GFS Data Pruning and De-identification

Design Document



Author: Innovacx

Creation Date: June 04, 2018

Last Update: December 28, 2018

1. **Title, Subject, Last Updated Date, Reference Number** , **and** **version** are marked by a Word bookmark so that they can be easily reproduced in the header and footer of documents. When you change any of these values, be careful not to accidentally delete the bookmark. **You can make bookmarks visible by selecting Tools->Options…View and checking the Bookmarks option in the Show region.**
2. **Title, Subject, Last Updated Date, Reference Number** , **and** **version** are marked by a Word bookmark so that they can be easily reproduced in the header and footer of documents. When you change any of these values, be careful not to accidentally delete the bookmark. **You can make bookmarks visible by selecting Tools->Options…View and checking the Bookmarks option in the Show region.**
3. To add additional approval lines, press [Tab] from the last cell in the table above.

**Approvers:**

|  |  |
| --- | --- |
| Maroun Azzi |  |
| Vivek Jaiswal |  |

1. You can delete any elements of this cover page that you do not need for i ur document.

# Document of Control

## Change Control

2

| Date | Author | Version | Reference of changes |
| --- | --- | --- | --- |
| 4th June, 18 | Innovacx | 0.1 | 1st Draft |
| 12th June | Innovacx | 0.2 | 2nd Draft |
| 18th June | Innovacx | 0.3 | 3rd Draft |
| 21st June | Innovacx | 0.4 | Incorporated Feedback  Added more details |
| 22nd June | Innovacx | 0.5 | Incorporated Feedback  Added more details |
| 1ST July | Innovacx | 0.6 | Incorporated Feedback  Added more details |
| 2nd July | Innovacx | 0.7 | Incorporated Feedback  Added more details |
| 9th July | Innovacx | 0.8 | Incorporated Feedback  Added more details |
| 17th July | Innovacx | 0.9 | Incorporated Feedback  Added more details with respect to the comments given by Girish and H. Tran |
| 28th December | Innovacx | 1.0 | Incorporating Enhancements |

Contents

[1 Document of Control 2](#_Toc518214413)

[1.1 Change Control 2](#_Toc518214414)

[1 Requirement summary 3](#_Toc518214415)

[2 High Level Design Approach 4](#_Toc518214416)

[3 Detail Level Design Approach 5](#_Toc518214417)

[3.1 Database Copy from Production to Test 5](#_Toc518214418)

[3.2 Data Purging 5](#_Toc518214419)

[3.2.1 Export Activity 5](#_Toc518214420)

[3.2.2 Disabling Indexes 9](#_Toc518214421)

[3.2.3 Import activity 9](#_Toc518214422)

[4. Data Masking 12](#_Toc518214423)

[4.1 Requirement Summary 12](#_Toc518214424)

[4.2 Identified GFS attributes to be Masked 12](#_Toc518214425)

[5 Analyzing the Imported tables 13](#_Toc518214426)

[6 Issues 14](#_Toc518214427)

[6.1 Open Issues 14](#_Toc518214428)

[6.2 Closed Issues 14](#_Toc518214429)

# 2 Requirement Summary

As per the input from the NRMA team, the data should be consistent between the GFS and GMS systems after Pruning and the Masking activities. This helps the respective testing team to accomplish end to end testing. Moreover, the team ideally be performing testing on last one year’s data. Given the volume of data, it is very high even if we consider retaining data for one year. To address this required no of members are identified from all the available members from the GMS system and the corresponding data would be retained in the related to Customer, Invoice and Payment modules. The GL and XLA modules’ tables are retained based on the values provided for the Jenkins global parameters DID\_GFS\_PRUN\_FROM\_DATE and DID\_GFS\_PRUN\_TO\_DATE.

# 3 High Level Design Approach

Innovacx proposes below approach

**Step 1:** Clone the EBS production database to Test database. This will be ‘as is’ process and will be performed by NRMA DBA team using existing assets. This cloning may not be required when NRMA feels that the existing data available in the Test database will suffice for their testing.

