Maximise Configuration Scripts for new Business Entity.

**Create lookup Populate Script -Maximise**

--==================================================================================

**Populate Maximise lookups**

INSERT INTO xxmx\_lookup\_values VALUES

('XXMX','BUSINESS\_ENTITIES',<business\_entity>,<business\_entity>,null,'Y','Y',NULL);

--==================================================================================

**Create Tables for Staging , Transformation or External File**

--==================================================================================

**EXTERNAL TABLE CREATION**

-- Only for Non EBS Clients

-- Number of Tables depends on how many data source file

-- columns depend on the Data Source file

DROP TABLE XXMX\_CORE.<External\_Table\_name>;

CREATE TABLE XXMX\_CORE.<External\_Table\_name>

(

FILE\_SET\_ID VARCHAR2(30) -- Mandatory for Maximise

,column2 VARCHAR2(200)

,column2 VARCHAR2(200)

,column2 VARCHAR2(200)

)

ORGANIZATION EXTERNAL

(

TYPE ORACLE\_LOADER

DEFAULT DIRECTORY "SOURCE\_DATAFILE"

ACCESS PARAMETERS

(

RECORDS DELIMITED BY NEWLINE SKIP 1 LOGFILE <staging\_Table.log> FIELDS TERMINATED BY "," OPTIONALLY ENCLOSED BY '"' MISSING FIELD VALUES ARE NULL

)

LOCATION (<Staging\_Table>.csv)

)

REJECT LIMIT UNLIMITED

PARALLEL 5 ;

--

----=============================================================================

--=============================================================================

**STG TABLE CREATION**

-- Create Staging table for the entity

-- Table should have all the columns needed for FBDI or HDL Generation

-- Create the columns in same sequence

DROP TABLE XXMX\_STG.<Staging\_table\_name> PURGE;

CREATE TABLE XXMX\_STG.<Staging\_table\_name>

(

FILE\_SET\_ID VARCHAR2(30) -- Mandatory for Maximise

,MIGRATION\_SET\_ID NUMBER -- Mandatory for Maximise

,MIGRATION\_SET\_NAME VARCHAR2(100) -- Mandatory for Maximise

,MIGRATION\_STATUS VARCHAR2(50) -- Mandatory for Maximise

--All FBDI Columns should be defined

);

--==============================================================================

--==============================================================================

--

--

**XFM TABLE CREATION**

--

DROP TABLE XXMX\_XFM.<Transformation\_table\_name> PURGE;

-- Create Transformation table for the entity

-- Table should have all the columns needed for FBDI or HDL Generation

-- it should a replica of staging tables

-- Create the columns in same sequence

CREATE TABLE XXMX\_XFM.<Transformation\_table\_name>

(

FILE\_SET\_ID VARCHAR2(30) -- Mandatory for Maximise

,MIGRATION\_SET\_ID NUMBER -- Mandatory for Maximise

,MIGRATION\_SET\_NAME VARCHAR2(100) -- Mandatory for Maximise

,MIGRATION\_STATUS VARCHAR2(50) -- Mandatory for Maximise

--All FBDI Columns should be defined

);

--

--==================================================================================

**Create Grants to Tables**

--==================================================================================

**--GRANTS And SYNONYM**

Grant all on XXMX\_STG.<Staging\_table\_name> to XXMX\_CORE;

Grant all on XXMX\_XFM.<Transformation\_table\_name> to XXMX\_CORE;

Create Synonym XXMX\_CORE.<Staging\_table\_name> for XXMX\_STG.<Staging\_table\_name>;

Create Synonym XXMX\_CORE.<Transformation\_table\_name> for XXMX\_XFM.<Transformation\_table\_name>;

--

--==================================================================================

**Create Archive scripts for Stage and Transformation**

--==================================================================================

--

**Create Archive Script**

-- No parameters needed to execute the script



--==================================================================================

**Create Metadata Entries**

--==================================================================================

--

**Metadata Configuration**

SELECT BUSINESS\_ENTITY\_SEQ FROM XXMX\_MIGRATION\_METADATA ORDER BY BUSINESS\_ENTITY\_SEQ DESC; -- to get the next number for Metadata\_tables.

SELECT \* FROM XXMX\_MIGRATION\_METADATA ORDER BY 1 DESC;

--Populate Metadata\_tables.

