



Maximise Data Migration

Technical Guide

@Copyright 2022 Version 1 – All Rights Reserved

Contents

1. Overview 5

1.1. About Maximise Data Migration (Maximise DM) 5

1.2. Target Audience 5

1.3. Related Documents 5

1.4. Terminology 5

1.5. What is not covered under Maximise DM 6

2. Maximise DM Overview 7

2.1. Architecture 7

2.2. Extract 7

2.3. Transform 8

2.4. Load 8

3. Maximise Technical SPECIFICATION 9

3.1. Business Entities/ Sub Entities 9

3.1.1. Finance 9

3.1.2. HCM 10

3.2. Maximise Core 13

3.2.1. Core Tables 13

3.2.2. Core Utilities 14

3.3. Maximise Extract 16

3.3.1. EBS - Extract 16

3.3.2. Non-EBS Extract 19

3.4. Maximise Transform 19

3.4.1. Simple Transformation 19

3.4.2. Complex Transformation 19

3.4.3. How to Link Simple Transforms 19

3.5. Maximise Load 21

3.5.1. Maximise Configurations 21

4. Maximise OIC and VBCS 22

4.1. OIC Components 22

4.2. VBCS Components 23

5. Overview 26

5.1. Batching 26

5.2. Non-Batching 26

6. Design Flow 27

7. Configuration 28

7.1. Batching 28

7.2. Non-Batching 28

8. Technical Specification 29

8.1. Batching 29

8.1.1. Extract 29

8.1.2. Transform 30

8.1.3. Load 30

8.2. Non-Batching 32

8.2.1 Extract 32

8.2.2. Transform 32

8.2.3. Load 32

9. Batching and Non-Batching Snippets 34

9.1 Batching 34

9.2. Non-Batching 34

**Version Control**

|  |  |  |  |
| --- | --- | --- | --- |
| Date | Version | Changed By | Reason for Change |
| 16-MAY-2022 | 0.1 | Pallavi Kanajar | No previous version |
|  |  |  |  |

**Circulation List**

|  |  |
| --- | --- |
| Name | Organisation/Title |
| Ken MacMahon | Version 1 |
| Bridget Morrissey | Version 1 |
| Praveen Nair | Version 1 |

# Overview

## About Maximise Data Migration (Maximise DM)

Maximise Data Migration is Version1’s custom-built toolkit to accelerate Oracle SaaS implementation. It provides a framework for extracting, transforming, and loading data from Oracle EBS and non-EBS systems into Oracle Fusion Cloud. The toolkit provides reconciliation dashboard and covers most common data migration scenarios. Maximise Data Migration accelerator offers coverage across various applications within Oracle ERP Cloud and HCM Cloud.

Use Maximise Data Migration accelerator to :

* quickly setup database links to extract data from a compatible source database or use pre-defined templates for populating source data from a non-EBS database
* Configure mappings in an easy-to-use template to perform simple and complex transformations
* Capture additional customer business or validation requirements, by extending the functionality provided by the standard accelerator
* Generate load files and perform import to Fusion Cloud all through a very simple easy to use user interface

## Target Audience

This guide is intended for technical developers interested in understanding the technical aspects of Maximise Data Migration accelerator.

## Related Documents

Additional reference documents for Maximise Data Migration accelerator are provided below:

* [Maximise Data Migration Technical Checklist](https://version1.sharepoint.com/sites/ERPSolutionArchitecture/SiteAssets/SitePages/Maximise--Data-Migration/Maximise_Technical_Requirements_Checklist_v1_0.pdf?web=1)
* [Maximise Data Migration Installation Guide](https://git.version1.com/projects/VESA/repos/maximise-dm-production-release/raw/Maximise_Installation_Guide.docx?at=refs/heads/DEV)
* [Maximise Data Migration User Guide](https://git.version1.com/projects/VESA/repos/maximise-dm-production-release/raw/Maximise_Installation_Guide.docx?at=refs/heads/DEV)

## Terminology

Some of the terminology used in this document and across Maximise Data Migration is shown below:

|  |  |
| --- | --- |
| Key | Description |
| Maximise DM | Maximise Data Migration accelerator |
| Business Entity | Refers to a Business Application Area e.g., Finance, HCM |
| Sub Entity | Represents each level of data with Business Entity |
| Migration Set | Maximise Term used to describe the group of all sub entity data being processed in a single execution for a specific business entity. e.g., AR, AP, GL |
| File Set | Maximise DM term used to group the Source Data File from Non – EBS customers |
| Metadata | This is the driver for all the ETL processes |
| Simple Transformation | 1:1 Mapping Rule. Simple maps one source value to one target value |
| Complex Transformation | Transformation logic that includes a combination of source columns to derive a new target value |
| Phase | Data Migration Phases – Extract, Transform and Load |
| DBAAS | Database as a Service |
| STG | Staging table or act of staging data in a database table |
| XFM | Transformation activity |
| EBS | Client Source System is Oracle R12 or 11i Environment |
| NON- EBS | Client Source System is not Oracle ERP. |

## What is not covered under Maximise DM

* Provisioning database for Maximise DM installation. This will be the responsibility of Customer IT or Version 1 implementation teams
* All configuration items e.g., Organisation, location, job, grade, talent, position etc are the responsibility of the implementation team
* Cloud configurations cannot be implemented using Maximise DM
* Maximise only provides tools to perform Transformation and Load. Mapping and Fusion Load is out of scope for Maximise should be performed by Delivery Team.
* Customer specific customisations and mapping rules will not be covered under Maximise DM
* Data cleansing is outside the scope of Maximise DM and should be performed before the Extract task

# Maximise DM Overview

## Architecture

This section outlines the architecture of the Maximise Data Migration accelerator.