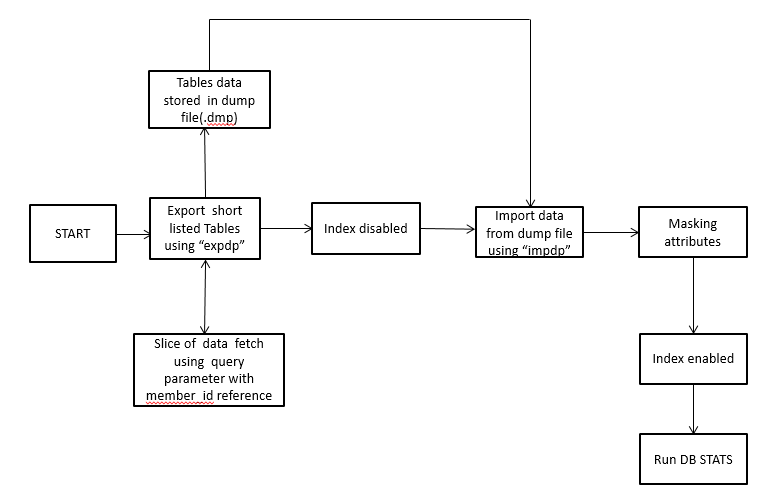
**Step 2**: The short-listed tables will be exported from the Test database using Oracle’s Export utility (EXPDP). Only the slice of data which needs to be retained will be exported. A flat file will be provided from GMS. This file consists Member\_ID information which has to be deleted. For the type contact CSN from S\_CONTACT will be populated and for the type organization OU\_NUM from S\_ORG\_EXT will be populated in that flat file. This flat file will be loaded to a custom table XX\_SBL\_REF. This table is joined with the ATTRIBUTE2 column in the HZ\_CUST\_ACCOUNTS table and CUST\_ACCOUNT\_ID of all the members which are not in the custom table will be loaded into another custom table XX\_CUST\_ACCOUNT\_ID. The identified tables in the Customer, Invoice, Payment modules schema are mapped with XX\_CUST\_ACCOUNT\_ID and the resultant data will be exported. The GL and XLA tables will be retained based on the From Date and To Date values provided in the XX\_DATE\_REF table. This export activity will be carried out separately for all the identified tables.

**Step 3**: To achieve the best performance, the indexes of the identified tables will also be dropped in test database.

**Step 4**: The exported files will now be imported to the target tables using Oracle’s Import utility (IMPDP). The data is imported after truncating the table.

**Step 5**: Create indexes dropped in #3 and gather DB statistics.

**High level solution specification diagram**



# 4 Detail Level Design Approach



## Database Copy from Production to Test

The 1st step for creating the EBS test database is copying the production database to Test database. We assume that this will be done as database copy like Db export to Db import or any of standard oracle utility. Innovacx recommend that this will be a ‘as is process’ and will continue. This step is prerequisite for rest of the operations, so will not be included in Innovacx deliverance.

## Data Purging

The fundamental goal is to keep data in sync with GMS system after purging the tables. So that the size of the EBS tables & database is also reduced to a manageable size. However, if we use delete DML on these tables, it will take large amount of time and will consume resources. To avoid any costly DML, our approach is to export the slice of data which is to be retained from the identified tables, truncate them and import.

### Export Activity

Once the production database is copied to Test Database, The Jenkins jobs invokes the script which leverages the EXPDP utility and exports the slice of data from the short-listed tables from the Test database and store in the form of dump files. This export activity would be carried out separately for each individual table.

As per the primary objective the data in GFS and GMS should be in sync. To achieve this the flat file with member data will be shared by GMS which will be loaded to a custom table (XX\_SBL\_REF). The attribute CSN from S\_CONTACT table in the GMS and OU\_NUM from S\_ORG\_EXT is mapped to Attribute2 of HZ\_CUST\_ACCOUNTS IN GFS in the existing NRMA system. The HZ\_CUST\_ACCOUNTS is joined with the custom table to strip the required data. All the subsequent tables are stripped similarly by joining with the corresponding tables. The join conditions are enlisted in the table below.

The pruning criteria for Customer, Invoice and Payment modules is only the members identified from GMS and the pruning criteria for GL and XLA modules is based on the From Date and To Date values provided in the XX\_DATE\_REF table. The data to the XX\_DATE\_REF table will be populated by the values provided by NRMA in the Jenkins global variables. There is a definite chance that some Orphan records may exists in the identified tables. However, this is a conscious requirement from NRMA.