INSERT INTO xxmx\_migration\_metadata

VALUES

(XXMX\_MIGRATION\_METADATA\_IDS\_S.nextval,<application\_suite>,<application>,<Maximum business\_entity\_seq+1>

,<business\_entity>,'1',<Sub\_entity>,<Extract Package Name> ,NULL,<Extract Procedure Name> ,<staging\_Table\_name>,NULL,NULL,<Transformation\_table\_name>, null,null,null,*<FBDIFilename- csvfilename>,*'csv',1,'Y',NULL,NULL);

--

--

--

--==================================================================================

**Create Data Dictionary Entries**

--==================================================================================

-- Populate Data Dictionarytables - Mandatory for new business\_entity

-- Execute it in same order first xfm\_populate and then stg\_populate scripts.

DECLARE

pv\_o\_ReturnStatus VARCHAR2(3000);

BEGIN

xxmx\_dynamic\_sql\_pkg.xfm\_populate(

pt\_i\_ApplicationSuite=><application\_suite>,

pt\_i\_Application=> <application>,

pt\_i\_BusinessEntity=> <business\_entity>,

pt\_i\_SubEntity=> <sub\_entity>,

pt\_fusion\_template\_name=> <FBDIFilename- csvfilename>.csv, -- With extension

pt\_fusion\_template\_sheet\_name => FBDIFilename- csvfilename, -- without extension

pt\_fusion\_template\_sheet\_order=>1,

pv\_o\_ReturnStatus=> pv\_o\_ReturnStatus);

END ;

/

--

--

--

--

DECLARE

pv\_o\_ReturnStatus VARCHAR2(3000);

BEGIN

xxmx\_dynamic\_sql\_pkg.stg\_populate(

pt\_i\_ApplicationSuite=><application\_suite>,

pt\_i\_Application=> <application>,

pt\_i\_BusinessEntity=> <business\_entity>,

pt\_i\_SubEntity=> <sub\_entity>,

pt\_import\_data\_file\_name => <Staging\_table\_name>.csv,

pt\_control\_file\_name => <Staging\_table\_name>.ctl,

pt\_control\_file\_delimiter => ',',

pv\_o\_ReturnStatus=> pv\_o\_ReturnStatus);

END ;

/

Update XXMX\_XFM\_TABLE\_COLUMNS

Set Include\_in\_outbound\_file = ‘Y’

Where xfm\_Table\_id = < Table id of the new business entity>

AND Column\_name NOT IN ( ‘FILE\_SET\_ID’,’MIGRATION\_SET\_ID’,’MIGRATION\_SET\_NAME’,’MIGRATION\_STATUS’);

Note: Get table id from XXMX\_XFM\_TABLES

--==================================================================================

**Create FBDI File Sequence Entries**

--==================================================================================

--

Populate the table XXMX\_DM\_SUBENTITY\_FILE\_MAP table with the columns in xfm tables , in the same sequence as FBDI Control File.

--

--==================================================================================

**Integration and VBCS**

--==================================================================================

**Integration Changes**

**Add new entry in Load and Import integrations by adding a new If loop**

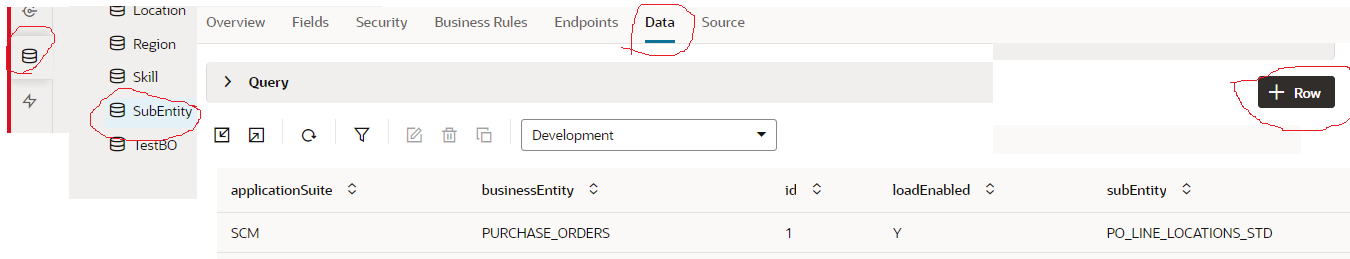
INTDM999006 Load FBDI to Interface Table

INTDM999007 DM Run Fusion Import Process

**Add new entry in VBCS BusinessEntity and SubEntity**

Graphical user interface, text, application, chat or text message

Description automatically generated

****

--==================================================================================

--==================================================================================

**Add Entry in xxmx\_dm\_ess\_job\_definitions for Fusion Import Parameters**

--==================================================================================