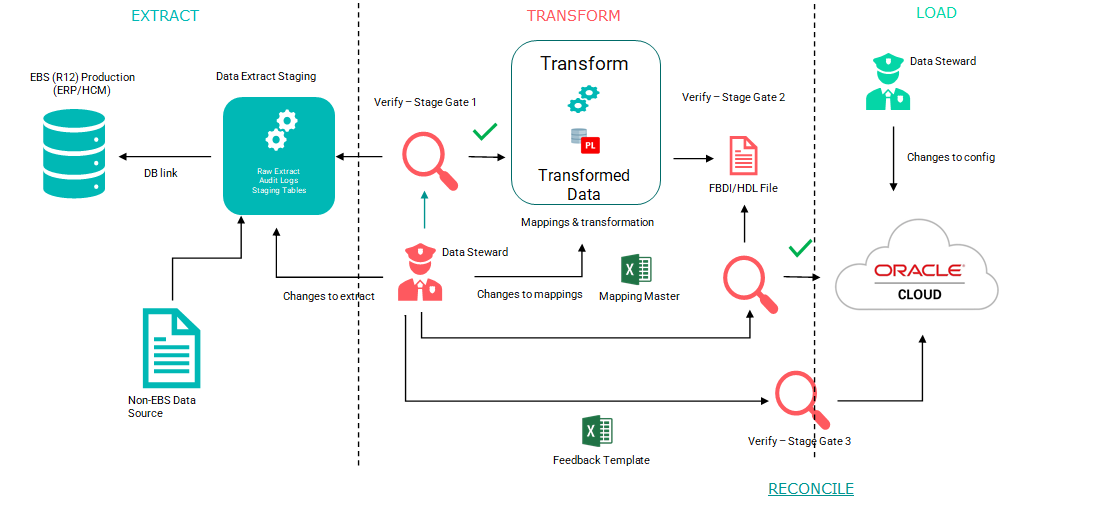


Figure 1 Maximise DM Architecture

## Extract

Diagram

Description automatically generated

* A standard set of technical extract routines are deployed in a separate dedicated Oracle database (typically as DBaaS in customer cloud tenancy). These extract routines manage the extraction of data from the “Production” legacy data tables, which are then loaded to the Maximise staging (STG) tables

**NOTE:** If the source is a non-EBS Source system. A set of standardised flat files are loaded using standard Maximise extract routines through external tables.

* <<Customer>> business stewards validate the technical extracts
* The <<Customer>> business stewards provide any requisite “standard” mappings e.g., Code Combination mappings, in a pre-defined (Excel) format (“Mapping\_Master Spreadsheet”). This is then loaded into the Maximise engine to drive the automatic <customer> specific mapping process, to the Transform (XFM) tables

## Transform

Chart, diagram

Description automatically generated

Data is moved, between STG and XFM, by transform routines. This also perform transformation in simple transforms and Custom extensions (if any are defined in the Maximise Data Migration accelerator)

**Note:***the pre-defined mappings are covered outside Maximise DM and is captured by project teams during workshops with business teams*

## Load

The loading of the data into the Fusion interface tables can be manual or automatic with Maximise DM. The steps in both the cases, are as follows: -

* The HDL/FBDI output is generated in CSV (“.dat”) format, by Maximise DM, which is then loaded to the Fusion (Interface Tables)
* Verify that the load interface file, for the Import process, completes successfully
* Verify that the import process completes successfully

**NOTE:** If OIC is not the Load mechanism used e.g., for SCM, these steps are performed manually after generating the “.xlsm” file.

# Maximise Technical SPECIFICATION

This section outlines the Business Entities and Sub Entities included in Maximise Data Migration (Maximise DM) accelerator.

## Business Entities/ Sub Entities

Maximise DM covers the following modules/business areas - HCM, Finance and Supply Chain and the key business and sub-entities covered under Maximise DM are listed below.

**Note:** *Not all business entities are covered within Maximise DM. New functions and features are added to the accelerator regularly. For the latest coverage of business entities, refer to the Maximise portal (*[*Maximise: Data Migration*](https://version1.sharepoint.com/sites/ERPSolutionArchitecture/SitePages/Maximise--Data-Migration.aspx)*)*

## Finance

The table below shows a list of all business and sub-entities covered by Maximise DM for Finance.

|  |  |
| --- | --- |
| Business Entity | Sub Entity |
| Suppliers | Suppliers |
| Supplier Addresses |
| Supplier Sites |
| Supplier Third Party Relationships |
| Supplier Site Assignments |
| Supplier Contacts |
| Supplier Contact Addresses |
| Supplier Payees |
| Supplier Bank Accounts |
| Supplier Payment Instrument Usages |
| Invoices | Invoice Headers |
| Invoice Lines |
| Transactions | Lines |
| Distributions |
| Sales Credits |
| Customers | Parties |
| Party Sites |
| Party Site Uses |
| Customer Accounts |
| Customer Account Sites |
| Customer Account Site Uses |
| Customer Profiles |
| Locations |
| Relationships |
| Customer Account Contacts |
| Organization Contacts |
| Contact Roles |
| Contact Points |
| Person Language |
| Party Classifications |
| Roles and Responsibilities |
| Customer Account Relationships |
| Receipt Methods |
| Customer Bank Accounts |
| Cash Receipt | Cash Receipt |
| Balances | Detail Balances |
| Summary Balances |
| Open Balances |
| Daily Rates | Daily Rates |
| Purchase Orders | Headers |
| Lines |
| Line Locations |
| Distributions |
| Blanket Purchase Agreements |
| Contract Purchase Agreements |
| Fixed Assets | Mass Additions |
| Mass Addition Distributions |
| Mass Rates |
| Projects | Projects |
| Tasks |
| Transaction Controls |
| Team Members |
| Project Classifications |
| Costs | Miscellaneous Cost |
| Labor Cost |
| Non Labor Cost |
| Supplier Cost |

## HCM

The table below shows a list of all business and sub-entities covered by Maximise DM for HCM.

|  |  |
| --- | --- |
| Business Entity | Sub Entity |
| Person | Bank |
| Bank Branches |
| Person |
| Person National Identifier |
| Contacts |
| Address |
| Address Usage |
| Names |
| Legislative Data |
| Ethnicities |
| Phone |
| Citizenship |
| Email |
| Passport |
| Contact Relationship |
| Contact Address |
| Contact Phone |
| Religion |
| Employee Photos |
| Visa |
| Disability |
| Worker | WorkRelationship |
| Assignments |
| Seniority Date |
| Assignment Supervisor |
| Assignment WorkMeasure |
| Assignment GradeSteps |
| Talent | Talent Profile |
| Performance Rating |
| Job Profile |
| Position Profile |
| Payroll | External Bank Accounts |
| Assignment Payroll |
| Personal Payment Method |
| Assignment Salary |
| Element Entries |
| Cost Allocations |
| Balances |
| Statutory Deductions |
| Pension Auto Enrolment |
| Balances |
| Statutory Deductions |
| Pension Auto Enrolment |
| Benefits and Pension |
| Student Load |
| New Starter Declaration |

