



CA Test Data Manager
Mainframe InFlight Masking
Best Practices Guide

Author : Walter Guerrero

Version: 1.1

Date: 4/20/2018

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.....	23
Useful Links	24

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:

FILTER SEARCH RESULTS					
Search By Product Name...					
PRODUCTS	RELEASE	SERVICE PACK	DOCKER	ADD TO CART	DOWNLOAD
Click the product name for product details for the Release & Service Pack selected					
CA Virtual Test Data Manager Add-On MULTI-PLATFORM	4.5	0000			
CA Test Data Manager Trial MULTI-PLATFORM	4.7	0004			
CA Test Data Manager for Test Data on Demand MULTI-PLATFORM	4.5	0000			
CA Test Data Manager for Masking and Subsetting MULTI-PLATFORM	4.7	0004			
CA Test Data Manager Mainframe IMS Add On MVS	5.4	0007			
CA Test Data Manager for Data Generation and Test Matching MULTI-PLATFORM	4.5	0000			
CA Test Data Manager Data Source Type MULTI-PLATFORM	4.5	0000			
<u>CA Test Data Manager Mainframe DB2 Add On MVS</u>	<u>5.4</u>	<u>0007</u>			

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.

RELEASE	SERVICE PACK	COMPONENT	DATE		
		Search by Component Name...			
5.4.14	0000	<u>CA TEST DATA MANAGER FOR MAINFRAME 5.4.14</u> GEN500000000001207.zip	03/25/2018		
5.4.14	0000	CA TEST DATA MANAGER FILE DEFINITION MANAGER 5.4.14 GEN500000000001267.zip	03/21/2018		
5.4.14	0000	CA TEST DATA MANAGER FILE CONVERSION UTILITY 5.4.14 GEN500000000001270.zip	03/21/2018		

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 and 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.

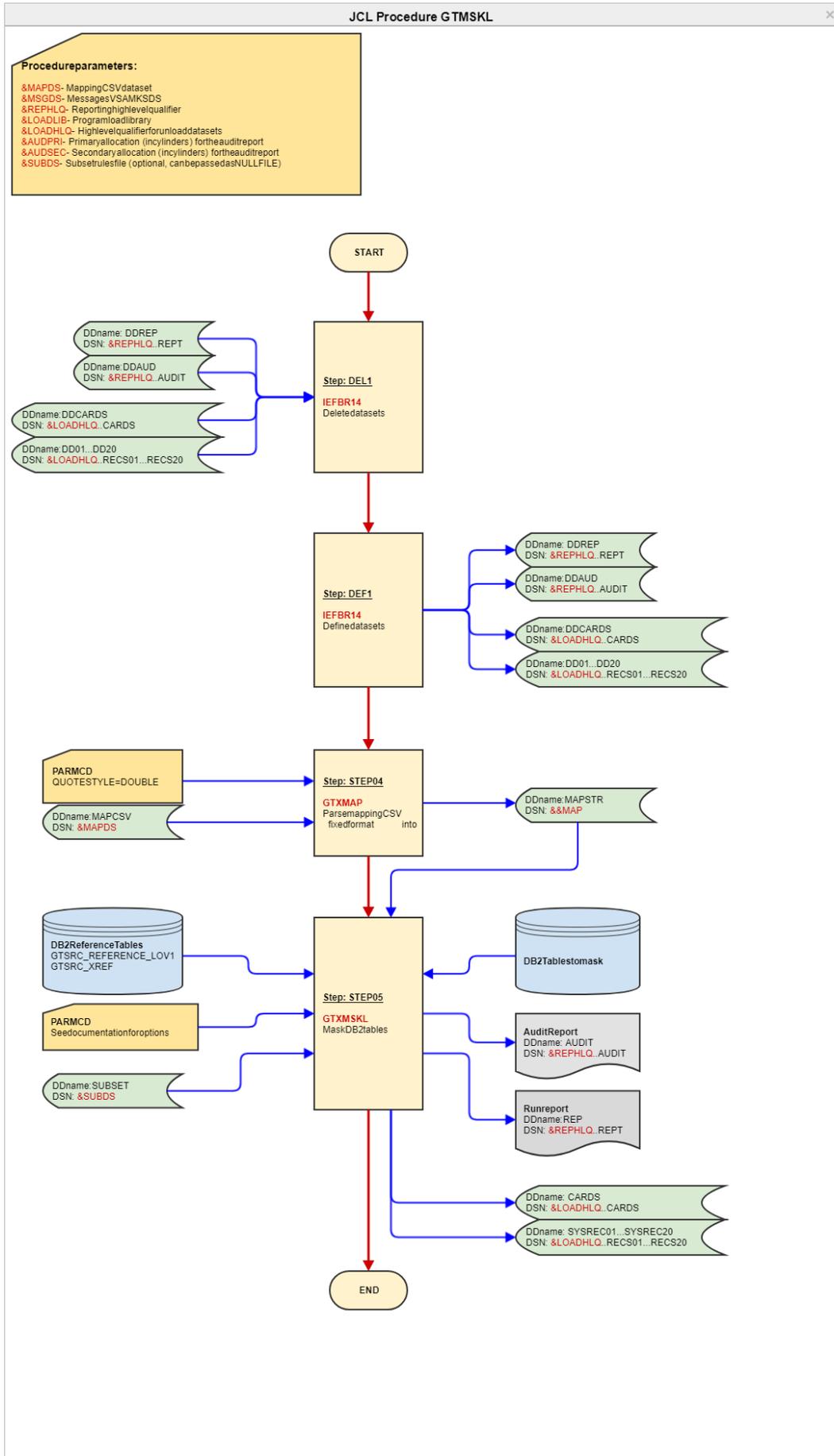
The above programs require the uploaded members in the following datasets:

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.

