CA Test Data Manager

Mainframe InFlight Masking

Best Practices Guide

Author : Walter Guerrero

Version: 1.21

Date: 6/11/2018

Table of Contents

Overview 3
TDM MF Requirements 3
TDM Mainframe Support 3
Mainframe In-Flight Data Masking 5
GTXMSKL – JCL Procedure Flow 6
In-Flight Masking Scenario 8
Best Practices 21
Appendix A 25
Useful Links 25

Overview

This setup will demonstrate a customer that needs to mask a set of columns in DB2 for z/OS tables, and generate the necessary masking and sub-setting rules for these DB2 for z/OS tables.

There is a requirement that different sets of datasets that have been exported from the production DB2 for z/OS subsystems be masked and subset based on the rules that the test data engineer has defined.

We will show how to setup the data masking and sub-setting rules (in-flight masking), which will be used to generate the desired masking and sub-setting rules and execute the mainframe processes to make sure that the masking rules are applied to test data.

TDM MF Requirements

The following requirements need to be completed prior to the use of the TDM Mainframe toolkit:

- Mainframe user access
  - TSO access
  - ISPF editor access
  - DB2 instance access
- Mainframe user needs to have ftp capabilities
- QWS3270 or equivalent installed
- TDM 4.x
- DB2 Connect v10.x or better (or DB2 standard with DB2 Connect feature)

TDM Mainframe Support

The Test Data Management Mainframe package is composed of:

- CA TDM mainframe objects (PGMs and JCL procs)

The above packages can be downloaded from the CA Support site as needed, in this example, the version being downloaded is 5.4.13 or greater:

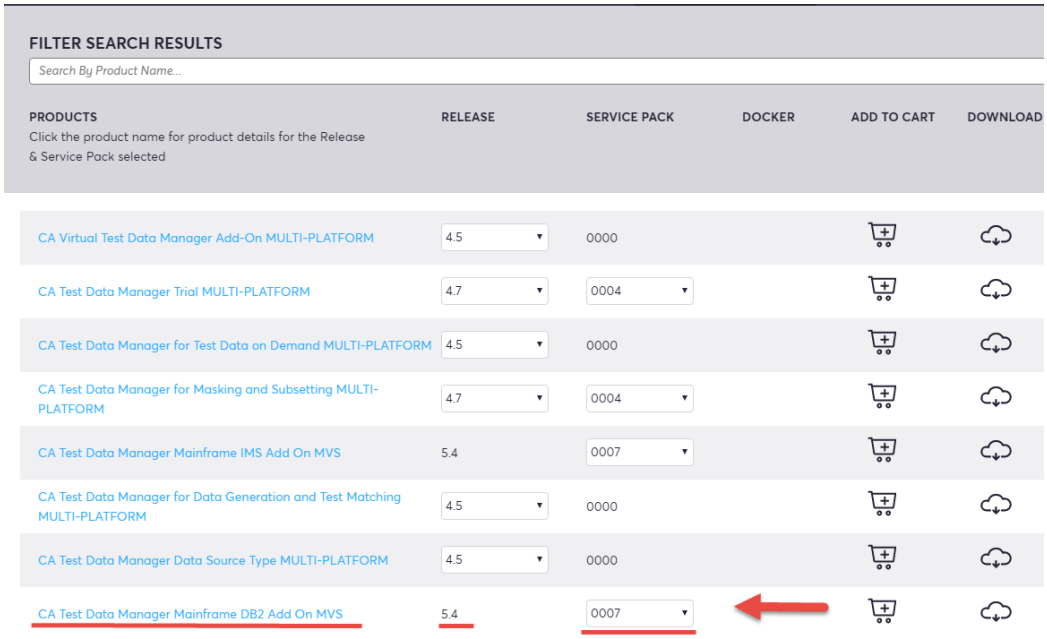


Figure: CA Support with the correct parameters

Select the mainframe package listed above, in this case it will be the package for DB2 for MVS, click on name to take to the following panel.

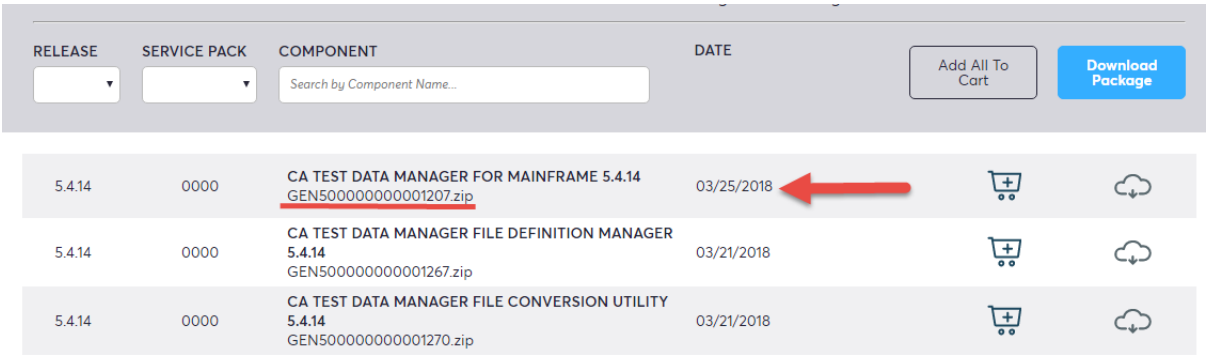


Figure: Typical listing of TDM Mainframe components

At this level, you will select the package highlighted, which contains the mainframe binaries needed to perform in-place/in-flight masking natively in the mainframe.

To conduct the mainframe toolkit installation, please follow the instructions defined in the TDM online docs.

https://docops.ca.com/ca-test-data-manager/4-5/en/installing/mainframe-installation-and-upgrade

Mainframe In-Flight Data Masking

Data masking hides or obfuscates sensitive and classified data. The goal is to protect data that is used for purposes such as development, testing, and QA cycles. Data masking is a standard practice that is often required for compliance with national and international data protection legislation.

To perform the necessary data masking natively in the mainframe, you can use Datamaker transformation maps to mask the data. The approach that you select depends on your business requirements and feasibility. You can adopt one of the following approaches to masking with regards to which stage the data is masked at:

- In-place masking

  In this case, a typical scenario is that the production data is copied over to a staging area. You use DataMaker to create a transformation map with the necessary rules, upload this transformation map and use the RUNJCL(GTXMSK) JCL procedure pointing to this staging database and masks the data that resides there. This *masked* data is then copied over to different testing environments as required.

- In-flight masking

  In this case, you use Datamaker transformation maps and Subset scripts. You first define a transformation map (Oracle or MSSQL) in Datamaker, create masking functions for the columns you want to mask. You use the Subset interface to create the *masked* export scripts. These scripts perform masking as they export the source data to a dump file. The dump file (which contains masked data) is then imported into the target database. Testers can use the same database, which now includes masked data, for testing.

The mainframe data masking facilities are design to help with the masking of DB2 datasets natively in the mainframe environment. These facilities provide you with consistent, robust, and repeatable methodologies for securing sensitive data.

Below is a listing of the more common mainframe programs that you will be using for in-place anc in-flight masking. Prior to the transfer of the XMI files, it is advisable that you pre-allocate these files in the mainframe based off the following values and defined as partitioned dataset files (PDS).

Program Name	JCL Proc	Purpose
GTXMSK	RUNJCL(GTXMSK)	This is the program that will perform the in-place masking.
GTXMSKL	RUNJCL(GTXMSKL)	This is the program that will perform the in-flight masking
GTXMSKF	GRIDT01.LOADLIB	This is the program that will perform the in-place masking of a flat file.