Filters for Slicing of Data

|  |  |
| --- | --- |
| **Module** | **Filter** |
| Customer, Invoice, Payments | All the Members identified from GMS system |
| GL, XLA | From\_Date & To\_Date |

🡪 Parameters being used in EXPDP:

|  |  |
| --- | --- |
| **PARAMETERS** | **USAGE** |
| DIRECTORY | Directory in which Dump, log files have to be created |
| PARALLEL | No of parallel threads to be executed |
| TABLES | Table name that is intended to export |
| DUMPFILE | The name of the dump file |
| LOGFILE | The name of the log file |
| QUERY | Data specific filtering is implemented through this clause |

**Pseudo Code of Export Commands:**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Seq no** | **Module** | **Table** | **QUERY** | **Row Count(NFT Environment)** |
| 1 | Customer | HZ\_CUST\_ACCOUNTS | QUERY=AR.HZ\_CUST\_ACCOUNTS:"WHERE ATTRIBUTE2 IN (SELECT SIEB\_CSN FROM NRMADID.XX\_SBL\_REF)” | 2849430 |
| 2 | Customer | HZ\_CUST\_ACCT\_SITES\_ALL | QUERY=AR.HZ\_CUST\_ACCT\_SITES\_ALL:"WHERE CUST\_ACCOUNT\_ID IN (SELECT CUST\_ACCOUNT\_ID FROM NRMADID.XX\_CUST\_ACCOUNT\_ID)" | 4493485 |
| 3 | Customer | HZ\_CUST\_SITE\_USES\_ALL | QUERY=AR.HZ\_CUST\_SITE\_USES\_ALL:"WHERE CUST\_ACCT\_SITE\_ID IN (SELECT SITE.CUST\_ACCT\_SITE\_ID  FROM AR.HZ\_CUST\_ACCT\_SITES\_ALL SITE, NRMADID.XX\_CUST\_ACCOUNT\_ID REF  WHERE SITE.CUST\_ACCOUNT\_ID = REF.CUST\_ACCOUNT\_ID)" | 7329200 |
| 4 | Invoice | RA\_CUSTOMER\_TRX\_ALL | QUERY=AR.RA\_CUSTOMER\_TRX\_ALL:"WHERE BILL\_TO\_CUSTOMER\_ID IN (SELECT RA.BILL\_TO\_CUSTOMER\_ID  FROM AR.RA\_CUSTOMER\_TRX\_ALL RA, NRMADID.XX\_CUST\_ACCOUNT\_ID REF  WHERE RA.BILL\_TO\_CUSTOMER\_ID = REF.CUST\_ACCOUNT\_ID)" | 8709233 |
| 5 | Invoice | RA\_CUSTOMER\_TRX\_LINES\_ALL | QUERY=AR.RA\_CUSTOMER\_TRX\_LINES\_ALL:"WHERE CUSTOMER\_TRX\_ID IN (SELECT RA.CUSTOMER\_TRX\_ID  FROM AR.RA\_CUSTOMER\_TRX\_ALL RA, NRMADID.XX\_CUST\_ACCOUNT\_ID REF  WHERE RA.BILL\_TO\_CUSTOMER\_ID = REF.CUST\_ACCOUNT\_ID)" | 38244873 |
| 6 | Invoice | RA\_CUST\_TRX\_LINE\_GL\_DIST\_ALL | QUERY=AR.RA\_CUST\_TRX\_LINE\_GL\_DIST\_ALL:"WHERE EXISTS (SELECT RA.CUSTOMER\_TRX\_ID  FROM AR.RA\_CUSTOMER\_TRX\_ALL RA, NRMADID.XX\_CUST\_ACCOUNT\_ID REF  WHERE RA.BILL\_TO\_CUSTOMER\_ID = REF.CUST\_ACCOUNT\_ID  AND RA.CUSTOMER\_TRX\_ID = ku$.CUSTOMER\_TRX\_ID)" | 473413359 |
| 7 | Payments | AR\_CASH\_RECEIPTS\_ALL | QUERY=AR.AR\_CASH\_RECEIPTS\_ALL:"WHERE PAY\_FROM\_CUSTOMER IN (SELECT CUST\_ACCOUNT\_ID FROM NRMADID.XX\_CUST\_ACCOUNT\_ID)" | 7293109 |
| 8 | Payments | AR\_CASH\_RECEIPT\_HISTORY\_ALL | QUERY=AR.AR\_CASH\_RECEIPT\_HISTORY\_ALL:"WHERE CASH\_RECEIPT\_ID IN (SELECT CASH\_RECEIPT\_ID FROM AR.AR\_CASH\_RECEIPTS\_ALL AR, NRMADID.XX\_CUST\_ACCOUNT\_ID REF WHERE AR.PAY\_FROM\_CUSTOMER = REF.CUST\_ACCOUNT\_ID)" | 7337405 |