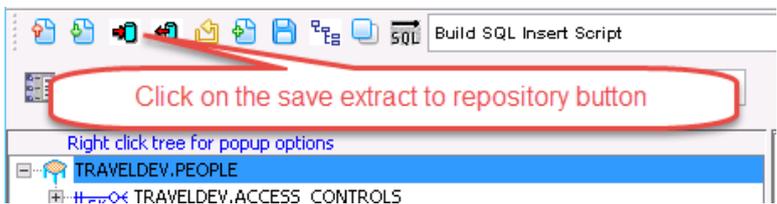
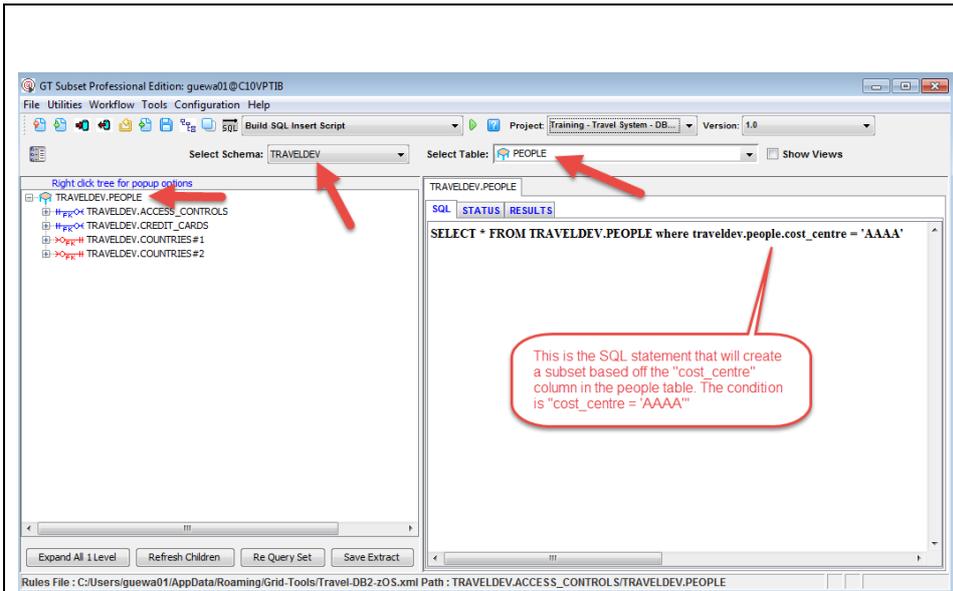


In-Flight Masking Scenario

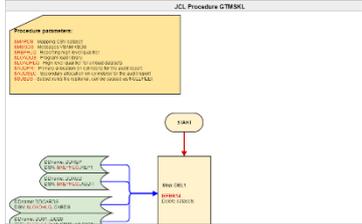
Now we are going to step thru a “*In-flight*” scenario, where we are going to be creating a transformation map, setting up a subset job and attaching the subset job prior to the generation of the transformation map. These files will be uploaded to the mainframe and placed in the two datasets listed above.

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																								
<p>Context: Travel System - DB2 for zOS 1.0 Data Group: Travel System Data Group Data Set: Travel System Data Set Data Pool: Travel subsetting</p> <p>Maintain Projects</p> <p>Rowcount for all levels</p> <table border="1"> <thead> <tr> <th>Table Name</th> <th>Total Rows</th> <th>Excluded Rows</th> <th>Publish Rows</th> </tr> </thead> <tbody> <tr> <td>ACCESS_CONTROLS</td> <td>1</td> <td>0</td> <td>1</td> </tr> <tr> <td>COUNTRIES</td> <td>1</td> <td>0</td> <td>1</td> </tr> <tr> <td>CREDIT_CARDS</td> <td>1</td> <td>0</td> <td>1</td> </tr> <tr> <td>ITINERARIES</td> <td>1</td> <td>0</td> <td>1</td> </tr> <tr> <td>PEOPLE</td> <td>1</td> <td>0</td> <td>1</td> </tr> </tbody> </table>	Table Name	Total Rows	Excluded Rows	Publish Rows	ACCESS_CONTROLS	1	0	1	COUNTRIES	1	0	1	CREDIT_CARDS	1	0	1	ITINERARIES	1	0	1	PEOPLE	1	0	1	<p>Click on the “Maintain Project” Icon.</p> <p>You will see the available projects, and pay close attention to the DB2 for z/OS project.</p> <p>There are two schemas in this scenario:</p> <ul style="list-style-type: none"> - 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.
Table Name	Total Rows	Excluded Rows	Publish Rows																						
ACCESS_CONTROLS	1	0	1																						
COUNTRIES	1	0	1																						
CREDIT_CARDS	1	0	1																						
ITINERARIES	1	0	1																						
PEOPLE	1	0	1																						
<p>CA Test Data Manager - Datamaker CA-czaan01 - 105246 - TDMMAS990</p> <p>Projects Test Data Data Target Data Source Security Data Subset Data Profiler Data Design Data Tools Tools Settings Window Help</p> <p>Design Extracts and Transactions</p> <p>Verify and Prepare Subset Schema</p> <p>Context: Training-Travel-System-DB2 for zOS 1.0 Data Group: Travel System Data Group Data Set: Travel System Data Set Data Pool: Travel System Data Pool</p>	<p>Now, we need to go create the subset that will be use as part of this use case.</p> <p>Select the “Design Extracts and Transactions” menu option.</p>																								