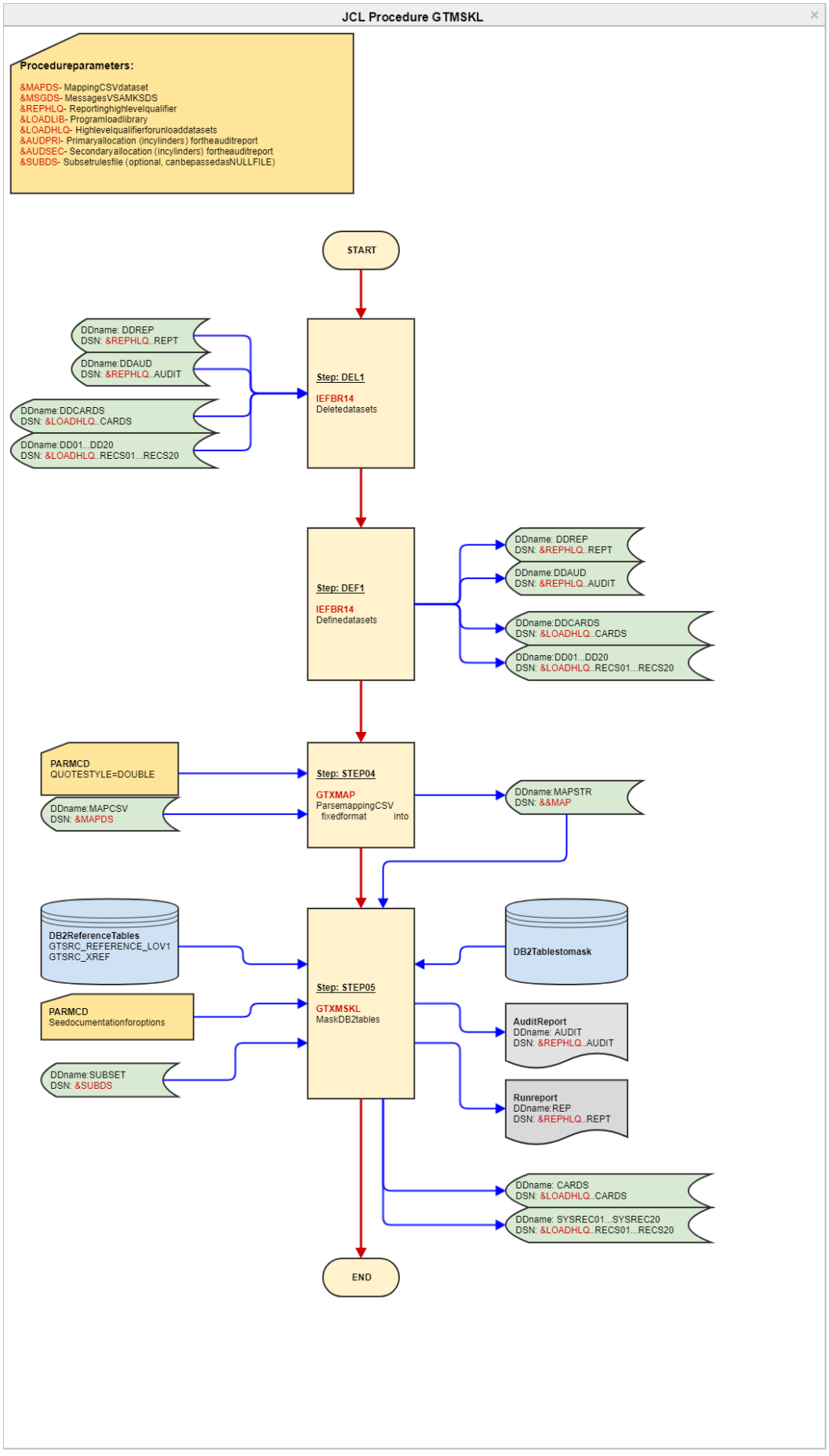
The above programs require the uploaded members in the following datasets:

Dataset	Member	Purpose
LIB.MAPCSV	Transformation_map_name	This dataset will contain the transformation map rules that will be used for the in-place and in-flight masking.

LIB.SUBS	Subset name	This file contains the subset members that you generated and will be used for the in-flight masking effort.

GTXMSKL – JCL Procedure Flow

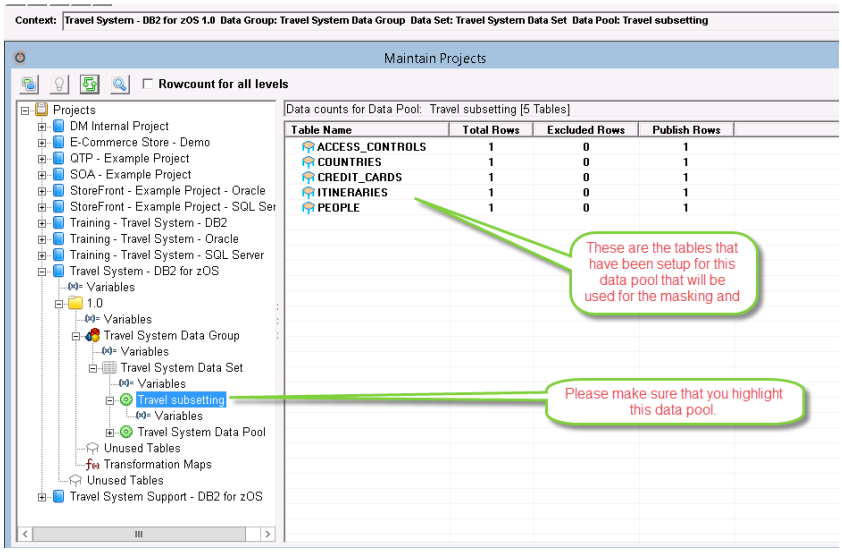
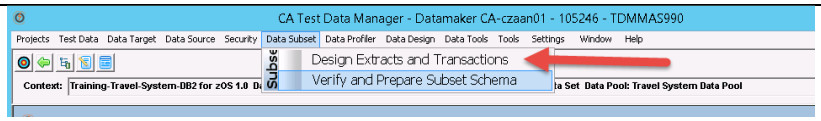
This diagram shows the flow of the GTXMSKL JCL procedure.

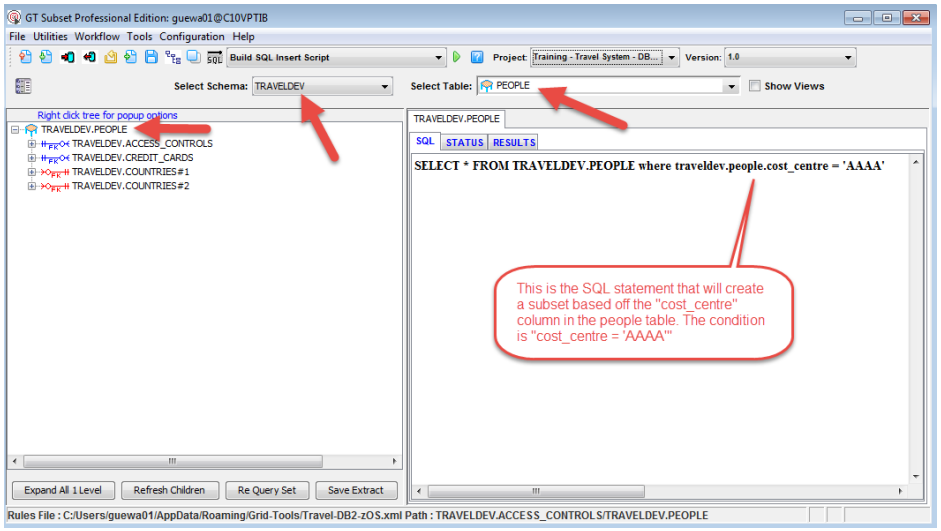

JCL Procedure GTMSKL
Procedureparameters:
&MAPDS- MappingCSVdataset
&MSGDS- MessagesVSAMKSDS
&REPHLQ- Reportinghighlevelqualifier
&LOADLIB- Programloadlibrary
&LOADHLQ- Highlevelqualifierforunloaddatasets
&AUDPRI- Primaryallocation (incylinders) fortheauditreport
&AUDSEC- Secondaryallocation (incylinders) fortheauditreport
&SUBDS- Subsetrulesfile (optional, canbepassedasNULLFILE)
START
DDname: DDREP
DSN: &REPHLQ..REPT
DDname:DDAUD
DSN: &REPHLQ..AUDIT
DDname:DDCARDS
DSN: &LOADHLQ..CARDS
DDname:DD01...DD20
DSN: &LOADHLQ..RECS01...RECS20
Step: DEL1
IEFBR14
Deletedatasets
Step: DEF1
IEFBR14
Definedatasets
DDname: DDREP
DSN: &REPHLQ..REPT
DDname:DDAUD
DSN: &REPHLQ..AUDIT
DDname:DDCARDS
DSN: &LOADHLQ..CARDS
DDname:DD01...DD20
DSN: &LOADHLQ..RECS01...RECS20
PARMCD
QUOTESTYLE=DOUBLE
DDname:MAPCSV
DSN: &MAPDS
Step: STEP04
GTXMAP
ParsemappingCSV
fixedformat into
DDname:MAPSTR
DSN: &&MAP
DB2ReferenceTables
GTSRC_REFERENCE_LOV1
GTSRC_XREF
PARMCD
Seedocumentationforoptions
DDname:SUBSET
DSN: &SUBDS
Step: STEP05
GTXMSKL
MaskDB2tables
DB2Tablestomask
AuditReport
DDname: AUDIT
DSN: &REPHLQ..AUDIT
Runreport
DDname:REP
DSN: &REPHLQ..REPT
DDname: CARDS
DSN: &LOADHLQ..CARDS
DDname: SYSREC01...SYSREC20
DSN: &LOADHLQ..RECS01...RECS20
END

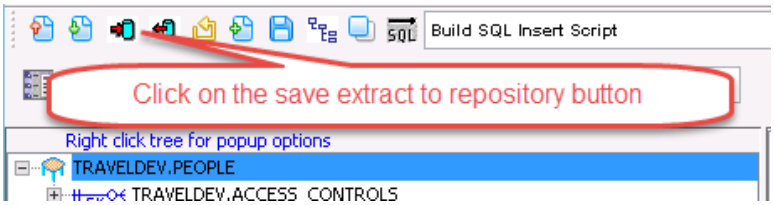
In-Flight Masking Scenario

Now we are going to step thru a "*In-flight*" scenario, where we are going to be creating a transformation map, setting up a subset job and attaching the subset job prior to the generation of the transformation map. These files will be uploaded to the mainframe and placed in the two datasets listed above.

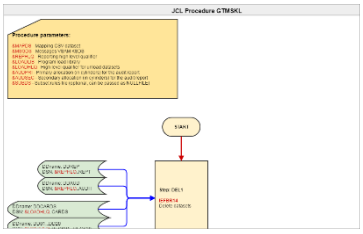
Please keep in mind that this document also includes a JCL proc TDMDBLD (Appendix A) to load the masked data back into another DB2 subsystem.