|  |  |
| --- | --- |
| Business Entity | Sub Entity |
| Benefits | Participant Enrollment |
| Compensation Object |
| Dependent Enrolment |
| Designate Dependent |
| Beneficiary Enrolment |
| Designate Beneficiary |
| Benefit Group |
| Benefit Balance |
| Person Habits |
| Benefit Organization |

|  |  |
| --- | --- |
| Business Entity | Sub Entity |
| Learning | Legacy Learning Item |
| Legacy Learning Item Translation |
| Non-Catalog Learning Item |
| Non-Catalog Learning Item Translation |
| Classroom Resource |
| Classroom Resource Translation |
| Instructor Resource |
| Course |
| Offering |
| Instructor Led Activity |
| Adhoc Resource |
| Classroom Reservation |
| Instructor Reservation |
| Self-Paced Activity |
| Offering Default Access |
| Course Default Access |
| Course Translation |
| Offering Translation |
| Instructor Led Activity Translation |
| Self-Paced Led Activity Translation |
| Course Offering Pricing Default |
| Course Offering Pricing Component |
| Offering Custom Pricing |
| Offering Custom Pricing Component |
| Specialization |
| Specialization Section |
| Specialization Section Activity |
| Specialization Default Access |
| Specialization Translation |
| Specialization Section Translation |
| Global Access Group Relation |
| Community Relation |
| Learning Record |
| Learning Record Activity Attempt |

## Maximise Core

Maximise data migration accelerator has 4 phases - Extract, Transform, Load and Reconciliation. All Maximise Data Migration phases use the Core utilities, Core tables to function.

## Core Tables

The table below provides a list of core tables and descriptions for each. These tables are created as part of the Maximise Data Migration accelerator. These tables are populated as part of Maximise Installation.

|  |  |
| --- | --- |
| Table Name | Description |
| XXMX\_CORE\_PARAMETERS | To Determine if the Client is EBS (DB\_LINK) or Non- EBS Clients (DATA\_FILE) |
| XXMX\_MIGRATION\_DETAILS | Batch Identifier generated by Maximise for each data set extracted |
| XXMX\_MIGRATION\_HEADERS | Batch Identifier generated by Maximise for each data set extracted |
| XXMX\_MIGRATION\_METADATA | Table holds details of Business Entity and Sub Entity. Extract packages, Staging, Transformation and External Tables. Client can enable and disable any business entity or sub entity for Extract, Transform or Load. |
| XXMX\_MIGRATION\_PARAMETERS | Table holds scope parameters for Extract. |
| XXMX\_STG\_TABLES | These are Data Dictionary Tables for Stage tables to hold details for file generation – Filename and File Delimiter etc.. |
| XXMX\_STG\_TABLE\_COLUMNS | These are Data Dictionary Tables for Stage tables to hold details of staging columns. |
| XXMX\_XFM\_TABLES | These are Data Dictionary Tables for Transform tables to hold details for file generation – Filename and File Delimiter etc.. |
| XXMX\_XFM\_TABLE\_COLUMNS | These are Data Dictionary Tables for Transform tables to hold details of staging columns. |
| XXMX\_SIMPLE\_TRANSFORMS | This Table holds the Rules for Simple Transformation – 1:1 Mapping |
| XXMX\_MAPPING\_MASTER | This table holds the Rules for Complex Transformation |
| XXMX\_MODULE\_MESSAGES | This table has log messages for each phase – Extract, Transform and Load |
| XXMX\_FILE\_LOCATIONS | This table is required to hold the File Path information to generate the csv file in Oracle Path. |
| XXMX\_HDL\_FILE\_TEMP | HDL file for Fusion load is loaded to this table before generating the csv file. |
| XXMX\_CSV\_FILE\_TEMP | CSV file for Fusion load is loaded to this table before generating the csv file. |
| XXMX\_CUSTOM\_EXTENSIONS | Customization for Business Entity can be placed in this table to plug in to Maximise. |
| XXMX\_CUSTOM\_SUB\_EXTENSIONS | Customization for Sub Entity can be placed in this table to plug in to Maximise. |
| XXMX\_UTILITIES\_PKG | Maximise Core utility package |
| XXMX\_DYNAMIC\_SQL\_PKG | Maximise Common Package for Transformation, Extract and Load. |
| XXMX\_FIN\_STG\_EXTRACT\_PKG | Maximise Generic Package for Finance Extract and Transformation. |
| XXMX\_HCM\_STG\_EXTRACT\_PKG | Maximise Generic Package for HCM Extract and Transformation. |
| XXMX\_FUSION\_LOAD\_GEN\_PKG | Maximise Generic Package for Load file generation. |

## Core Utilities

**Note:** *Core Utilities should not be changed by the delivery/implementation team.*

There are two main core utilities packages, which drives all Maximise DM Phases.

**XXMX\_UTILITIES\_PKG** has the below procedures

|  |  |
| --- | --- |
| Procedure Name | Uses |
| log\_module\_message | Log debug messages |
| valid\_lookup\_code | Validation for maximise lookup code |
| get\_business\_entity\_seq | Get business entity seq from xxmx\_migration\_metadata |
| get\_sub\_entity\_seq | Get Sub Entity seq from xxmx\_migration\_metadata |
| clear\_messages | Clear module messages for procedure |
| get\_client\_config\_value | Get Client Config Values if any |
| verify\_parameter\_exists | Verify if migration Parameter exists |
| get\_single\_parameter\_value | Get single migration parameters |
| get\_parameter\_value\_list | Get list of migration parameters |
| simple\_transform\_exists | Check if simple Transforms exist |
| init\_migration\_set | Initialise migration set id |
| get\_migration\_set\_name | Get migration set name |
| get\_migration\_set\_id | Get migration set id |
| init\_migration\_details | Initialize migration set Details |
| write\_csv | Write load csv file to table |
| Open\_csv | Open csv file in unix server |
| p\_extract\_data | Used by OIC to call extract |
| get\_csvdata\_count | Used by OIC to get csv data count |
| get\_stgxfm\_data\_count | Used by OIC to get stage and xfm table count |
| insert\_into\_arch\_table | Used by OIC to archive the data |
| truncate\_stg\_xfm\_table | Used by OIC to truncate stg and xfm tables |
| delete\_mapping\_master | Used by OIC to truncate mapping master |
| call\_fusion\_load\_gen | Used by OIC to call Load Generation package |
| get\_FBDI\_filenames | Used by OIC to get FBDI File names. |