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"

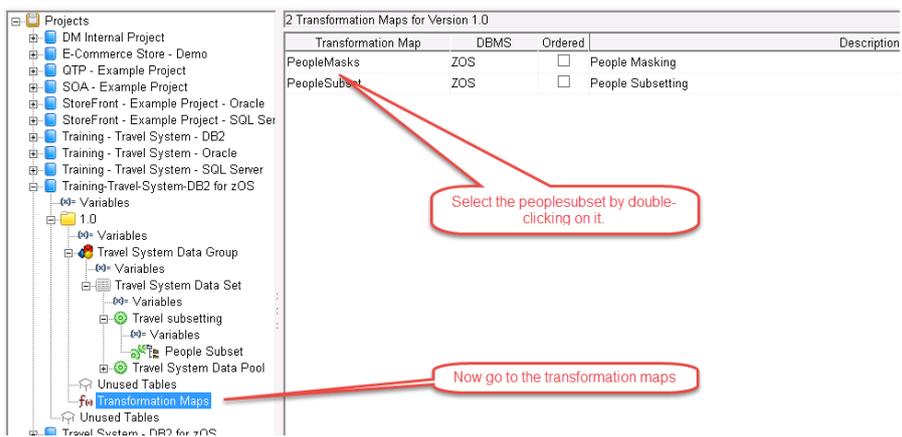
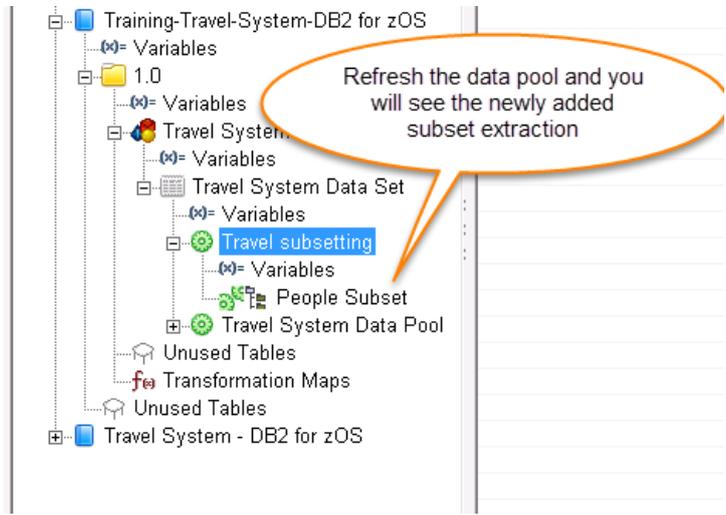


schema is also selected.

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.

Table	Table Name	Column Seq	Column Name	Data type	Mandatory	Transformation	Rel	Key/Null
ITN	ITN	13	HOTEL_COUNT	decimal (10)				
ITN	ITN	14	CAR_COUNT	decimal (10)				
ITN	ITN	15	FLYING_MILES	decimal (10)				
ITN	ITN	16	ITN	decimal (10)				
ITN	ITN	17	ITN	decimal (10)				
ITN	ITN	18	ITN	decimal (10)				
ITN	ITN	19	ITN	decimal (10)				
ITN	ITN	20	HOTEL_COSTS	decimal (10, 2)				
ITN	ITN	21	TRIP_DESC	long varchar				
ITN	ITN	22	AUTHORISATION_ID	decimal (10)				
ITN	ITN	23	CCD_ID	decimal (10)				
ITN	ITN	1	ID	decimal (10)				
ITN	ITN	2	DESIGNATION	varchar (4)				
ITN	ITN	3	FIRST_NAME	varchar (40)		PASSTO/PFIRSTNAME		
ITN	ITN	4	LAST_NAME	varchar (40)		PASSTO/LASTNAME		
ITN	ITN	5	JOB_TITLE	varchar (2)				
ITN	ITN	6	LOB	varchar (2)				
ITN	ITN	7	EMAIL	varchar (40)		EMAIL		
ITN	ITN	8	CONTACT_PHONE	varchar (20)				
ITN	ITN	9	HOME_PHONE	varchar (20)				
ITN	ITN	10	MOBILE_PHONE	varchar (20)		TRANSLATE (023480181159779506		
ITN	ITN	11	ADDRESS	varchar (200)		TRANSLATE (02348018115980773306		
ITN	ITN	12	START_DATE	date				
ITN	ITN	13	TERMINATION_DATE	date				
ITN	ITN	14	NATIONALITY_ID	decimal (10)				
ITN	ITN	15	RESIDENT_ID	decimal (10)				
ITN	ITN	16	COST_CENTRE	varchar (4)				
ITN	ITN	17	PHOTO_FILENAME	varchar (200)				
ITN	ITN	18	AUTHORISATION_ID	decimal (10)				
ITN	ITN	19	EMPRD	decimal (8)				
ITN	ITN	20	SSN	varchar (15)				
ITN	ITN	1	ID	decimal (10)		PASSTO/ID		
ITN	ITN	2	DESIGNATION	varchar (4)				

Save to file

Select the type of file to create.