Screen Shot	Message & Action
 	Click on the "Maintain Project" Icon. You will see the available projects, and pay close attention to the DB2 for z/OS project. There are two schemas in this scenario: - The source schema is pointed to GRIDDEMO. - The target schema, which will be very relevant to us is pointing to the TRAVELDEV schema. - The target tables are already present in the target schema.
	Now, we need to go create the subset that will be use as part of this use case. Select the "Design Extracts and Transactions" menu option.





Now that the sub setting dialog is opened, you will need to make sure that the people table is selected as shown, as well as the “traveldev”

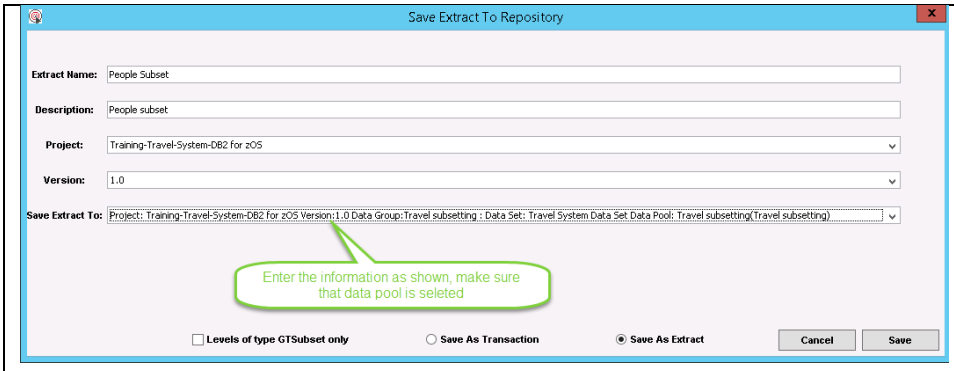


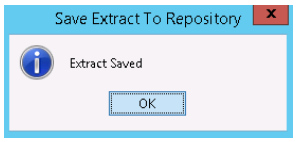
schema is also selected.

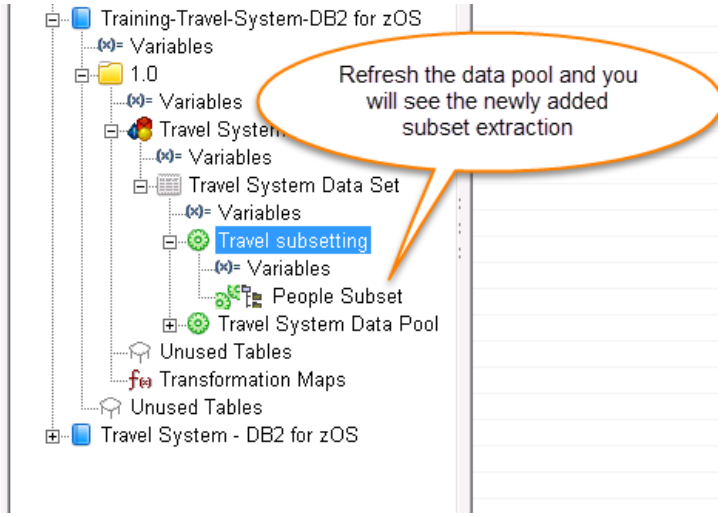
We are going to be creating a subset that will be used in the same schema that provided the data, which will be subset and masked.

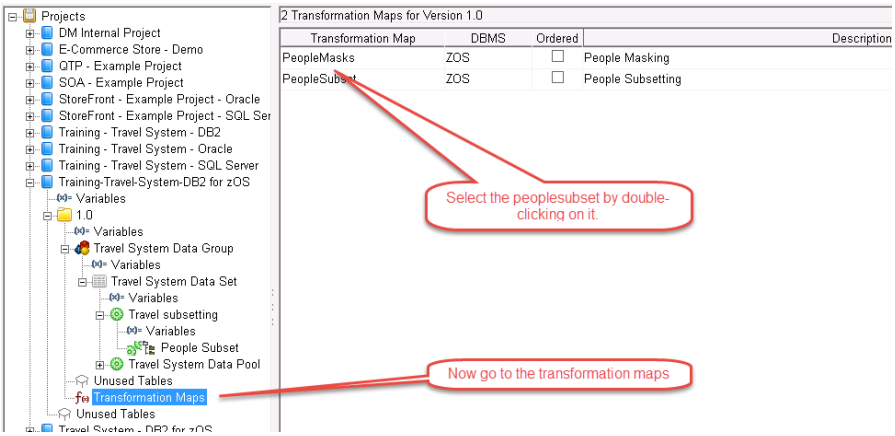
Make sure that the SQL statement is completed as shown.

You will then create an extract to the TDM repository.







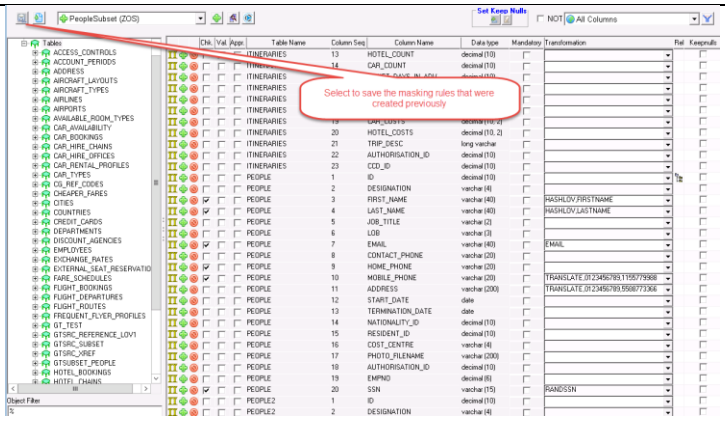


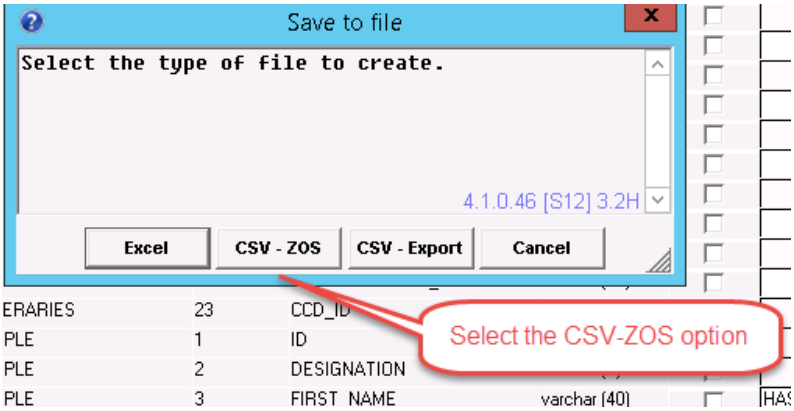
Make sure that this subset is saved as shown to the data pool selected.

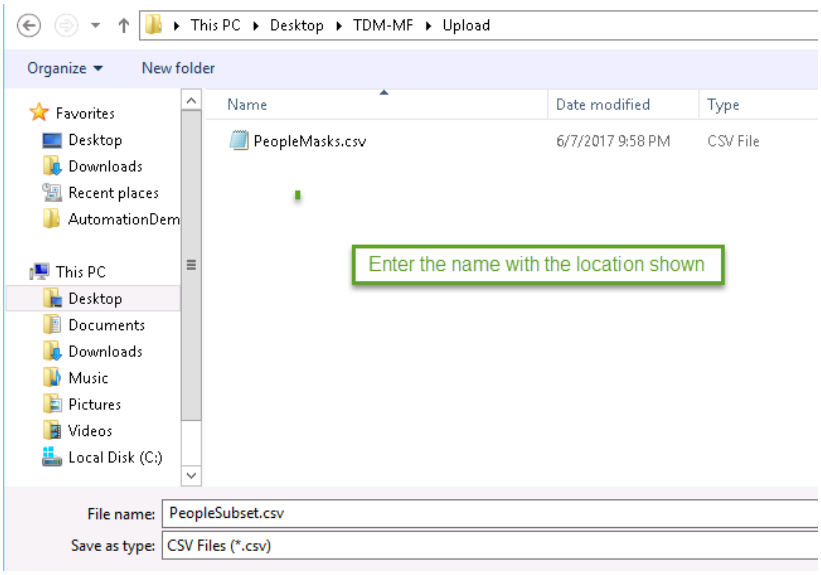
Afterwards go back to TDM and refresh the project tree, you will see that the subset extract that you created has been placed under the “Travel subsetting”.

After that you will need to click on the transformation maps in the project tree.

You will need to select the “PeopleSubset”, so that you can create a set of masking rules that will be converged with the subset extract that you created previously.