**XXMX\_DYNAMIC\_SQL\_PKG** has the below procedures.

|  |  |
| --- | --- |
| Procedure Name | Uses |
| prevalidate\_stg\_data | Validation for non EBS clients |
| transfer\_stg\_data | Transfer Data from staging tables to xfm tables |
| transform\_data | Transform Data in staging table based on mapping |
| execute\_extension | Call Custom Extensions |
| call\_sub\_extension | Call Custom Sub Extensions if any |
| log\_purge\_message | Log Purge messages tables |
| purge\_migration\_data | Purge migration Data |
| xfm\_populate | Populate xfm Data Dictionary Tables |
| stg\_populate | Populate staging Data Dictionary Tables |
| stg\_update\_columns | To populate mandatory and other fields in Data Dictionary tables |
| xfm\_update\_columns | To populate mandatory and other fields in Data Dictionary tables |

**XXMX\_FIN\_STG\_EXTRACT\_PKG** has below procedures

|  |  |
| --- | --- |
| Procedure Name | Uses |
| stg\_main | Calls Individual extract procedures setup in xxmx\_migration\_metadata for Finance Modules |
| xfm\_main | Calls generic transform procedures for Finance Modules |

**XXMX\_HCM\_STG\_EXTRACT\_PKG** has below procedures

|  |  |
| --- | --- |
| Procedure Name | Uses |
| stg\_main | Calls Individual extract procedures setup in xxmx\_migration\_metadata for HCM Modules |
| xfm\_main | Calls generic transform procedures for HCM Modules |

**XXMX\_FUSION\_LOAD\_GEN\_PKG** has below procedures

|  |  |
| --- | --- |
| Procedure Name | Uses |
| Generate\_csv\_data | Generates Fusion FBDI File |
| Generate\_hdl\_data | Generate Fusion HDL File |

## Maximise Extract

## EBS - Extract

The core utility packages **XXMX\_HCM\_STG\_EXTRACT\_PKG** and **XXMX\_FIN\_STG\_EXTRACT\_PKG** handles both Non-EBS and EBS customer database sources.

The code units below are to extract data from an EBS data source. Details of the packages and the staging tables can be found by querying the **XXMX\_MIGRATION\_METADATA**. Parameters used by each extract routines are defined in the **XXMX\_MIGRATION\_PARAMETERS** table.