4.1.0.46 [S12] 3.2H

Excel **CSV - ZOS** CSV - Export Cancel

ERARIES	23	CCD_ID		
PLE	1	ID		
PLE	2	DESIGNATION		
PLE	3	FIRST NAME	varchar (40)	HA

This PC > Desktop > TDM-MF > Upload

Organize New folder

Name	Date modified	Type
PeopleMasks.csv	6/7/2017 9:58 PM	CSV File

File name: PeopleSubSet.csv

Save as type: CSV Files (*.csv)

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.

Val.	Appr.	Table Name	Column Seq	Column Name	Data type	Mandatory	Transformation
<div style="border: 1px solid blue; padding: 5px;"> <p style="text-align: center; margin: 0;">Save Transformation Map</p> <p style="margin: 0;">Do you wish to attach a subset to the transformation map?</p> <p style="text-align: right; margin: 0;">4.1.0.46 [S12] 3.2H</p> <p style="text-align: center; margin: 0;"> <input type="button" value="Yes"/> <input type="button" value="No"/> </p> </div>							
<input type="checkbox"/>		ITINERARIES	23	CCD_ID	decimal (10)	<input type="checkbox"/>	
<input type="checkbox"/>		PEOPLE			decimal (10)	<input type="checkbox"/>	
<input type="checkbox"/>		PEOPLE			varchar (4)	<input type="checkbox"/>	
<input type="checkbox"/>		PEOPLE			varchar (40)	<input type="checkbox"/>	HASHLOV.FIRSTNA
<input type="checkbox"/>		PEOPLE	4	LAST_NAME	varchar (40)	<input type="checkbox"/>	HASHLOV.LASTNA
<input type="checkbox"/>		PEOPLE	5	JOB TITLE	varchar (2)	<input type="checkbox"/>	

Yes, we need to attach the newly created transformation map.

Extract	Description	Saved In Context
<input checked="" type="checkbox"/>	People Subset	Project: Training-Travel-System-DB2 for zOS Version: 1.0 Data Group: Travel System

Select the map and press the Ok button

Choose Subset Condition							
<div style="border: 1px solid blue; padding: 5px;"> <p style="margin: 0;">Subset Condition: Join SQL</p> <p style="text-align: center; margin: 10px 0;"> You are going to use the condition above <input checked="" type="checkbox"/> </p> </div>							
<input type="checkbox"/>		ITINERARIES	22	AUTHORISATION_ID	decimal (10)	<input type="checkbox"/>	
<input type="checkbox"/>		ITINERARIES	23	CCD_ID	decimal (10)	<input type="checkbox"/>	
<input type="checkbox"/>		PEOPLE	1	ID	decimal (10)	<input type="checkbox"/>	
<input type="checkbox"/>		PEOPLE	2	DESIGNATION	varchar (4)	<input type="checkbox"/>	
<input type="checkbox"/>		PEOPLE	3	FIRST_NAME	varchar (40)	<input type="checkbox"/>	HAS

You are going to use the condition above

Val.	Appr.	Table Name	Column Seq	Column Name	Data type	Mandatory	Transformation
<div style="border: 1px solid blue; padding: 5px;"> <p style="text-align: center; margin: 0;">Success</p> <p style="margin: 0;">Save to CSU - files: "C:\Users\Administrator\Desktop\TDM-MF\Upload\PeopleSubset.csv" "C:\Users\Administrator\Desktop\TDM-MF\Upload\PeopleSubset.txt" succeeded!</p> <p style="text-align: right; margin: 0;">4.1.0.46 [S12] 3.2H</p> <p style="text-align: center; margin: 0;"> <input type="button" value="Open Directory?"/> <input type="button" value="OK"/> </p> </div>							
<input type="checkbox"/>		ITINERARIES	23	CCD_ID	decimal (10)	<input type="checkbox"/>	
<input type="checkbox"/>		PEOPLE	1	ID	decimal (10)	<input type="checkbox"/>	

Open the directory

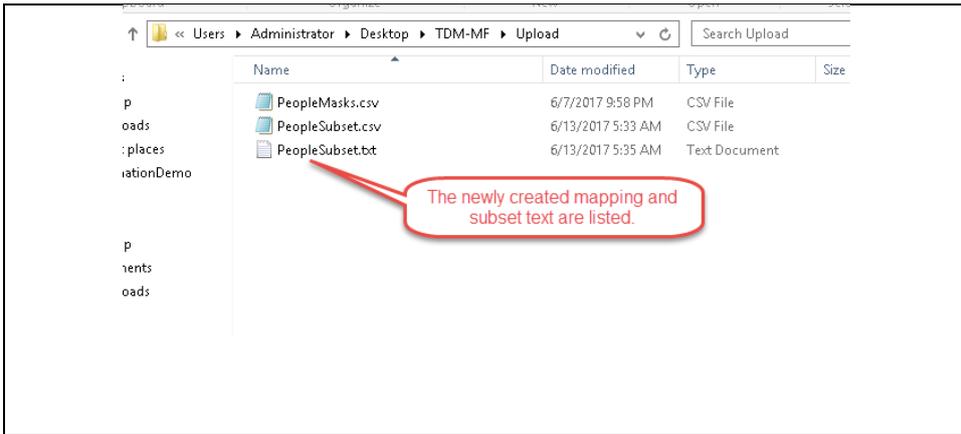
Since there is a subset extract that is part of this data pool, you will be asked if you would like to include the transformation map.