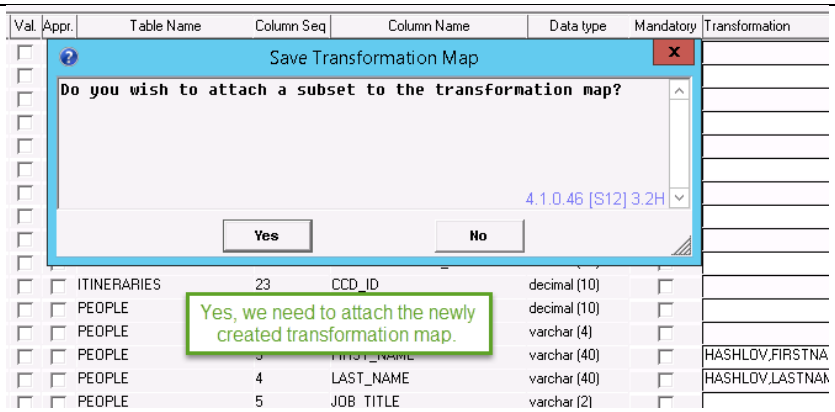


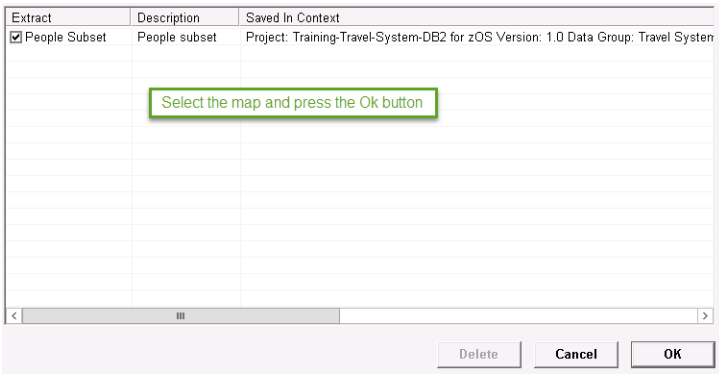


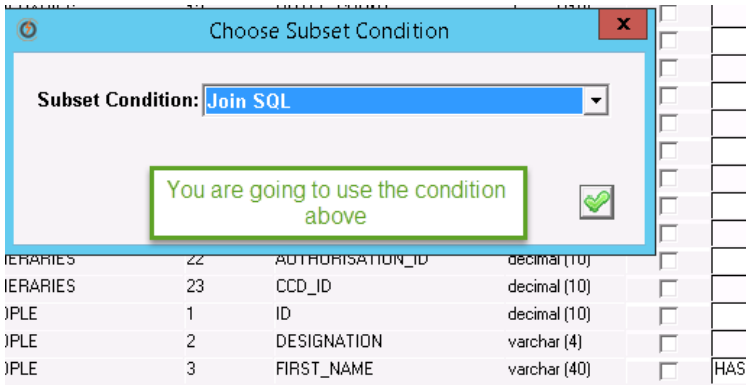
You can scroll down until you see the “People” table, and the fields that have been selected to be masked.

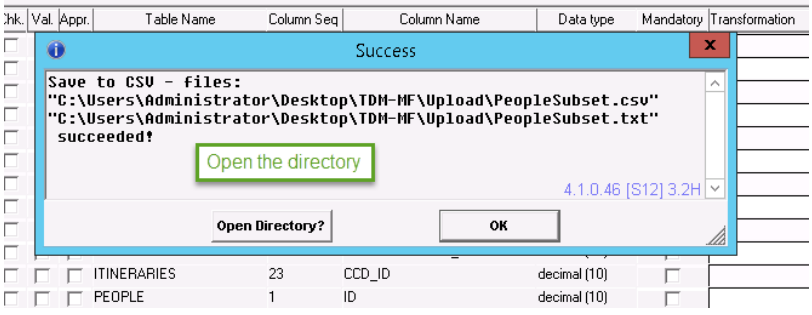
You will need to save the masking rules to a “***CSV-ZOS***” option, which you can do by clicking on the “Save” button located on the left of the dialog.

Enter the name of the file as shown on the location shown. This is important, since this is the location from which you will be uploading the files to the mainframe.









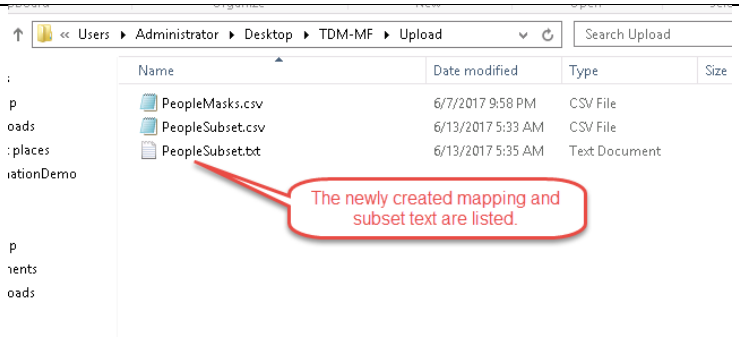
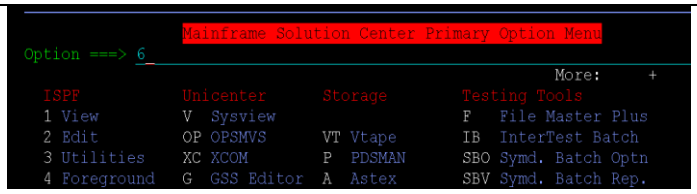
Since there is a subset extract that is part of this data pool, you will be asked if you would like to include the transformation map.

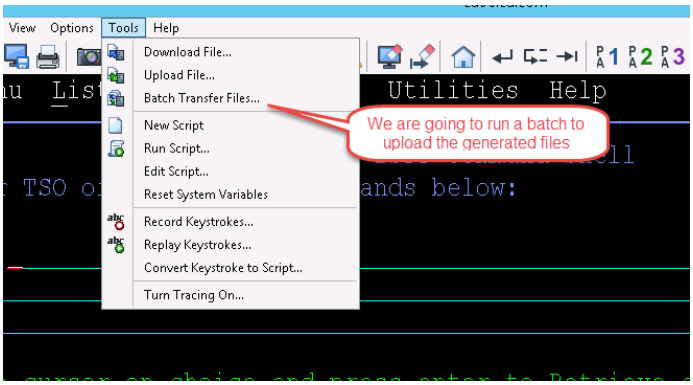
Select the “Yes” button to include the subset extract.

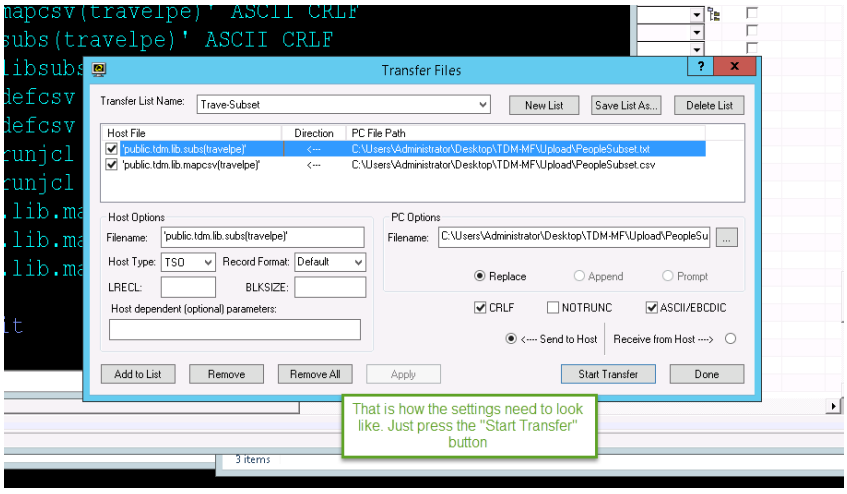
Select the subset extract created previously.

Select “JOIN SQL” as the subset condition.

Once the masking and subset rules are created, please open the directory location to make sure that you show the contents of these newly created files.

<table>
<tr>
<td> 
</td>
<td>Here are the newly created files.</td>
</tr>
<tr>
<td> 



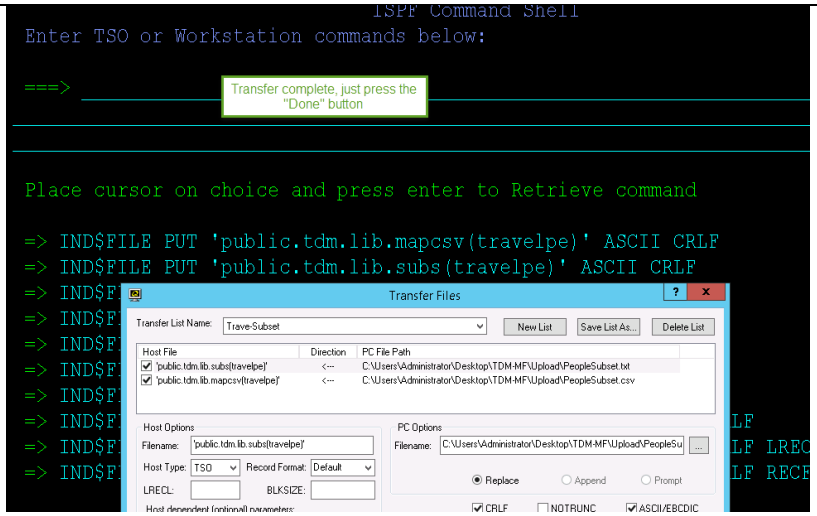

</td>
<td>Now you need to connect to the mainframe system using your mainframe credentials.

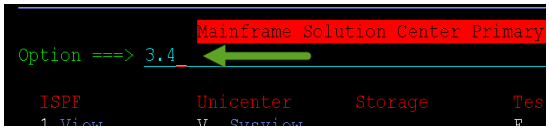
You will need to select the option to enter the “TSO” environment, and enter “ISPF” to start at the base menu.