| 9 | Payments | AR\_RECEIVABLE\_APPLICATIONS\_ALL | QUERY=AR.AR\_RECEIVABLE\_APPLICATIONS\_ALL:"WHERE CASH\_RECEIPT\_ID IN (SELECT CASH\_RECEIPT\_ID FROM AR.AR\_CASH\_RECEIPTS\_ALL AR, NRMADID.XX\_CUST\_ACCOUNT\_ID REF WHERE AR.PAY\_FROM\_CUSTOMER = REF.CUST\_ACCOUNT\_ID)" | 23489396 |
| 10 | Payments | AR\_PAYMENT\_SCHEDULES\_ALL | QUERY=AR.AR\_PAYMENT\_SCHEDULES\_ALL:"WHERE CUSTOMER\_ID IN (SELECT CUST\_ACCOUNT\_ID FROM NRMADID.XX\_CUST\_ACCOUNT\_ID)" | 15992320 |
| 11 | GL | GL\_JE\_HEADERS | GL\_JE\_HEADERS:  QUERY=GL.GL\_JE\_HEADERS:" WHERE EXISTS(SELECT GHEAD.JE\_HEADER\_ID FROM  GL.GL\_PERIODS GP, GL.GL\_JE\_HEADERS GHEAD, NRMADID.XX\_DATE\_REF  WHERE  GHEAD.PERIOD\_NAME = GP.PERIOD\_NAME  AND GHEAD.JE\_HEADER\_ID = ku$.JE\_HEADER\_ID  AND TO\_DATE(GP.START\_DATE, 'DD-MON-YY') >= TO\_DATE(FROM\_DATE,'DD-MON-YY')  AND TO\_DATE(GP.END\_DATE, 'DD-MON-YY') <= TO\_DATE(TO\_DATE,'DD-MON-YY'))" | 84359428 |
| 12 | GL | GL\_JE\_LINES | QUERY=GL.GL\_JE\_LINES:" WHERE JE\_HEADER\_ID IN(SELECT GLINE.JE\_HEADER\_ID FROM  GL.GL\_JE\_LINES GLINE, GL.GL\_PERIODS GP, GL.GL\_JE\_HEADERS GHEAD, NRMADID.XX\_DATE\_REF  WHERE  GLINE.JE\_HEADER\_ID = GHEAD.JE\_HEADER\_ID  AND GHEAD.PERIOD\_NAME = GP.PERIOD\_NAME  AND TO\_DATE(GP.START\_DATE, 'DD-MON-YY') >= TO\_DATE(FROM\_DATE,'DD-MON-YY')  AND TO\_DATE(GP.END\_DATE, 'DD-MON-YY') <= TO\_DATE(TO\_DATE,'DD-MON-YY'))" | 249423764 |
| 13 | GL | GL\_IMPORT\_REFERENCES | QUERY=GL.GL\_IMPORT\_REFERENCES:" WHERE EXISTS(SELECT GHEAD.JE\_HEADER\_ID FROM  GL.GL\_PERIODS GP, GL.GL\_JE\_HEADERS GHEAD, NRMADID.XX\_DATE\_REF  WHERE  GHEAD.PERIOD\_NAME = GP.PERIOD\_NAME  AND GHEAD.JE\_HEADER\_ID=ku$.JE\_HEADER\_ID  AND TO\_DATE(GP.START\_DATE, 'DD-MON-YY') >= TO\_DATE(FROM\_DATE,'DD-MON-YY')  AND TO\_DATE(GP.END\_DATE, 'DD-MON-YY') <= TO\_DATE(TO\_DATE,'DD-MON-YY'))" | 294410525 |
| 14 | XLA | XLA\_AE\_HEADERS | QUERY=XLA.XLA\_AE\_HEADERS:"WHERE ACCOUNTING\_DATE BETWEEN (SELECT FROM\_DATE FROM NRMADID.XX\_DATE\_REF)  AND (SELECT TO\_DATE FROM NRMADID.XX\_DATE\_REF)" | 101270777 |
| 15 | XLA | XLA\_AE\_LINES | QUERY=XLA.XLA\_AE\_LINES:"WHERE ACCOUNTING\_DATE BETWEEN (SELECT FROM\_DATE FROM NRMADID.XX\_DATE\_REF) AND (SELECT TO\_DATE FROM NRMADID.XX\_DATE\_REF)" | 286212149 |
| 16 | XLA | XLA\_DISTRIBUTION\_LINKS | QUERY=XLA.XLA\_DISTRIBUTION\_LINKS:"WHERE (AE\_HEADER\_ID, AE\_LINE\_NUM) IN  (SELECT AE\_HEADER\_ID, AE\_LINE\_NUM  FROM XLA.XLA\_AE\_LINES  WHERE ACCOUNTING\_DATE BETWEEN (SELECT FROM\_DATE FROM NRMADID.XX\_DATE\_REF) AND (SELECT TO\_DATE FROM NRMADID.XX\_DATE\_REF))" | 1512490701 |
| 17 | XLA | XLA\_EVENTS | QUERY=XLA.XLA\_EVENTS:"WHERE TRANSACTION\_DATE BETWEEN (SELECT FROM\_DATE FROM NRMADID.XX\_DATE\_REF) AND (SELECT TO\_DATE FROM NRMADID.XX\_DATE\_REF)" | 116413833 |