|  |  |  |  |
| --- | --- | --- | --- |
| Business Entity | Package Name | Staging tables | Dependent Code Units /Parameters |
| Suppliers | xxmx\_ap\_suppliers\_pkg | XXMX\_AP\_SUPPLIERS\_STG  XXMX\_AP\_SUPP\_ADDRS\_STG  XXMX\_AP\_SUPPLIER\_SITES\_STG  XXMX\_AP\_SUPP\_3RD\_PTY\_RELS\_STG  XXMX\_AP\_SUPP\_CONT\_ADDRS\_STG  XXMX\_AP\_SUPP\_CONTACTS\_STG  XXMX\_AP\_SUPP\_SITE\_ASSIGNS\_STG  XXMX\_AP\_SUPP\_PAYEES\_STG  XXMX\_AP\_SUPP\_PMT\_INSTRS\_STG  XXMX\_AP\_SUPP\_BANK\_ACCTS\_STG | xxmx\_ap\_supplier\_scope\_v\_dbi  xxmx\_iby\_payee\_bank\_accts\_v  VENDOR\_TYPE  MONTHS\_TO\_MIGRATE  ORDER\_OF\_PREFERENCE\_LIMIT  ORGANIZATION\_NAME |
| Invoices | xxmx\_ap\_inv\_pkg | XXMX\_AP\_INVOICES\_STG  XXMX\_AP\_INVOICE\_LINES\_STG | xxmx\_ap\_inv\_scope\_v |
| Customers | xxmx\_ar\_customers\_pkg | XXMX\_HZ\_PARTIES\_STG  XXMX\_HZ\_PARTY\_SITES\_STG  XXMX\_HZ\_PARTY\_SITE\_USES\_STG  XXMX\_HZ\_CUST\_ACCOUNTS\_STG  XXMX\_HZ\_CUST\_ACCT\_SITES\_STG  XXMX\_HZ\_CUST\_SITE\_USES\_STG  XXMX\_HZ\_CUST\_PROFILES\_STG  XXMX\_HZ\_LOCATIONS\_STG  XXMX\_HZ\_RELATIONSHIPS\_STG  XXMX\_HZ\_CUST\_ACCT\_CONTACTS\_STG  XXMX\_HZ\_ORG\_CONTACTS\_STG  XXMX\_HZ\_ORG\_CONTACT\_ROLES\_STG  XXMX\_HZ\_CONTACT\_POINTS\_STG  XXMX\_HZ\_PERSON\_LANGUAGE\_STG  XXMX\_HZ\_PARTY\_CLASSIFS\_STG  XXMX\_HZ\_ROLE\_RESPS\_STG  XXMX\_HZ\_CUST\_ACCT\_RELATE\_STG  XXMX\_RA\_CUST\_RCPT\_METHODS\_STG  XXMX\_AR\_CUST\_BANKS\_STG | xxmx\_ar\_customer\_scope\_v  INCLUDE\_INACTIVE\_ACCOUNTS  INCLUDE\_INACTIVE\_SITES  MONTHS\_TO\_MIGRATE  NO\_ACTIVITY\_CUSTOMER\_MONTHS  ORGANIZATION\_NAME |
| AR Transactions | xxmx\_ar\_trx\_pkg | XXMX\_AR\_TRX\_LINES\_STG  XXMX\_AR\_TRX\_DISTS\_STG  XXMX\_AR\_TRX\_SALESCREDITS\_STG | xxmx\_ar\_trx\_scope\_v\_dbi  INCLUDE\_CREDIT\_MEMOS  INCLUDE\_DEBIT\_MEMOS  AR\_TRX\_SOURCE\_ID |
| AR Cash Receipts | xxmx\_ar\_cash\_receipts\_pkg | XXMX\_AR\_CASH\_RECEIPTS\_STG | xxmx\_ar\_cash\_receipts\_scope\_v\_dbi |
| Balances | xxmx\_gl\_balances\_pkg | XXMX\_GL\_OPENING\_BALANCES\_STG  XXMX\_GL\_DETAIL\_BALANCES\_STG  XXMX\_GL\_SUMMARY\_BALANCES\_STG | JE\_CATEGORY  JE\_SOURCE  STATUS  EXTRACT\_YEAR  PERIOD\_NAME  LEDGER\_NAME  PERIOD\_NAME |
| Fixed Assets | xxmx\_fa\_mass\_additions\_pkg | XXMX\_FA\_MASS\_ADDITIONS\_STG  XXMX\_FA\_MASS\_ADDITION\_DIST\_STG  XXMX\_FA\_MASS\_RATES\_STG | CUT\_OFF\_DATE  BOOK\_TYPE\_CODE |
| Projects | xxmx\_ppm\_projects\_pkg | XXMX\_PPM\_PROJECTS\_STG  XXMX\_PPM\_PRJ\_TASKS\_STG  XXMX\_PPM\_PRJ\_CLASS\_STG  XXMX\_PPM\_PRJ\_TRX\_CONTROL\_STG  XXMX\_PPM\_PRJ\_TEAM\_MEM\_STG | EXTRACT\_START\_DATE  EXTRACT\_END\_DATE  GL\_PERIOD\_NAME  PROJECT\_STATUS\_CODE  PROJECT\_TYPE  EXP\_SYSTEM\_LINK\_TYPE  COST\_EXT\_TYPE |
| Project Costs | xxmx\_ppm\_prj\_billevent\_pkg | XXMX\_PPM\_PRJ\_BILLEVENT\_STG  XXMX\_PPM\_PRJ\_MISCCOST\_STG  XXMX\_PPM\_PRJ\_LBRCOST\_STG  XXMX\_PPM\_PRJ\_SUPCOST\_STG  XXMX\_PPM\_PRJ\_NONLABCOST\_STG |  |
| Person | xxmx\_kit\_person\_stg | XXMX\_PER\_PERSONS\_STG  XXMX\_PER\_NID\_F\_STG  XXMX\_PER\_CONTACTS\_STG  XXMX\_PER\_ADDRESS\_F\_STG  XXMX\_PER\_ADDR\_USG\_F\_STG  XXMX\_PER\_NAMES\_F\_STG  XXMX\_PER\_RELIGION\_STG  XXMX\_PER\_DISABILITY\_STG  XXMX\_PER\_IMAGES\_STG  XXMX\_PER\_VISA\_F\_STG  XXMX\_PER\_ABSENCE\_STG  XXMX\_PER\_MAT\_ABSENCE\_STG  XXMX\_PER\_LEG\_F\_STG  XXMX\_PER\_ETHNICITIES\_STG  XXMX\_PER\_PHONES\_STG  XXMX\_CITIZENSHIPS\_STG  XXMX\_PER\_EMAIL\_F\_STG  XXMX\_PER\_PEOPLE\_F\_STG  XXMX\_PER\_PASSPORT\_STG  XXMX\_PER\_CONTACT\_REL\_STG  XXMX\_PER\_CONTACT\_ADDR\_STG  XXMX\_PER\_CONTACT\_PHONE\_STG | PERSON\_TYPE  MIGRATE\_DATE\_FROM  PEOPLE\_GROUP  MIGRATE\_DATE\_TO  ASSIGNMENT\_DFF  BUSINESS\_GROUP\_NAME  PAYROLL\_NAME |
| Worker | xxmx\_kit\_worker\_stg | XXMX\_PER\_SENIORITYDT\_STG  XXMX\_PER\_POS\_WR\_STG  XXMX\_PER\_ASSIGNMENTS\_M\_STG  XXMX\_PER\_ASG\_SUP\_F\_STG  XXMX\_ASG\_PAYROLL\_STG  XXMX\_PER\_PAY\_METHOD\_STG  XXMX\_PER\_ASG\_SALARY\_STG  XXMX\_ASG\_WORKMSURE\_STG  XXMX\_ASG\_GRADESTEP\_STG  XXMX\_EXT\_BANK\_ACC\_STG |  |
| Talent | xxmx\_kit\_talent\_stg | TBD |  |
| Purchase Orders | xxmx\_po\_headers\_pkg | XXMX\_SCM\_PO\_HEADERS\_STD\_STG  XXMX\_SCM\_PO\_LINES\_STD\_STG  XXMX\_SCM\_PO\_LINE\_LOCATIONS\_STD\_STG  XXMX\_SCM\_PO\_DISTRIBUTIONS\_STD\_STG  XXMX\_SCM\_PO\_HEADERS\_BPA\_STG  XXMX\_SCM\_PO\_LINES\_BPA\_STG  XXMX\_SCM\_PO\_LINE\_LOCATIONS\_BPA\_STG  XXMX\_SCM\_PO\_HEADERS\_CPA\_STG | xxmx\_scm\_po\_open\_qty\_mv  xxmx\_purchase\_order\_scope\_v  MONTHS\_TO\_MIGRATE  PO\_TYPE  ORGANIZATION\_NAME  INCLUDE\_FULLY\_RECEIPTED\_POS |
| Job Referral | xxmx\_irec\_job\_referral\_stg | XXMX\_HCM\_IREC\_REFERRAL\_STG |  |
| Job Requisition | xxmx\_irec\_job\_requisition\_stg | XXMX\_HCM\_IREC\_JOB\_REQ\_STG  XXMX\_HCM\_IREC\_JR\_HIRE\_TEAM\_STG  XXMX\_HCM\_IREC\_JR\_POST\_DET\_STG |  |
| Candidate | xxmx\_irecruit\_cand\_stg | XXMX\_HCM\_IREC\_CAN\_STG  XXMX\_HCM\_IREC\_CAN\_ADDRESS\_STG  XXMX\_HCM\_IREC\_CAN\_EMAIL\_STG  XXMX\_HCM\_IREC\_CAN\_NAME\_STG  XXMX\_HCM\_IREC\_CAN\_PHONE\_STG  XXMX\_HCM\_IREC\_CAN\_ATTMT\_STG | MIGRATE\_DATE\_TO  MIGRATE\_DATE\_FROM  PERSON\_TYPE |
| Candidate Pool | xxmx\_irecruit\_cand\_pool\_stg | XXMX\_HCM\_IREC\_CAN\_POOL\_STG  XXMX\_HCM\_IREC\_CP\_TAL\_COMM\_DET\_STG  XXMX\_HCM\_IREC\_CAN\_POOL\_MEMBR\_STG  XXMX\_HCM\_IREC\_CAN\_POOL\_INTERACT\_STG  XXMX\_HCM\_IREC\_CAN\_POOL\_OWNER\_STG |  |
| Prospect | xxmx\_irecruit\_prospect\_stg | XXMX\_HCM\_IREC\_PROSPECT\_STG |  |