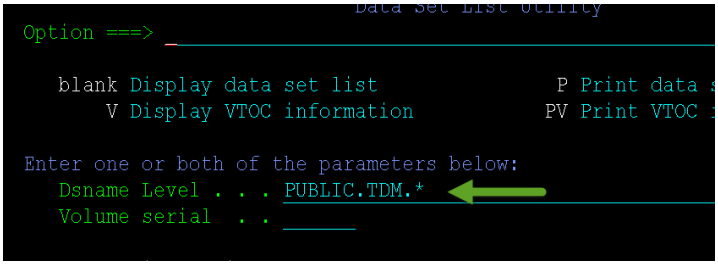
Select option 6 to enter the “TSO Command Line”.

From the Tools→Batch Transfer Files menu option, you will start a batch job, which has been preset already.

Just press the “Start Transfer” button.</td>
</tr>
</table>







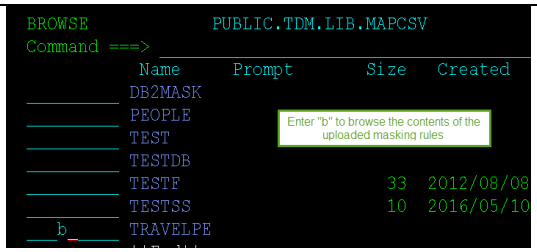
Once the transfers have been completed, you can close the batch transfer dialog by pressing “Done”.

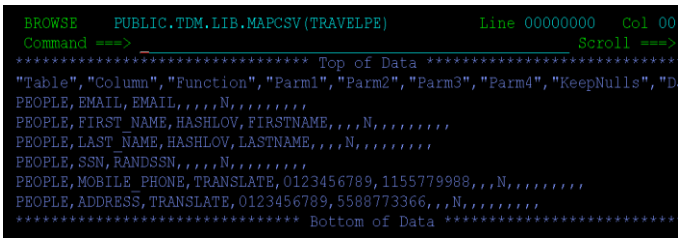
Press PF3 to return to the main menu.

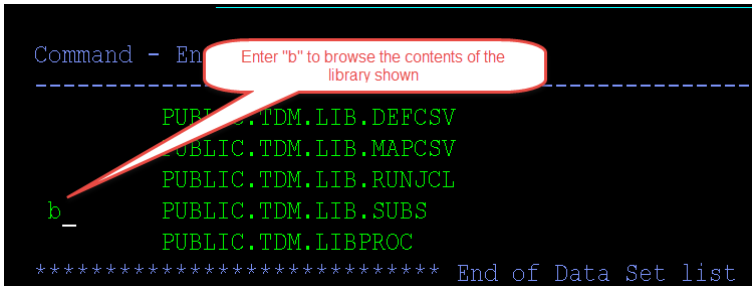
Select option “3.4” to display the data sets.

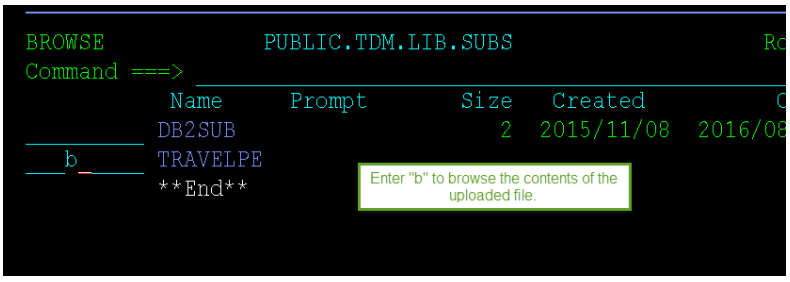
The dsnname level that you will be starting at will be “PUBLIC.TDM.*” This is just an example, your entries could be starting with “*GRIDT01.LIB*”..

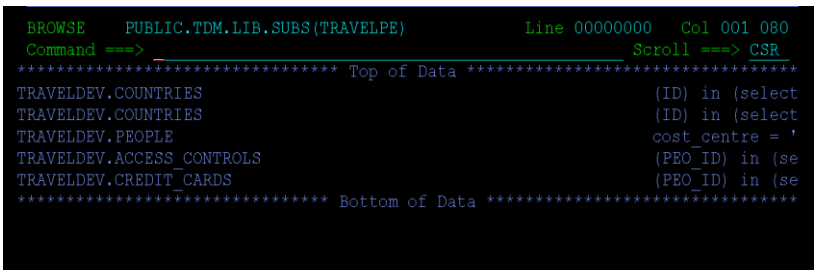
Once you have the list of datasets display, we can start by browsing the dataset library as shown. Here we are going to look at the masking file contents for the file that we just uploaded.









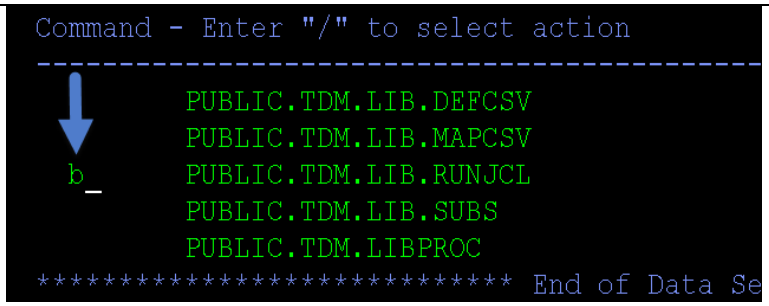


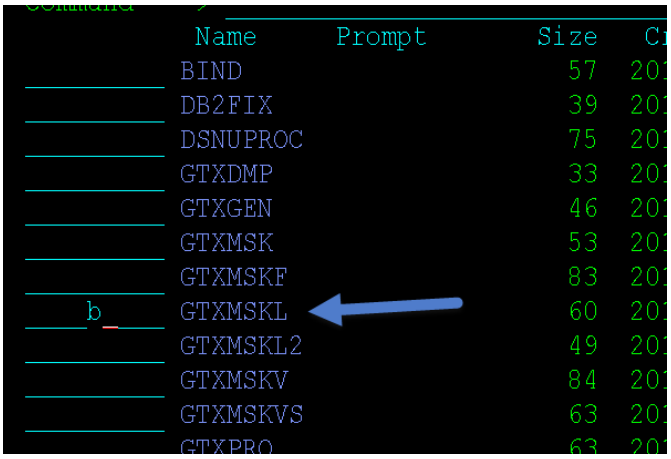
Once inside the dataset library, we are going to look for “TRAVELPE”, and browse the contents of this file.

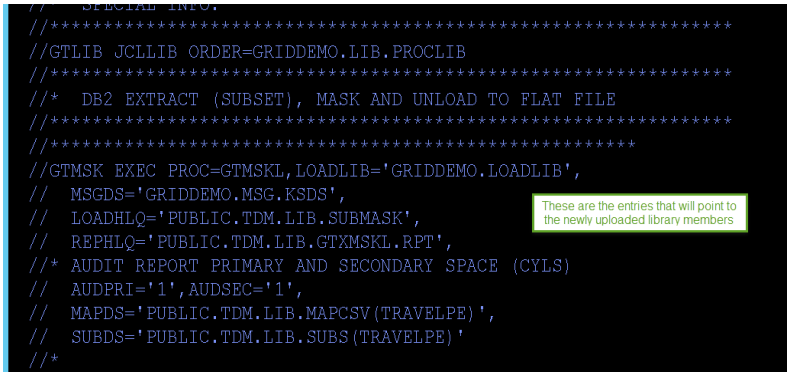
After looking at the contents of the masking file, we need to press PF3 twice to return the dataset listing, and browse the “PUBLIC.TDM.LIB.SUBS” for the subset rules that we uploaded.

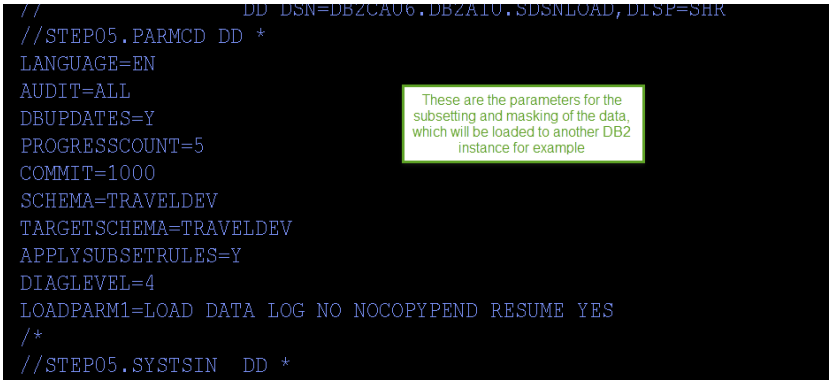
Once inside of this dataset, we are going to browse the “TRAVELPE” member, which contains all the subset rules.

After reviewing these subset rules, we are going to press PF3 twice to return to the dataset listing.