### Dropping Indexes

As part of the Jenkins pipeline, upon successful execution of the export activity, a script will be invoked by Jenkins which is responsible for dropping the indexes of the identified tables.

Sample Disabling Command:

DROP index INDEX\_NAME;

### Import activity

As part of the Jenkins pipeline, upon successful execution of the previous step, a script will be invoked by Jenkins which is responsible for Importing the data from the dump files to the respective tables.

|  |  |
| --- | --- |
| **PARAMETERS** | **USAGE** |
| DIRECTORY | Directory in which Dump, log files have to be created |
| PARALLEL | No of parallel threads to be executed |
| TABLES | Table name that is intended to export |
| DUMPFILE | The name of the dump file |
| LOGFILE | The name of the log file |
| TABLE\_EXISTS\_ACTION | **APPEND**: This option appends the data from the data dump. The extra rows in the dump will be appended to the table and the existing data remains unchanged.  **TRUNCATE**: This option Truncates the existing rows in the table and insert the selected rows from the dump  **REPLACE**: This option Drops the current table and create the table as it is in the dump file. Both SKIP and REPLACE options are not valid if you set the CONTENT=DATA\_ONLY for the IMPDP.  **SKIP**: Default value for this parameter is SKIP. This parameter is exactly same as the IGNORE=Y option in conventional import utility |

**Sample IMPDP Command:**

**impdp** PARALLEL=4 DUMPFILE= HZ\_CUST\_ACCOUNTS.dmp DIRECTORY= EBSDUMP TABLES = APPS.HZ\_CUST\_ACCOUNTS LOGFILE= HZ\_CUST\_ACCOUNTS.log