Select the "Yes" button to include the subset extract.

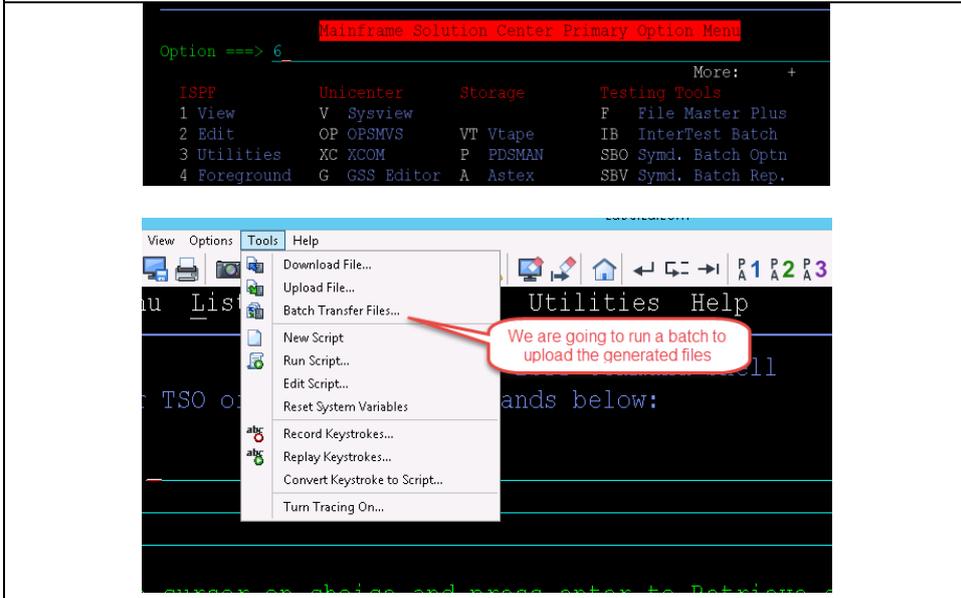
Select the subset extract created previously.

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.



Here are the newly created files.

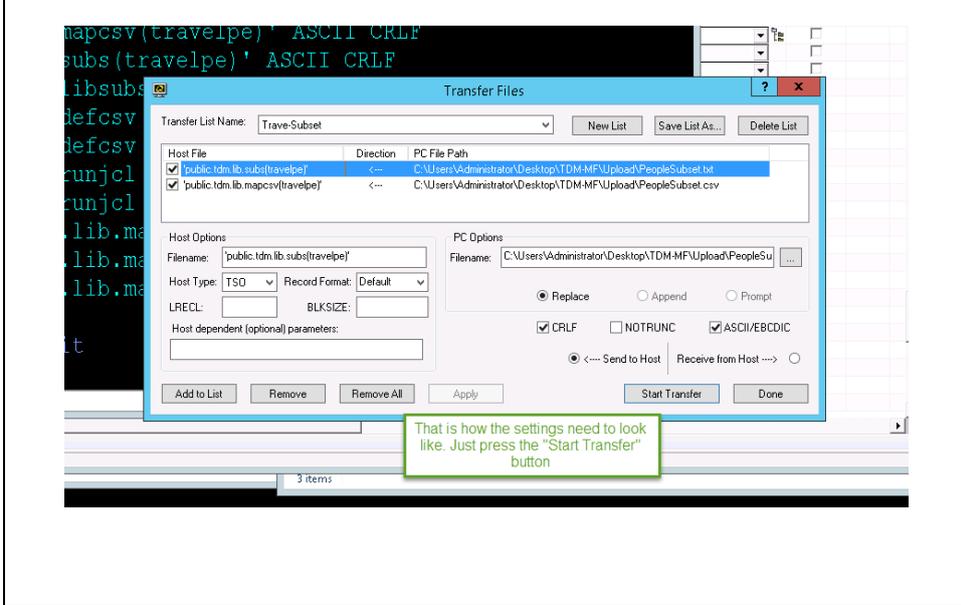


Now you need to connect to the mainframe system using your mainframe credentials.