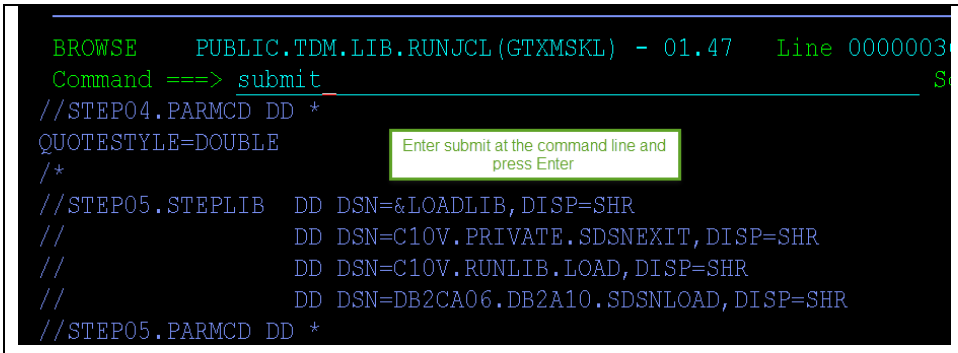


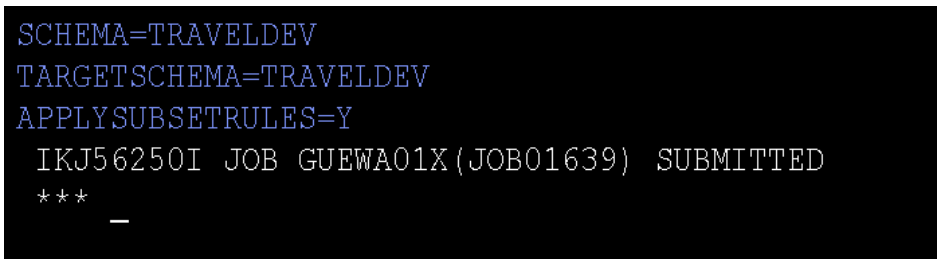
At the dataset listing, we are going to browse the RUNJCL dataset library as shown.

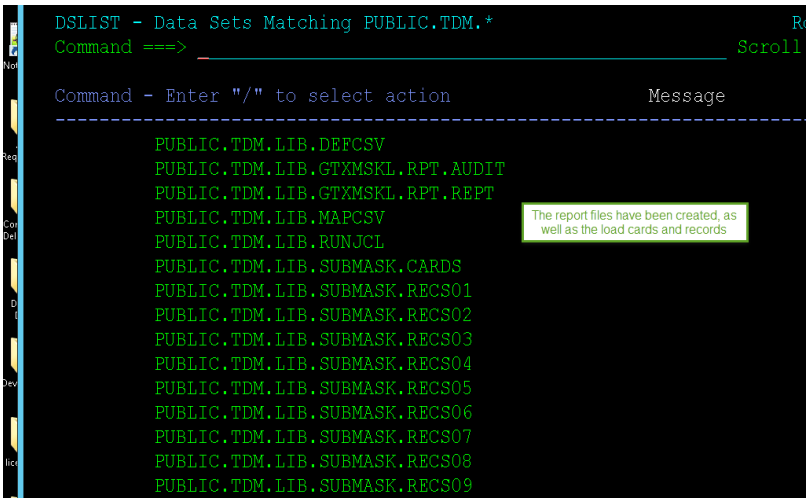
The objective here is to browse the contents of the "GTXMSKL" JCL procedure, which perform the unloading of the data based off the subset rules and mask the resulting data files.

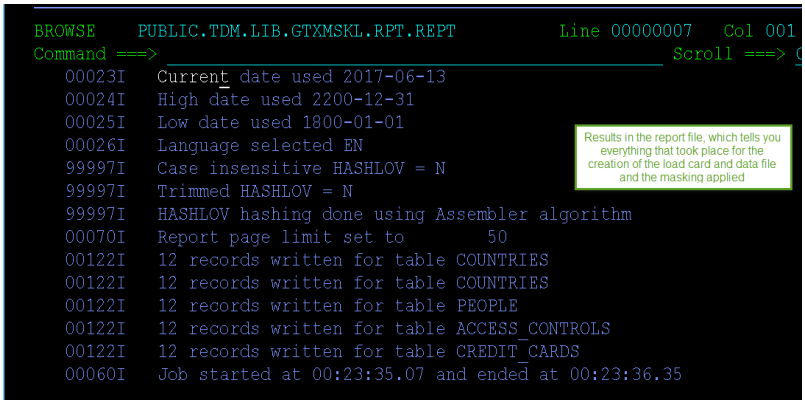
You will need to update the proc with the map and sub member names, as well as update the "***schema***" and "***targetschema***".

You will need to update the "***LOADHLQ***" and "***REPHLQ***" entries.









Enter “Submit” at the command line in the JCL procedure.

You will get a message that the JCL procedure has been submitted to the JOB subsystem.

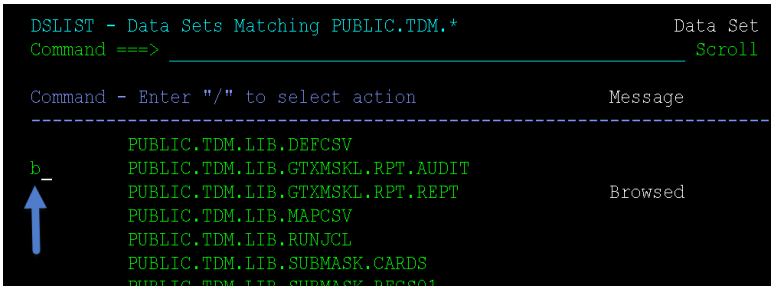
Waiting about 45 seconds and the files shown will appear.

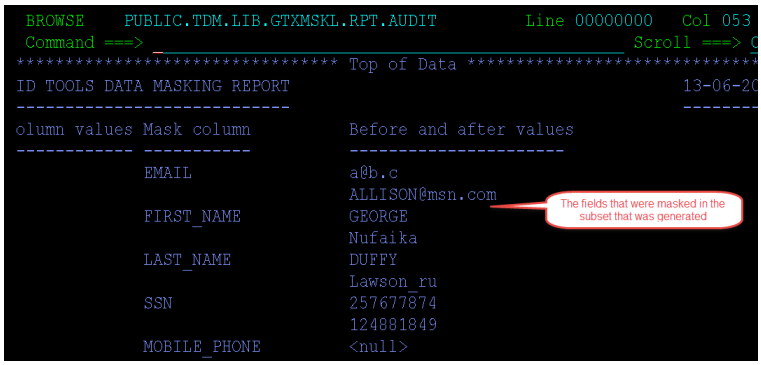
These files are:

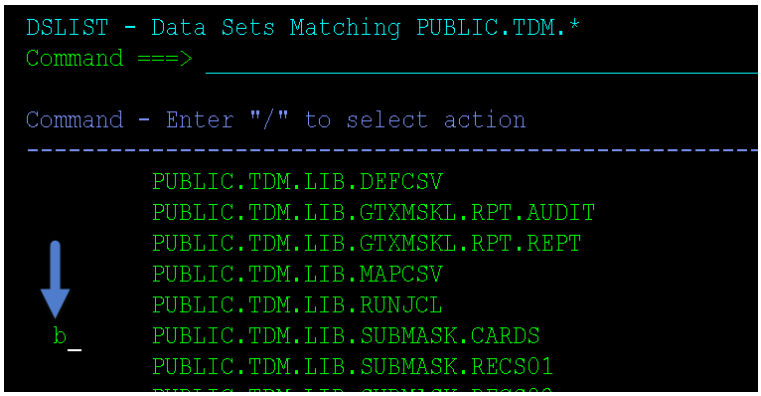
- Load data card
- Data files
- Report file
- Audit file

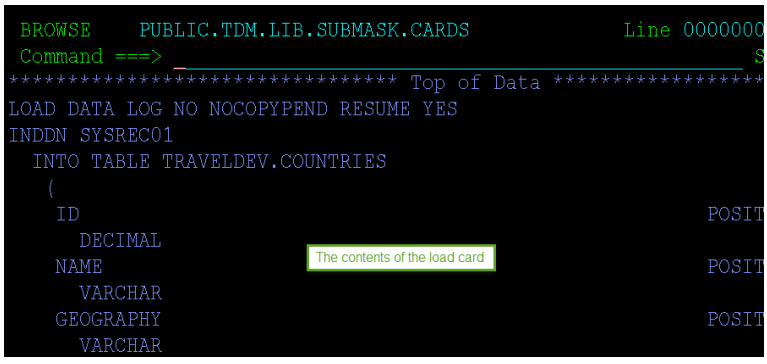
We are going to browse the REPT file to get a status report of the unloading and masking of the data.

Press PF3 once to return to the main dataset listing.







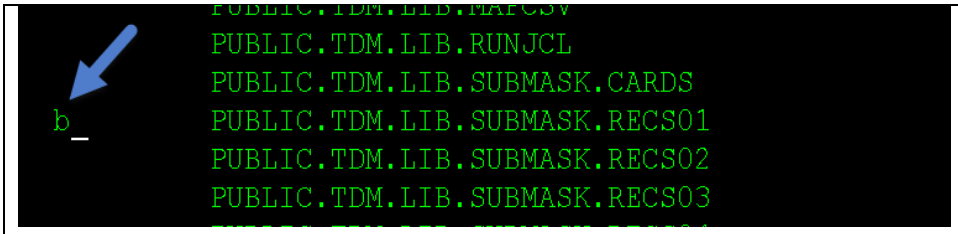


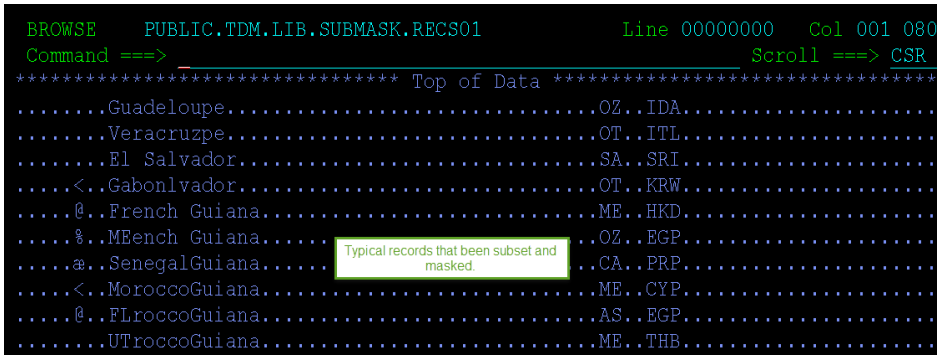
Now we are going to look at the audit report, which will show us the fields that were masked with their original and masked values.