**Pseudo Code of Import Commands:**

|  |  |  |  |
| --- | --- | --- | --- |
| **Seq no** | **Module** | **Table** | **Import Command** |
| 1 | Customer | HZ\_CUST\_ACCOUNTS | **impdp** PARALLEL=4 DUMPFILE= HZ\_CUST\_ACCOUNTS.dmp DIRECTORY= EBSDUMP TABLES = APPS.HZ\_CUST\_ACCOUNTS LOGFILE= HZ\_CUST\_ACCOUNTS.log |
| 2 | Customer | HZ\_CUST\_ACCT\_SITES\_ALL | **impdp** PARALLEL=4 DUMPFILE= HZ\_CUST\_ACCT\_SITES\_ALL.dmp DIRECTORY= EBSDUMP TABLES = APPS.HZ\_CUST\_ACCT\_SITES\_ALL LOGFILE= HZ\_CUST\_ACCT\_SITES\_ALL.log |
| 3 | Customer | HZ\_CUST\_SITE\_USES\_ALL | **impdp** PARALLEL=4 DUMPFILE= HZ\_CUST\_SITE\_USES\_ALL.dmp DIRECTORY= EBSDUMP TABLES = APPS.HZ\_CUST\_SITE\_USES\_ALL LOGFILE= HZ\_CUST\_SITE\_USES\_ALL.log |
| 4 | Invoice | RA\_CUSTOMER\_TRX\_ALL | **impdp** PARALLEL=4 DUMPFILE= RA\_CUSTOMER\_TRX\_ALL.dmp DIRECTORY= EBSDUMP TABLES = APPS.RA\_CUSTOMER\_TRX\_ALL LOGFILE= RA\_CUSTOMER\_TRX\_ALL.log |
| 5 | Invoice | RA\_CUSTOMER\_TRX\_LINES\_ALL | **impdp** PARALLEL=4 DUMPFILE= RA\_CUSTOMER\_TRX\_LINES\_ALL.dmp DIRECTORY= EBSDUMP TABLES = APPS.RA\_CUSTOMER\_TRX\_LINES\_ALL LOGFILE= RA\_CUSTOMER\_TRX\_LINES\_ALL.log |
| 6 | Invoice | RA\_CUSTOMER\_TRX\_LINE\_GL\_DIST\_ALL | **impdp** PARALLEL=4 DUMPFILE= RA\_CUSTOMER\_TRX\_LINE\_GL\_DIST\_ALL.dmp DIRECTORY= EBSDUMP TABLES = APPS.RA\_CUSTOMER\_TRX\_LINE\_GL\_DIST\_ALL LOGFILE= RA\_CUSTOMER\_TRX\_LINE\_GL\_DIST\_ALL.log |
| 7 | Payments | AR\_CASH\_RECEIPTS\_ALL | **impdp** PARALLEL=4 DUMPFILE= AR\_CASH\_RECEIPTS\_ALL.dmp DIRECTORY= EBSDUMP TABLES = APPSAR\_CASH\_RECEIPTS\_ALL LOGFILE= AR\_CASH\_RECEIPTS\_ALL.log |
| 8 | Payments | AR\_CASH\_RECEIPTS\_HISTORY\_ALL | **impdp** PARALLEL=4 DUMPFILE= AR\_CASH\_RECEIPTS\_HISTORY\_ALL.dmp DIRECTORY= EBSDUMP TABLES = APPS.AR\_CASH\_RECEIPTS\_HISTORY\_ALL LOGFILE= AR\_CASH\_RECEIPTS\_HISTORY\_ALL.log |
| 9 | Payments | AR\_RECEIVABLE\_APPLICATIONS\_ALL | **impdp** PARALLEL=4 DUMPFILE= AR\_RECEIVABLE\_APPLICATIONS\_ALL.dmp DIRECTORY= EBSDUMP TABLES = APPS.AR\_RECEIVABLE\_APPLICATIONS\_ALL LOGFILE= AR\_RECEIVABLE\_APPLICATIONS\_ALL.log |
| 10 | Payments | AR\_PAYMENT\_SCHEDULES\_ALL | **impdp** PARALLEL=4 DUMPFILE= AR\_PAYMENT\_SCHEDULES\_ALL.dmp DIRECTORY= EBSDUMP TABLES = APPS.AR\_PAYMENT\_SCHEDULES\_ALL LOGFILE= AR\_PAYMENT\_SCHEDULES\_ALL.log |
| 11 | GL | GL\_JE\_HEADERS | **impdp** PARALLEL=4 DUMPFILE= GL\_JE\_HEADERS.dmp DIRECTORY= EBSDUMP TABLES = APPS.GL\_JE\_HEADERS LOGFILE= GL\_JE\_HEADERS.log |
| 12 | GL | GL\_JE\_LINES | **impdp** PARALLEL=4 DUMPFILE= GL\_JE\_LINES.dmp DIRECTORY= EBSDUMP TABLES = APPS.GL\_JE\_LINES LOGFILE= GL\_JE\_LINES.log |
| 13 | GL | GL\_IMPORT\_REFERENCES | **impdp** PARALLEL=4 DUMPFILE= GL\_JE\_LINES.dmp DIRECTORY= EBSDUMP TABLES = APPS.GL\_JE\_LINES LOGFILE= GL\_JE\_LINES.log |
| 14 | XLA | XLA\_AE\_HEADERS | **impdp** PARALLEL=4 DUMPFILE= XLA\_AE\_HEADERS.dmp DIRECTORY= EBSDUMP TABLES = APPS.XLA\_AE\_HEADERS LOGFILE= XLA\_AE\_HEADERS.log |
| 15 | XLA | XLA\_AE\_LINES | **impdp** PARALLEL=4 DUMPFILE= XLA\_AE\_LINES.dmp DIRECTORY= EBSDUMP TABLES = APPS.XLA\_AE\_LINES LOGFILE= XLA\_AE\_LINES.log |
| 16 | XLA | XLA\_DISTRIBUTION\_LINKS | **impdp** PARALLEL=4 DUMPFILE= XLA\_DISTRIBUTION\_LINKS.dmp DIRECTORY= EBSDUMP TABLES = APPS.XLA\_DISTRIBUTION\_LINKS LOGFILE= XLA\_DISTRIBUTION\_LINKS.log |
| 17 | XLA | XLA\_EVENTS | **impdp** PARALLEL=4 DUMPFILE= XLA\_EVENTS.dmp DIRECTORY= EBSDUMP TABLES = APPS.XLA\_EVENTS LOGFILE= XLA\_EVENTS.log |