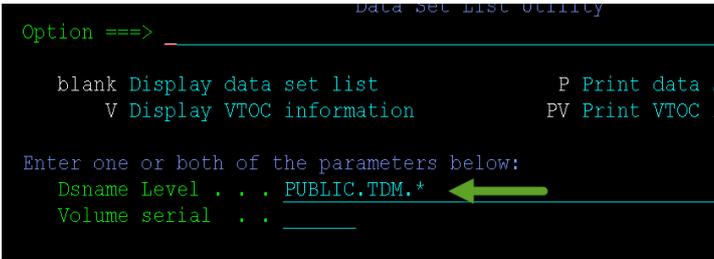
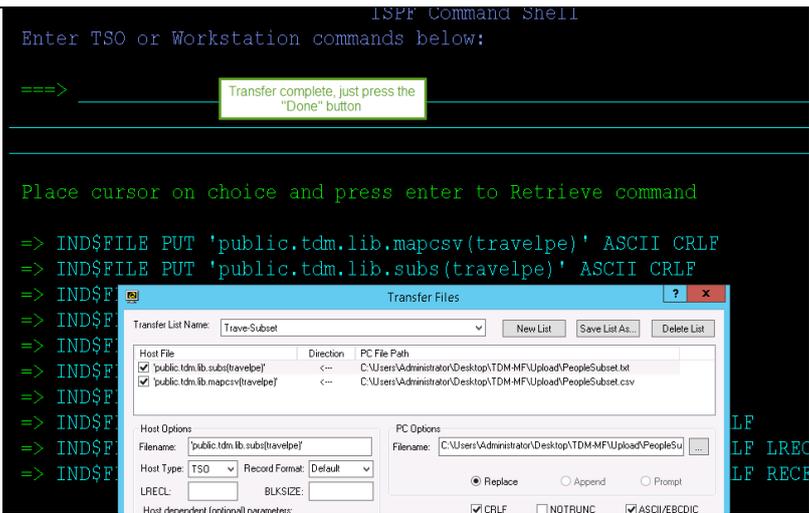
You will need to select the option to enter the “TSO” environment, and enter “ISPF” to start at the base menu.

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.



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 dsname level that you will be starting at will be "PUBLIC.TDM.*" This is just an example, your entries could be starting with "GRIDTO1.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.

```

BROWSE PUBLIC.TDM.LIB.MAPCSV
Command ==>
-----
Name Prompt Size Created
-----
DB2MASK
PEOPLE
TEST
TESTDB
TESTF 33 2012/08/08
TESTSS 10 2016/05/10
TRAVELPE
b

```

```

BROWSE PUBLIC.TDM.LIB.MAPCSV (TRAVELPE) Line 00000000 Col 00
Command ==> Scroll ==>
***** Top of Data *****
"Table", "Column", "Function", "Parm1", "Parm2", "Parm3", "Parm4", "KeepNulls", "D
PEOPLE, EMAIL, EMAIL, , , N, , , , ,
PEOPLE, FIRST_NAME, HASHLOV, FIRSTNAME, , , N, , , , ,
PEOPLE, LAST_NAME, HASHLOV, LASTNAME, , , N, , , , ,
PEOPLE, SSN, RANDSSN, , , N, , , , ,
PEOPLE, MOBILE_PHONE, TRANSLATE, 0123456789, 1155779988, , , N, , , , ,
PEOPLE, ADDRESS, TRANSLATE, 0123456789, 5588773366, , , N, , , , ,
***** Bottom of Data *****

```

```

Command - End
-----
PUBLIC.TDM.LIB.DEFCSV
PUBLIC.TDM.LIB.MAPCSV
PUBLIC.TDM.LIB.RUNJCL
b PUBLIC.TDM.LIB.SUBS
PUBLIC.TDM.LIBPROC
***** End of Data Set list

```

```

BROWSE PUBLIC.TDM.LIB.SUBS
Command ==>
-----
Name Prompt Size Created
-----
DB2SUB 2 2015/11/08 2016/08
b TRAVELPE
**End**

```

```

BROWSE PUBLIC.TDM.LIB.SUBS (TRAVELPE) Line 00000000 Col 001 080
Command ==> Scroll ==> CSR
***** Top of Data *****
TRAVELDEV.COUNTRIES (ID) in (select
TRAVELDEV.COUNTRIES (ID) in (select
TRAVELDEV.PEOPLE cost_centre = '
TRAVELDEV.ACCESS_CONTROLS (PEO_ID) in (se
TRAVELDEV.CREDIT_CARDS (PEO_ID) in (se
***** Bottom of Data *****

```

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.


```

BROWSE PUBLIC.TDM.LIB.RUNJCL(GTXMSKL) - 01.47 Line 0000003
Command ==> submit
//STEP04.PARMCD DD *
QUOTESTYLE=DOUBLE
/*
//STEP05.STEPLIB DD DSN=&LOADLIB,DISP=SHR
// DD DSN=C10V.PRIVATE.SDSNEXIT,DISP=SHR
// DD DSN=C10V.RUNLIB.LOAD,DISP=SHR
// DD DSN=DB2CA06.DB2A10.SDSNLOAD,DISP=SHR
//STEP05.PARMCD DD *

```

Enter submit at the command line and press Enter

```

SCHEMA=TRAVELDEV
TARGETSCHEMA=TRAVELDEV
APPLYSUBSETRULES=Y
IKJ56250I JOB GUEWA01X(JOB01639) SUBMITTED
***

```

```

DSLIST - Data Sets Matching PUBLIC.TDM.*
Command ==>
Command - Enter "/" to select action Message
-----
PUBLIC.TDM.LIB.DEFCSV
PUBLIC.TDM.LIB.GTXMSKL.RPT.AUDIT
PUBLIC.TDM.LIB.GTXMSKL.RPT.REPT
PUBLIC.TDM.LIB.MAPCSV
PUBLIC.TDM.LIB.RUNJCL
PUBLIC.TDM.LIB.SUBMASK.CARDS
PUBLIC.TDM.LIB.SUBMASK.RECS01
PUBLIC.TDM.LIB.SUBMASK.RECS02
PUBLIC.TDM.LIB.SUBMASK.RECS03
PUBLIC.TDM.LIB.SUBMASK.RECS04
PUBLIC.TDM.LIB.SUBMASK.RECS05
PUBLIC.TDM.LIB.SUBMASK.RECS06
PUBLIC.TDM.LIB.SUBMASK.RECS07
PUBLIC.TDM.LIB.SUBMASK.RECS08
PUBLIC.TDM.LIB.SUBMASK.RECS09