For Finance entities, the extract package - **XXMX\_FIN\_STG\_EXTRACT\_PKG**, will call specific procedures for each sub-entity with only 2 parameters, as shown below:

**pt\_i\_MigrationSetID IN xxmx\_migration\_headers. migration\_set\_id%TYPE**

**pt\_i\_SubEntity IN xxmx\_migration\_metadata.sub\_entity%TYPE**

The main procedure ***stg\_main*** from this package calls relevant sub entity procedures, which are configured in **XXMX\_MIGRATION\_METADATA**.

On the other hand, for HCM entities, the extract package **XXMX\_HCM\_STG\_EXTRACT\_PKG**, will call specific procedures for each sub-entity with 4 parameters, as below:

**p\_bg\_name IN VARCHAR2**

**p\_bg\_id IN number**

**pt\_i\_MigrationSetID IN xxmx\_migration\_headers. migration\_set\_id%TYPE**

**pt\_i\_MigrationSetName IN xxmx\_migration\_headers. migration\_set\_name%TYPE**

The main procedure ***stg\_main*** from this package calls relevant sub entity procedures, which are configured in the **XXMX\_MIGRATION\_METADATA** table.

***Migration Set ID*** and ***Migration Set Name*** are generated by Maximise DM for each business entity extract. The ***File Set ID*** is not mandatory column for EBS Client.

If the extract procedure fails at any point, the messages will be logged in the **XXMX\_MODULE\_MESSAGES** table.

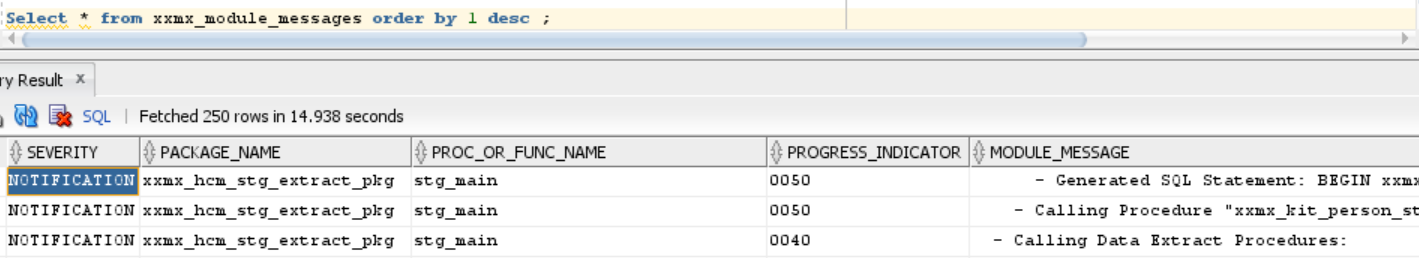


Figure 2 XXMX\_MODULE\_MESSAGES table

## Non-EBS Extract

For Non EBS customers, each staging table will correspond to an external table. These external tables will be populated by client data files placed in the path specified in DB Directory **SOURCE\_DATAFILE** directory. By default, Maximise team will configure the path as ‘/tmp’ for the customer. The implementation team can change it as per requirement.

The external tables are replicas of Staging tables. Names of external tables can be found in the **XXMX\_MIGRATION\_METADATA** table.

File Set ID is a mandatory column for Non-EBS customers and included in the external table definition. This marks the data file set id for each extract. Maximise, internally generates the Migration Set ID and Migration Set Name for each corresponding File Set ID.

**Note:** *Maximise\_User\_Guide.docx details about how to execute these packages from SqlDeveloper.*

*No extract code is written for Non-EBS customers. Assumption is, data for non EBS customers will be loaded using client data files (in CSV format).*

## Maximise Transform

Maximise by default performs only Simple Transformations.

Complex transformations should be managed by the delivery/implementation team, based on rules and inputs obtained from workshops.

All simple mappings are held in the **XXMX\_SIMPLE\_TRANSFORMS** table and the complex transformation are held in the **XXMX\_MAPPING\_MASTER** table. The complex transformations will need to be set up/configured by the delivery/implementation team.

## Simple Transformation

Simple transformations are simple 1:1 mapping and Maximise automatically refers to the **XXMX\_SIMPLE\_TRANSFORMS** table for the mapping/rules to be applied during transformation.

Transformations for sub-entities can be enabled or disabled in the **XXMX\_MIGRATION\_METADATA** table. The utilities package **XXMX\_DYNAMIC\_SQL\_PKG** and **XXMX\_UTILITIES\_PKG** are called internally to transform data into XFM tables.

Simple Transformation is performed in two steps:

1. Transfer the data to xfm table using the procedure transfer\_stg\_data
2. Transform the data using the simple transformation rule in procedure transform\_data

## Complex Transformation

Complex transformations rules must be coded in bespoke packages maintained by delivery/implementation team. The packages must be configured in the tables **XXMX\_CUSTOM\_EXTENSIONS** and **XXMX\_CUSTOM\_SUB\_EXTENSIONS**.

Complex transformation is performed only after simple transformations are completed. If complex transformation procedure is specific to a sub entity, then configure it in the table **XXMX\_CUSTOM\_SUB\_EXTENSIONS**.

