of those payments are made by the same alpha id



* Reason for creating tableau extracts – The client already has at least two main workbooks in Tableau to which the extracts which we provide which we provide are connected .So in these workbooks the client already has the various formulas which they need to analyze this data ,so we just have to provide the data in the particular format compatible to their workbooks .So the conditions which we put for the ENAM extract have been defined by the client as they have created the workbook for that particular extract accordingly. And moreover the conditions to create this extract (either of these three flags are ‘YES’, flags- PAYEE\_PAYER\_NAME\_MATCH ,PAYEE\_EMPLOYEE\_NAME\_MATCH ,PAYEE\_EMPLOYEE\_ADDRESS\_MATCH) make these payments distinct from the other payments. So ultimately the client wants to analyze these particular payments differently.

**NOTE:**

1. Explicitly use Tableau for extract creation.
2. Check all the column names before creating the extract ,they should exactly match with the names in the final table of the old extracts for ex. final table for October run is dsm\_output\_oct
3. Ensure there is sufficient amount of space (at least 50GB) on the remote machine where the extract is created.
4. Follow the below naming convention for the extracts

Name of full extract - DSM\_OUTPUT\_OCT

Name of ENAM extract –DSM\_OUTPUT\_OCT\_ENAM (**OCT refers to October, so it will change with every extract** finally the extracts need to be placed at the following location

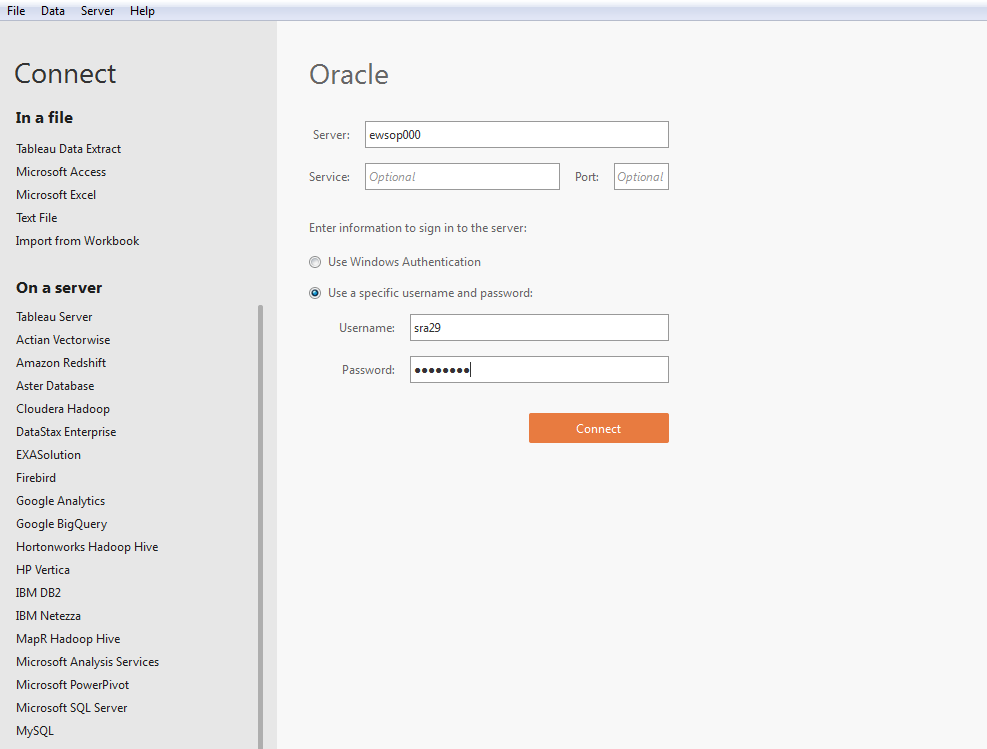
[\\ad\nas\NGDID\_CPATData\DSMTableauDatasources\2013](file:///\\ad\nas\NGDID_CPATData\DSMTableauDatasources\2013) October

Similarly the extracts for October 2012 run have been placed in the location –

\\ad\nas\NGDID\_CPATData\DSMTableauDatasources\2012 October

**Steps to create the tableau extract -**

1. Open tableau
2. Click on ‘connect to database’ option
3. Select oracle as connection
4. Connect using ‘ragav’(person handling DSM process) credentials for oracle



1. After connecting to oracle, select the schema in which the final table is created .Then select the final table (for ex. dsm\_output\_oct).
2. Last step is to go to the export option in the menu bar and click on the export data option
3. Time taken: 10-12 hours

**NOTE**- The steps may vary for different versions of Tableau

# Timeline for Bi Monthly refresh

Below is an example of the timelines for a DSM Refresh Run in general.