```

The report files have been created, as well as the load cards and records

```

BROWSE PUBLIC.TDM.LIB.GTXMSKL.RPT.REPT Line 0000007 Col 001
Command ==> Scroll ==>
00023I Current date used 2017-06-13
00024I High date used 2200-12-31
00025I Low date used 1800-01-01
00026I Language selected EN
99997I Case insensitive HASHLOV = N
99997I Trimmed HASHLOV = N
99997I HASHLOV hashing done using Assembler algorithm
00070I Report page limit set to 50
00122I 12 records written for table COUNTRIES
00122I 12 records written for table COUNTRIES
00122I 12 records written for table PEOPLE
00122I 12 records written for table ACCESS_CONTROLS
00122I 12 records written for table CREDIT_CARDS
00060I Job started at 00:23:35.07 and ended at 00:23:36.35

```

Results in the report file, which tells you everything that took place for the creation of the load card and data file and the masking applied

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.

```

DSLIST - Data Sets Matching PUBLIC.TDM.*
Command ==> _____ Data Set
                                Scroll

Command - Enter "/" to select action      Message
-----
PUBLIC.TDM.LIB.DEFCSV
PUBLIC.TDM.LIB.GTXMSKL.RPT.AUDIT
PUBLIC.TDM.LIB.GTXMSKL.RPT.REPT
PUBLIC.TDM.LIB.MAPCSV
PUBLIC.TDM.LIB.RUNJCL
PUBLIC.TDM.LIB.SUBMASK.CARDS
PUBLIC.TDM.LIB.SUBMASK.RECS01
                                Browsed
b_
↑

```

```

BROWSE PUBLIC.TDM.LIB.GTXMSKL.RPT.AUDIT Line 00000000 Col 053
Command ==> _____ Scroll ==> C
***** Top of Data *****
ID TOOLS DATA MASKING REPORT 13-06-20
-----
column values Mask column      Before and after values
-----
EMAIL                          a@b.c
FIRST_NAME                      ALLISON@msn.com
LAST_NAME                      Nufaika
SSN                             Lawson ru
MOBILE_PHONE                   257677874
                                124881849
                                <null>

```

```

DSLIST - Data Sets Matching PUBLIC.TDM.*
Command ==> _____
Command - Enter "/" to select action
-----
PUBLIC.TDM.LIB.DEFCSV
PUBLIC.TDM.LIB.GTXMSKL.RPT.AUDIT
PUBLIC.TDM.LIB.GTXMSKL.RPT.REPT
PUBLIC.TDM.LIB.MAPCSV
PUBLIC.TDM.LIB.RUNJCL
PUBLIC.TDM.LIB.SUBMASK.CARDS
PUBLIC.TDM.LIB.SUBMASK.RECS01
b_
↓

```

```

BROWSE PUBLIC.TDM.LIB.SUBMASK.CARDS Line 00000000
Command ==> _____ S
***** Top of Data *****
LOAD DATA LOG NO NOCOPYPEND RESUME YES
INDDN SYSRECO1
INTO TABLE TRAVELDEV.COUNTRIES
(
  ID                                POSIT
  DECIMAL
  NAME                              POSIT
  VARCHAR
  GEOGRAPHY                          POSIT
  VARCHAR

```

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.

```

PUBLIC.TDM.LIB.MAFCSV
PUBLIC.TDM.LIB.RUNJCL
PUBLIC.TDM.LIB.SUBMASK.CARDS
PUBLIC.TDM.LIB.SUBMASK.RECS01
PUBLIC.TDM.LIB.SUBMASK.RECS02
PUBLIC.TDM.LIB.SUBMASK.RECS03

```



```

BROWSE PUBLIC.TDM.LIB.SUBMASK.RECS01 Line 00000000 Col 001 080
Command ==> Scroll ==> CSR
***** Top of Data *****
.....Guadeloupe.....OZ..IDA.....
.....Veracruzpe.....OT..ITL.....
.....El Salvador.....SA..SRI.....
.....<..Gabonlvador.....OT..KRW.....
.....@..French Guiana.....ME..HKD.....
.....%.MEench Guiana.....OZ..EGP.....
.....æ..SenegalGuiana.....CA..PRP.....
.....<..MoroccoGuiana.....ME..CYP.....
.....@..FLroccoGuiana.....AS..EGP.....
.....UTroccoGuiana.....ME..THB.....

```

Typical records that been subset and masked.

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.

```

PUBLIC.TDM.LIB.DLPCSV
PUBLIC.TDM.LIB.GTXMSKL.RPT.AUDIT
PUBLIC.TDM.LIB.GTXMSKL.RPT.REPT
PUBLIC.TDM.LIB.MAPCSV
PUBLIC.TDM.LIB.RUNJCL
PUBLIC.TDM.LIB.SUBMASK.CARDS
PUBLIC.TDM.LIB.SUBMASK.RECS01
PUBLIC.TDM.LIB.SUBMASK.RECS02

```

We need to browse the RUNJCL dataset library again.