## How to Link Simple Transforms

To map simple transformation to xfm columns, follow the steps below:

1. Populate column *TRANSFORM\_CODE* in the **XXMX\_SIMPLE\_TRANSFORMATIONS** table, for each *CATEGORY\_CODE*

Example:

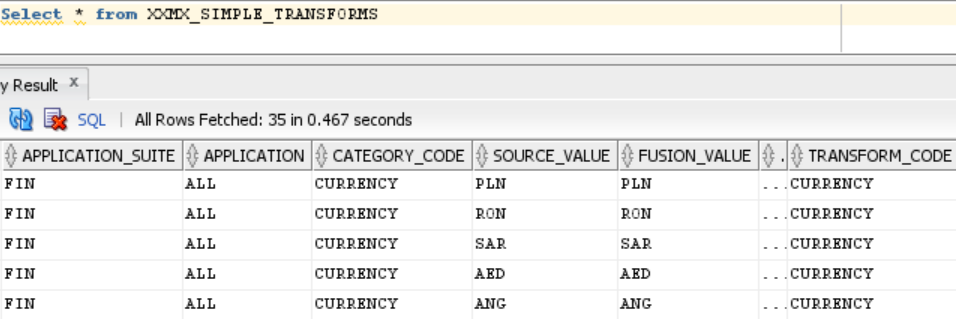


Figure 3 Link Simple Transformation

1. Populate column *TRANSFORM\_CODE* in the **XXMX\_XFM\_TABLE\_COLUMNS** table, for relevant table

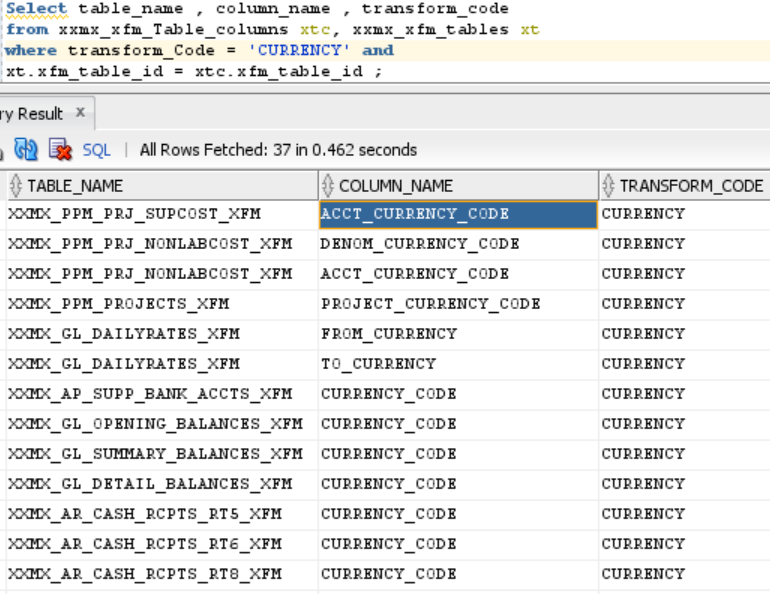


Figure 4 Populate TRANSFORM\_CODE

Use the excel sheet to list the custom mapping rules for both Simple and Complex Transformation Rules.



* Steps to load Maximise Mapping Tables are detailed in Maximise User Guide.docx.
* Data from the excel sheet must be loaded to Maximise tables **XXMX\_SIMPLE\_TRANSFORMS** and **XXMX\_MAPPING\_MASTER.**

## Maximise Load

## Maximise Configurations

* Table XXMX\_FILE\_LOCATIONS, XXMX\_XFM\_TABLES, XXMX\_XFM\_TABLE\_COLUMNS must be populated.

***Note****: All these configurations are part of Maximise Installation*

* Fusion import parameters must be setup in the table **XXMX\_DM\_ESS\_JOB\_DEFINITIONS**
* Table **XXMX\_DM\_ASSET\_BOOKS\_IN\_SCOPE** must be populated for Fixed Assets Fusion Migration.
* Table **XXMX\_FUSION\_BUSINESS\_UNITS** must be populated for all Finance Fusion migration.
* Table **XXMX\_DM\_FUSION\_DAS** must be populated for Journal Fusion Migration.

Package **XXMX\_FUSION\_LOAD\_GEN\_PKG** has two generic procedures to create HCM fusion File and Finance csv fusion file.

Before generating the file, few validations are checked.

* Mandatory columns for Fusion Load should not be blank
* There must be at least one column to be included in the fusion file as per configuration
* XFM tables should not be empty

File is then generated based on the *XFM\_COLUMN\_SEQ* in the **XXMX\_XFM\_TABLE\_COLUMNS** table for the fusion load file.

Fusion file can also be generated based on the column seq setup in the table **XXMX\_DM\_SUBENTITY\_FILE\_MAP**

HCM Data will be generated in the table **XXMX\_HDL\_FILE\_TEMP**

Finance Data will be generated in the table **XXMX\_CSV\_FILE\_TEMP**

# Maximise OIC and VBCS

Extract, Transformation and Load all the DM Phases are automated through OIC and VBCS UI interface.

## OIC Components

List of OIC process called from VBCS Screen.

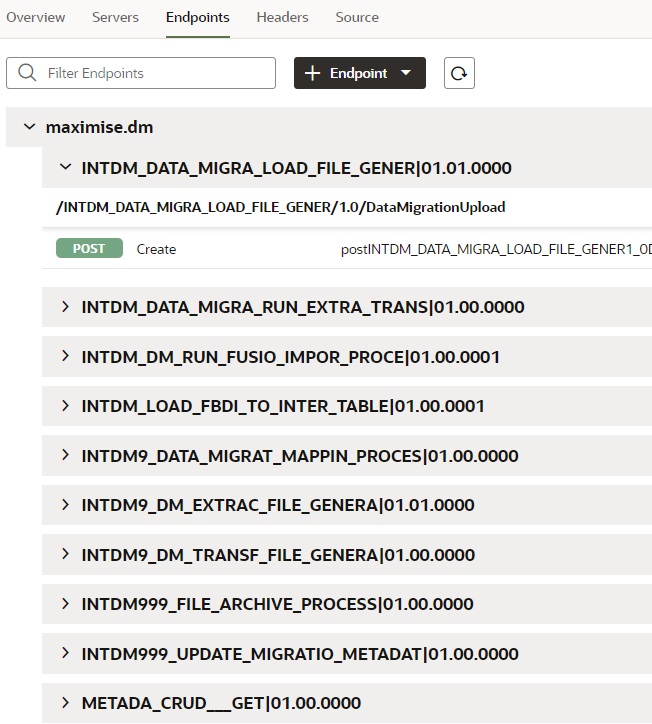
|  |  |
| --- | --- |
| OIC Process | Purpose |
| INTDM999005 Data Migration Mapping Process | OIC Integration to map the csv data into Maximise mapping tables XXMX\_SIMPLE\_TRANSFORMS and XXMX\_MAPPING\_MASTER |
| INTDM999001 Data Migration Run NONEBS | OIC to execute Extract and Transformation based on business Entity |
| INTDM999002 DM Extract File Generation | OIC to generate Extract file to OIC File Server Path |
| INTDM999003 DM Transform File Generation | OIC to generate transformation file to OIC File Server Path |
| INTDM999004 Data Migration Load File Generation | OIC to generates the load file in csv format, based on the Fusion Control file Format. |
| INTDM999006 Load FBDI to Interface Table | OIC to Loads the file to Fusion Interface tables |
| INTDM999007 DM Run Fusion Import Process | OIC to Import the data to Fusion. |
| INTDM999008 File Archive Process | File Archive process called across integration to archive the files in OIC File Server |

Extract and Transform process are currently designed to run as per business entity only.