Press PF3 to return to the dataset listings.

Now we are going to browse the load data card, so you can see who the load command with the supporting tables is organized.

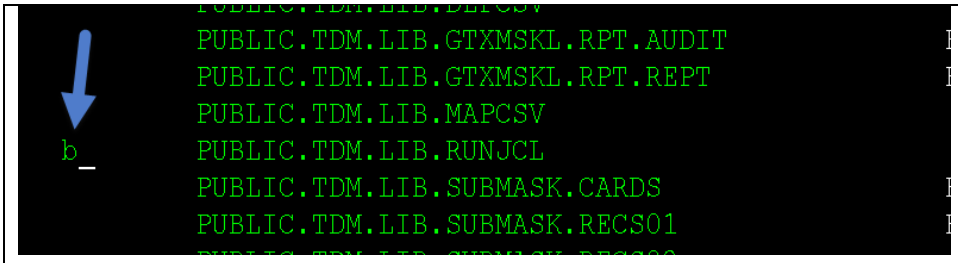
Press PF3 to return to the dataset listing.

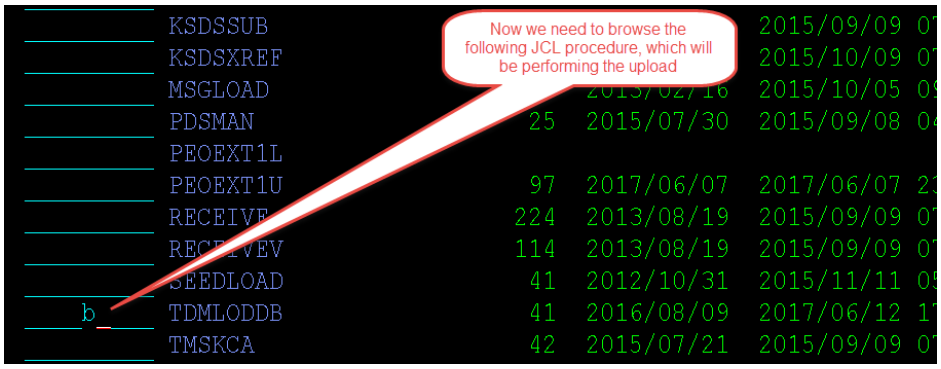


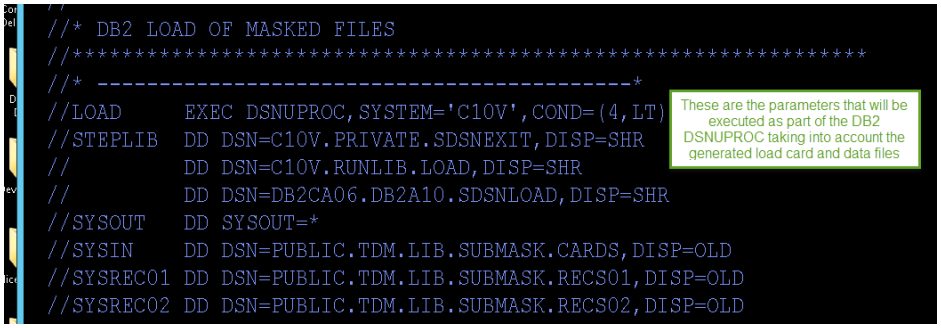


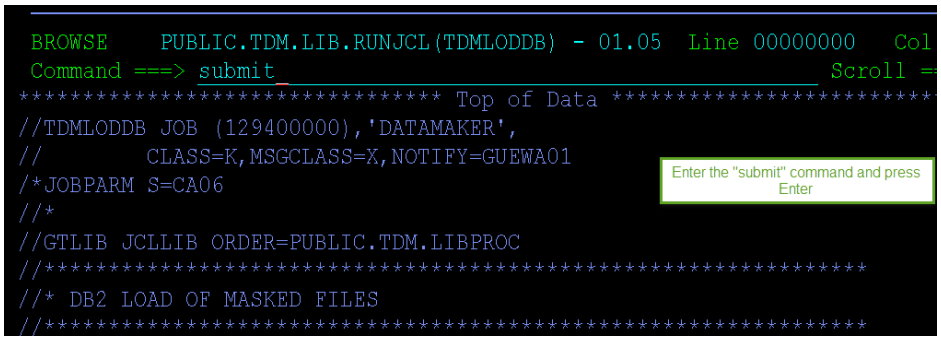
We need to browse one of the data files that was generated. These are rows that will be loaded back into the target database schema that was pre-defined in the GTXMKSL procedure.

Press PF3 to return to the dataset listing.









We need to browse the RUNJCL dataset library again.

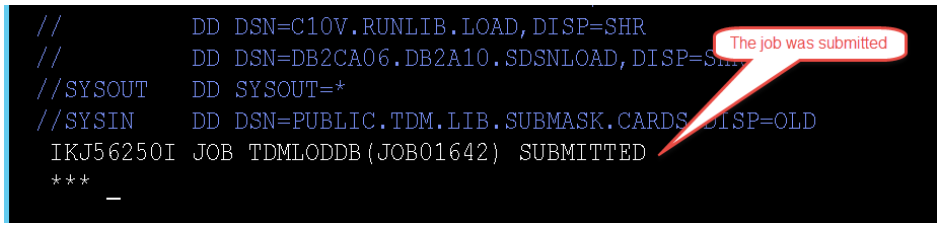
This time we are going to browse the TDMLODDB member.

This JCL procedure contains the necessary instructions to load data into the target schema based off the previously generated load card and data files.

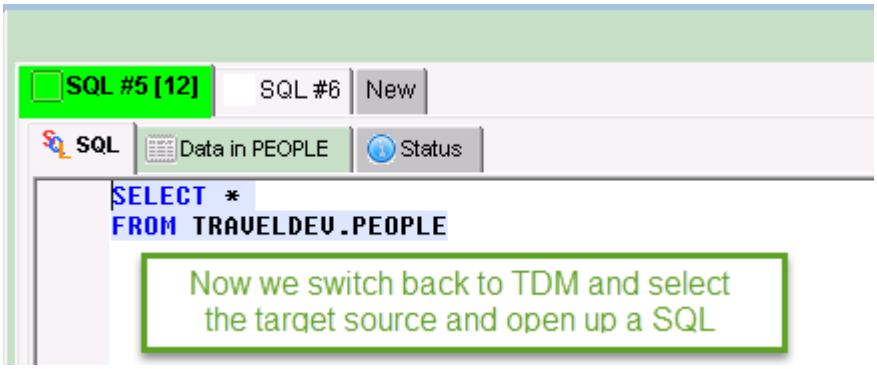
You will need to update the selected JCL proc copy and update the following entries:

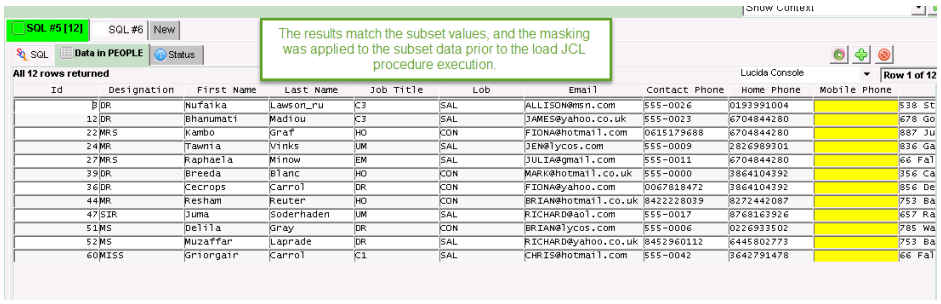
- DB2 subsystem info
- SYSIN
- SYSREC01…SYSREC15

Enter the submit command and press Enter.

<table>
<tr>
<td>


</td>
<td>You will receive a message that the job has been submitted.

If the message that you receive has a MAXCC value of 04 or less, then the job has completed successfully.</td>
</tr>
<tr>
<td>




</td>
<td>Now we go back to GT Data Maker to execute the SQL statement as shown.

You will see that the values that you selected have been masked as necessary.</td>
</tr>
</table>

Best Practices

The following best practices will help you in being successful in masking DB2 datasets.

DB2 Authorizations

Make sure that you have sufficient rights to the DB2 schemas (read/write/alter authorizations), at the same time make sure that you have setup DB2 connect and tested this connection from the system where TDM is installed. Add an ODBC entry to TDM that points to the DB2 subsystem in the mainframe.

Planning

Prior to starting your in-flight masking, it is necessary that you plan the process that you will be following:

- Select the proper entries in the transformation maps.
- Make sure that you have tested your subsets in GT Subset, which you can access from the start menu or via GT DataMaker.
- Proper access to the mainframe with the proper datasets authorizations.

JCL Procedures

Create a copy of the GTXMSKL procedure for a specific subset/transformation map.