This time we are going to browse the TDMLODDB member.

_____	KSDSSUB			2015/09/09	07
_____	KSDSXREF			2015/10/09	07
_____	MSGLOAD			2015/02/16	2015/10/05 09
_____	PDSMAN	25	2015/07/30	2015/09/08	04
_____	PEOEXT1L				
_____	PEOEXT1U	97	2017/06/07	2017/06/07	23
_____	RECEIVE	224	2013/08/19	2015/09/09	07
_____	RECEIVEV	114	2013/08/19	2015/09/09	07
_____	SEEDLOAD	41	2012/10/31	2015/11/11	09
_____	TDMLODDB	41	2016/08/09	2017/06/12	17
_____	TMSKCA	42	2015/07/21	2015/09/09	07

Now we need to browse the following JCL procedure, which will be performing the upload

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

```

// * DB2 LOAD OF MASKED FILES
// *****
// * -----*
//LOAD EXEC DSNUPROC, SYSTEM='C10V', COND=(4,LT)
//STEPLIB DD DSN=C10V.PRIVATE.SDSNEXIT, DISP=SHR
// DD DSN=C10V.RUNLIB.LOAD, DISP=SHR
// DD DSN=DB2CA06.DB2A10.SDSNLOAD, DISP=SHR
//SYSOUT DD SYSOUT=*
//SYSIN DD DSN=PUBLIC.TDM.LIB.SUBMASK.CARDS, DISP=OLD
//SYSREC01 DD DSN=PUBLIC.TDM.LIB.SUBMASK.RECS01, DISP=OLD
//SYSREC02 DD DSN=PUBLIC.TDM.LIB.SUBMASK.RECS02, DISP=OLD

```

These are the parameters that will be executed as part of the DB2 DSNUPROC taking into account the 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...SYSRE C15

```

BROWSE PUBLIC.TDM.LIB.RUNJCL(TDMLODDB) - 01.05 Line 00000000 Col
Command ==> submit
***** Top of Data *****
//TDMLODDB JOB (129400000), 'DATAMAKER',
// CLASS=K,MSGCLASS=X,NOTIFY=GUEWA01
/*JOBPARM S=CA06
//*
//GTLIB JCLLIB ORDER=PUBLIC.TDM.LIBPROC
//*****
//* DB2 LOAD OF MASKED FILES
//*****

```

Enter the "submit" command and press Enter

Enter the submit command and press Enter.

```
// DD DSN=C10V.RUNLIB.LOAD,DISP=SHR
// DD DSN=DB2CA06.DB2A10.SDSNLOAD,DISP=SHR
//SYSOUT DD SYSOUT=*
//SYSIN DD DSN=PUBLIC.TDM.LIB.SUBMASK.CARDS,DISP=OLD
IKJ56250I JOB TDML0DB (JOB01642) SUBMITTED
***
_
```

The job was submitted

```
00.38.22 JOB01642 $HASP165 TDML0DB ENDED AT USILCA06 MAXCC=0004 CN(INTERNAL)
***
_
```

You will get the message above about the job completing. Any MAXCC value below 0004 is good.

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.



The results match the subset values, and the masking was applied to the subset data prior to the load JCL procedure execution.

Id	Designation	First Name	Last Name	Job Title	Lob	Email	Contact Phone	Home Phone	Mobile Phone
8DR	Nufatka	Lawson_ru	C3	SAL	ALLISON@msn.com	555-0026	0193991004		538 ST
12DR	Bhanumati	Madiou	C3	SAL	JAMES@yahoo.co.uk	555-0023	6704844280		678 GO
22MR S	Kambo	grat	HO	CON	FIONA@hotmail.com	0615179688	6704844280		887 JU
24MR	Tawita	Winks	UM	SAL	JEN@lycos.com	555-0009	2826989301		896 GA
27MR S	Raphaella	Winos	EM	SAL	DULIA@gmail.com	555-0011	6704844280		66 FAI
39DR	Breeda	Blanc	HO	CON	MARK@hotmail.co.uk	555-0000	3864104392		356 CA
36DR	Cecrops	Carrol	DR	CON	FIONA@yahoo.com	0067818472	3864104392		856 DE
44MR	Resham	Reuter	HO	CON	BRIAN@hotmail.co.uk	8422228039	8272442087		753 BA
47SIR	Juma	Soderhaden	UM	SAL	RICHARD@aol.com	555-0017	8768163926		657 RA
51MS	DeTilla	Gray	DR	CON	BRIAN@lycos.com	555-0006	0226933502		785 WA
52MS	Muzaffar	Laprade	DR	SAL	RICHARD@yahoo.co.uk	8422960112	6445803773		753 BA
60MISS	Griorgair	Carrol	C1	SAL	CHRIS@hotmail.com	555-0042	3642791478		66 FAI

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.

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”.

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

LOADPARAM1=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

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/provisioning-test-data/using-for-mainframe>

<https://docops.ca.com/ca-test-data-manager/4-5/en/provisioning-test-data/mainframe-masking-and-subsetting-jobs>

<https://docops.ca.com/ca-test-data-manager/4-5/en/provisioning-test-data/mainframe-masking-and-subsetting-jobs/mask-and-unload-db2-tables-db2>

<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>