# 5 Data Masking

### 5.1 Requirement Summary

Today in NRMA’s GFS system, whenever production database is cloned to test database, all the key information about NRMA’s customers will also get exposed in the test system. This leads to the data security issues. NRMA wishes to build an automated and configurable framework to clone data to non-production environments, in which fields are d-identified, whilst maintaining the data integrity. The primary objective of this engagement is to ensure that production data, particularly PII and other sensitive data is not present outside of the production environment.

### 5.2 Identified GFS attributes to be Masked

The attributes listed in the attached sheet will be masked using the Oracle TRANSALATE function.



##### ‘Configuration File’ to select tables to mask

NRMA wishes to build a ‘data d-identification’ job that can be executed on any target test DB by passing dynamic criteria values. In order to achieve this, solution includes ‘Configuration File/Table’ that will be passed as an input file while executing the PL/SQL scripts.

In this ‘Configuration File’, user can set the flag to Y for which the masking has to be performed.

AR.RA\_CONTACTS|LAST\_NAME|Y

AR.RA\_CONTACTS|CONTACT\_KEY|Y

AR.RA\_CONTACTS|CONTACT\_PERSONAL\_INFORMATION|Y

AR.RA\_CONTACTS|DEPARTMENT\_CODE|Y

AR.RA\_CONTACTS|DEPARTMENT|Y

##### Data Masking Logic

Below masking logic will be followed to mask the ‘string’ and ‘numeric values’ in all identified tables.

TRANSLATE (<column name>, <string to replace>, <replacement string>)

‘DATA\_MASKING\_TRANSLATE’ table will be created to store ‘String to replace’ and ‘Replacement string‘. This can be referenced from subsequent PL/SQL script used to mask the attributes.