Main options

The following two options are key, if you want to just mask the data in-flight with no subset or you could just subset the data with no masking, if you so choose.

- No subset – You need to change the following entry: “SUBDS=NULLFILE”.
- No masking – You need to change the following entry: “MAPDS=NULLFILE”.

Report and Audit files

To make sure that you differentiate the in-flight masking job, it is important that you change the following entry to the type of masking that you are doing.

- Report path – Change the following entry “REPHLQ=GRIDT01.LIB.GTXMSKL”. For example, set that value to “REPHLQ=mypath.lib.inflgt.rpt”.
- Size of files – If you are processing a lot of fields and a lot of data, the size of the files is controlled via the following line in the JCL procedure:

  //* AUDIT REPORT PRIMARY AND SECONDARY SPACE (CYLS)

  // *AUDPRI='1',AUDSEC='1',*

  - Change the value from 1 to at least 10 to provide you with the necessary space for the entries in the report and audit files.

Default Parameters

Here is the list of the “shipped” parameters that are included in the GTXMSKL JCL procedure.

LANGUAGE=EN
AUDIT=ALL
DBUPDATES=Y → *Initially you should set this to N, so that you can see how the job would execute prior to running the job.*
PROGRESSCOUNT=5
COMMIT=1000
SCHEMA=<source schema> → *This is the source schema that will provide the data to be masked*
TARGETSCHEMA=<target schema> → *This the target schema that is scheduled to receive the masked and/or subset schema.*
APPLYSUBSETRULES=Y
LOADPARM1=LOAD DATA LOG NO NOCOPYPEND RESUME YES → *This is the instruction set that will be part of the job card that will be created. Please review the DB2 load parameters, just in case you need to change these entries.*

A more detailed information about all the valid parameters can be obtained from the link below:

https://docops.ca.com/ca-test-data-manager/4-5/en/provisioning-test-data/mainframe-masking-and-subsetting-jobs/program-parameters/gtxmsx-and-gtxmskl-parameters

To obtain additional diagnosis messages when the job executes, you can change the entry below to the value shown, be default this value is set to 1.

DIAGLEVEL=4

Recommended Parameters

These are the additional parameters that you should defined and use in the JCL procedure job.

- HASHTYPE=JAVA – The selection of this parameter allows the masking hash used by the ZOS to be the same as FDM. This is a requirement for consistent masking with FDM.
- LOADPARM1=LOAD DATA LOG NO NOCOPYPEND REPLACE – By using these values for the creation of the DB2 load card, where it will replace all the existing data in the target schema. This assumes that the DDL between source and target schemas are the same.
- PAGELIMIT=200 – If you are masking very large datasets, then it is important that you change report page limit from 50 to at least 200.

Handling large datasets

If you are trying to perform an in-flight masking of more than 20 tables at a time, then you would need to modify the "*GRIDT01.LIB.PROCLIB(GTMSKL)*" template procedure to be able to handle these large datasets. The changes to make are:

Delete section:

You will need to add additional entries based off the entry below, where you will change the DD20 for DD21, etc.

```
//DD20 DD DSN=&LOADHLQ..RECS20,
//       DISP=(MOD,DELETE),SPACE=(TRK,0),
```

```
//      MGMTCLAS=TSO,STORCLAS=TSO
```

Create file section:

After that you will need to the additional files based off the snippet below, where you make a copy starting from DD20, and rename the new section DD21,…, etc.

```
//DD20    DD DSN=&LOADHLQ..RECS20,
//       UNIT=SYSDA,DISP=(NEW,CATLG,CATLG),
//       SPACE=(CYL,(10,10)),
//       DCB=(RECFM=VB,LRECL=31996,BLKSIZE=32000),
//       MGMTCLAS=TSO,STORCLAS=TSO
```

Storage capacity and location:

After you have added the additional files in the delete and create sections of the template, you also need to update the *MGMTCLAS* and *STORCLAS* with the correct volume to use and with enough storage space available.

Another entry that you would need to modify is the "*SPACE=(CYL,(10,10))*", where the values of 10 need to change to at least 100 to make sure that there is enough space for the each of the sequential files, but this might need to larger as needed.

You also need to add the same files in the "*TDMLODDB*" to be able to load the data back into the new DB2 subsystem.

Running Multiple JCL Jobs

It is good practice that you submit a JCL job per given set of tables/subsets based on your masking needs. If you need to generate several sets of masking jobs, it is recommended that you create multiple copies of the JCL procedure and run these JCL procedures in parallel.

Appendix A

This is the "TDMLODDB" JCL procedure that should be placed in the RUNJCL dataset, please don't forget to update the job card info and the DB2 subsystem info.

```
//DB2LODDB JOB (002200000),'DATAMAKER',                                 00001007
//         CLASS=K,MSGCLASS=X,NOTIFY=&SYSUID                            00002008
/*JOBPARM S=CA06                                                        00003000
//*                                                                     00004000
//GTLIB JCLLIB ORDER=PUBLIC.TDM.LIBPROC                                 00005005
//*************************************************************        00150000
//* DB2 LOAD OF MASKED FILES                                            00160000
//*************************************************************        00170000
//* ----------------------------------------*                           00180000
//LOAD     EXEC DSNUPROC,SYSTEM='C10V',COND=(4,LT)                      00190000
//STEPLIB  DD DSN=C10V.PRIVATE.SDSNEXIT,DISP=SHR                        00200000
//         DD DSN=C10V.RUNLIB.LOAD,DISP=SHR                             00210000
//         DD DSN=DB2CA06.DB2A10.SDSNLOAD,DISP=SHR                      00220000
//SYSOUT   DD SYSOUT=*                                                  00230000
//SYSIN    DD DSN=PUBLIC.TDM.LIB.SUBMASK.CARDS,DISP=OLD                 00231005
//SYSREC01 DD DSN=PUBLIC.TDM.LIB.SUBMASK.RECS01,DISP=OLD                00232005
//SYSREC02 DD DSN=PUBLIC.TDM.LIB.SUBMASK.RECS02,DISP=OLD                00233005
//SYSREC03 DD DSN=PUBLIC.TDM.LIB.SUBMASK.RECS03,DISP=OLD                00234005
//SYSREC04 DD DSN=PUBLIC.TDM.LIB.SUBMASK.RECS04,DISP=OLD                00235005
//SYSREC05 DD DSN=PUBLIC.TDM.LIB.SUBMASK.RECS05,DISP=OLD                00236005
//SYSREC06 DD DSN=PUBLIC.TDM.LIB.SUBMASK.RECS06,DISP=OLD                00237005
//SYSREC07 DD DSN=PUBLIC.TDM.LIB.SUBMASK.RECS07,DISP=OLD                00238005
//SYSREC08 DD DSN=PUBLIC.TDM.LIB.SUBMASK.RECS08,DISP=OLD                00239005
//SYSREC09 DD DSN=PUBLIC.TDM.LIB.SUBMASK.RECS09,DISP=OLD                00239105
//SYSREC10 DD DSN=PUBLIC.TDM.LIB.SUBMASK.RECS10,DISP=OLD                00239205
//SYSREC11 DD DSN=PUBLIC.TDM.LIB.SUBMASK.RECS11,DISP=OLD                00239305
//SYSREC12 DD DSN=PUBLIC.TDM.LIB.SUBMASK.RECS12,DISP=OLD                00239405
//SYSREC13 DD DSN=PUBLIC.TDM.LIB.SUBMASK.RECS13,DISP=OLD                00239505
//SYSREC14 DD DSN=PUBLIC.TDM.LIB.SUBMASK.RECS14,DISP=OLD                00239605
//SYSREC15 DD DSN=PUBLIC.TDM.LIB.SUBMASK.RECS15,DISP=OLD                00239705
//SYSTSPRT DD SYSOUT=*                                                  00450000
//SYSPRINT DD SYSOUT=*                                                  00460000
//SYSUT1 DD DSN=&&SYSUT1,                                               00470000
// DISP=(,PASS),                                                        00480000
// SPACE=(4096,(20,20),,,ROUND)                                         00490000
//SORTOUT DD DSN=&&SORTO,                                               00500000
// DISP=(,PASS),UNIT=SYSDA,                                             00510000
// SPACE=(4096,(20,20),,,ROUND)                                         00520000
//SYSMAP  DD DSN=&&SYSMA,                                               00530000
// DISP=(,PASS),UNIT=SYSDA,                                             00540000
// SPACE=(4096,(20,20),,,ROUND)                                         00550000
```

Useful Links

https://docops.ca.com/ca-test-data-manager/4-5/en/mainframe/working-with-db2-data-sources/masking-db2-data-sources/executing-masking-db2-data-sources/mask-and-unload-db2-tables

https://docops.ca.com/ca-test-data-manager/4-5/en/mainframe/working-with-db2-data-sources/subsetting-db2-data/creating-extract-definitions-for-db2-subset

https://docops.ca.com/ca-test-data-manager/4-5/en/mainframe/working-with-db2-data-sources/subsetting-db2-data/executing-db2-subsetting

https://docops.ca.com/ca-test-data-manager/4-5/en/mainframe/working-with-db2-data-sources/subsetting-db2-data/executing-db2-subsetting/db2-subsetting-with-masking

https://docops.ca.com/ca-test-data-manager/4-5/en/mainframe/working-with-db2-data-sources/masking-db2-data-sources/executing-masking-db2-data-sources/mask-and-unload-db2-tables/gtxmskl-parameters