TRANSLATE (<column name>, <string to replace>, <replacement string>)

CREATE TABLE DATA\_MASKING\_TRANSLATE (

String\_to \_replace (60),

Replacement\_string varchar(60));

INSERT INTO DATA\_MASKING\_TRANSLATE (String\_to \_replace, Replacement\_string)

VALUES (‘abcdefghijklmnopqrstuvwxyzABCDEFGHIJKLMNOPQRSTUVWXYZ0123456789’,

‘zabycxdwevfugthsirjqkplonmZABYCXDWEVFUGTHSIRJQKPLONM2014365897’);

# 

# 5.3 Design Approach

# A Table/Config file would be maintained in which all the identified attributes would be available. NRMA has to set the Flag to Y to the desired set of attributes for which the masking logic would be applied. The masked attributes would be extracted from the Load\_XX\_MASK\_REF file and will be inserted into the XX\_MASK\_REF table. The logic would pick up the attributes belongs to each table and run the respective update statement once for each table. The predefined masking logic would be stored in another table DATA\_MASKING\_TRANSLATE which would be fetched to a variable after reading it to a cursor. The date columns are masked based on a Jenkins global parameter value DID\_Masking\_Date\_value. The logic for date columns is subtraction. The values in the columns will be subtracted from the value provided in the DID\_Masking\_Date\_value.

# 6 Analyzing the Imported tables

The Oracle RDBMS allows you to collect statistics of many different kinds as an aid to improving performance. One such procedure is DBMS\_STATS. This is applied only on the tables the import activity is performed.

**Sample DBMS\_STATS Command:**

DBMS\_STATS.GATHER\_TABLE\_STATS (ownname.tabname);

# Jenkins Pipeline

Jenkins is an application for continuous integration. Jenkins Pipeline allows to execute the scripts in the defined sequence. The following is the order of scripts which needs to be configured in the Jenkins Pipeline. This is also configured in such a way that the successful execution of each step will only invoke the next script.

|  |  |  |  |
| --- | --- | --- | --- |
| **Seq no** | **Script** | **Purpose** | **Input Parameter`s** |
| 1 | Script 1 | Copy Short-listed Member IDs from GMS DB to XX\_SBL\_REF | SOURCE TNS, TARGET TNS(@lsnrctl status $ORACLE\_SID),TABLE NAME, |
| 2 | Script 2 | Export Script | ORACLE\_SID, ORACLE\_HOME,DB USERNAME/PASSWORD,  DBA\_DIRECTORIES, DUMP FILE, PAR FILE. |
| 3 | Script 3 | Import Script | ORACLE\_SID, ORACLE\_HOME,DB USERNAME/PASSWORD,  DBA\_DIRECTORIES, DUMP FILE, PAR FILE. |
| 4 | Script 4 | Masking | TABLE/CONFIG FILE WHICH HAS TABLE\_NAMES, COLUMN NAMES |

# Issues

## 8.1 Open Issues

| ID | Issue | Resolution | Responsibility | Target Date | Impact Date |
| --- | --- | --- | --- | --- | --- |
|  |  |  |  |  |  |
|  |  |  |  |  |  |
|  |  |  |  |  |  |
|  |  |  |  |  |  |
|  |  |  |  |  |  |

## 8.2 Closed Issues

| ID | Issue | Resolution | Responsibility | Target Date | Impact Date |
| --- | --- | --- | --- | --- | --- |
|  |  |  |  |  |  |
|  |  |  |  |  |  |
|  |  |  |  |  |  |
|  |  |  |  |  |  |

# Table Structures

Table Name: XX\_SBL\_REF

|  |  |
| --- | --- |
| Column Name | Data Type |
| SIEB\_CSN | Number |

Table Name: XX\_CUST\_ACCOUNT\_ID

|  |  |
| --- | --- |
| Column Name | Data Type |
| CUST\_ACCOUNT\_ID | Number |

Table Name: XX\_PARTY\_ID

|  |  |
| --- | --- |
| Column Name | Data Type |
| PARTY\_ID | Number |

Table Name: XX\_DATE\_REF

|  |  |
| --- | --- |
| Column Name | Data Type |
| FROM\_DATE | Number |
| TO\_DATE | Number |

Table Name: XX\_MASK\_REF

|  |  |
| --- | --- |
| Column Name | Data Type |
| TABLE NAME | Varchar2 |
| COLUMN NAME | Varchar2 |
| MASK\_FLAG | Varchar2 |

Table Name: XX\_DATA\_MASKING\_TRANSLATE

|  |  |
| --- | --- |
| Column Name | Data Type |
| STRING\_TO\_REPLACE | Varchar2 |
| REPLACEMENT\_STRING | Varchar2 |