Load process is modified to run per sub\_entity for HCM, Finance – GL Balances and Suppliers only.

Rest all the loads should be executed as per business entity.

## VBCS Components

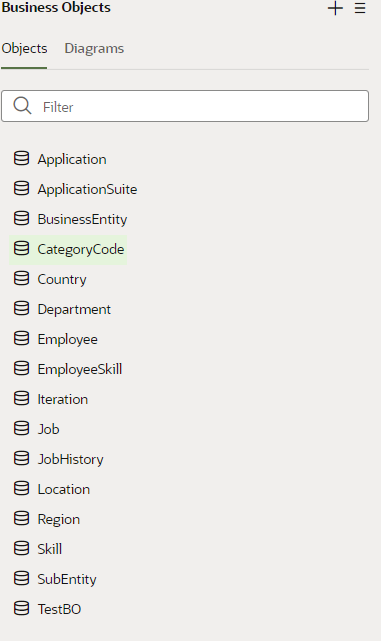
**All integrations are created as endpoints in VBCS.**

**Web Applications created in VBCS**

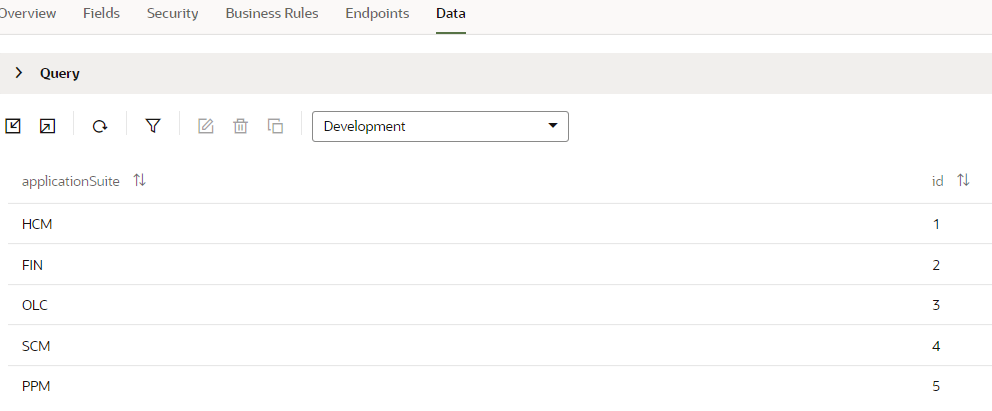
A picture containing text, device, receipt

Description automatically generated

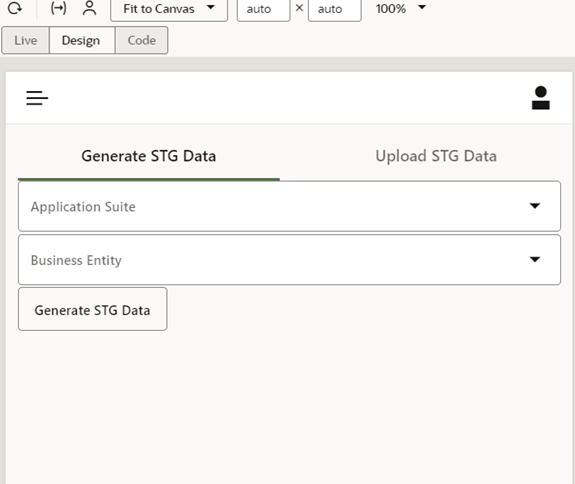
Business Objects Created in VBCS for XXMX\_MIGRATION\_METADATA Table



**Application Suite values**



**Web Application looks like below**



# 5. Overview

This section provides a high-level summary of the recent ETL process enhancement, highlighting both Batching and Non-Batching approaches.

## 5.1. Batching

## Batching, within the context of ETL (Extract, Transform, Load), involves collecting and processing data in chunks or batches rather than individually. This optimization enhances processing efficiency and resource utilization. The efficiency gains arise from reduced overhead associated with transactional operations when dealing with records collectively. Additionally, batching is crucial for managing substantial data volumes, allowing for parallel processing and horizontal scalability to accommodate increased data loads.

## 5.2. Non-Batching

Non-Batching describes a data processing method wherein individual records are processed one at a time without being grouped into batches. This implies that each record undergoes independent extraction, transformation, and loading, rather than being part of a larger collection. For scenarios with relatively small data volumes and manageable overhead in processing individual records, a non-batching approach may suffice.

# 6. Design Flow

This section outlines the Design Flow Diagram of Maximise DM Batching and Non-Batching process, illustrating how the data is displayed according to the provided input while Extracting into stg tables.

A diagram of a flowchart

Description automatically generated

# 7. Configuration

The following configurations must be set up for both Batching and Non-Batching.

## 7.1. Batching

* Verify that Batch\_Load column is ‘Y’ in the **XXMX\_MIGRATION\_METADATA**  table.
* For Finance, the details regarding batching is stored in the **LOAD\_BATCH** column of both the STG and XFM tables.
* Whereas for HCM, the details regarding batching is stored in the **BATCH\_NAME** column of both the STG and XFM tables.
* Ensure that the **COMMON\_LOAD\_COLUMN** in the **XXMX\_XFM\_TABLES** is updated for the corresponding business entity and sub-entity.
* Utilize the **Lookup\_Data\_Migration\_Extracts** OIC lookup to specify the desired batch count of records in the extract\_batch\_count field for extracting data in batches.
* Ensure that the below tables are populated.
* XXMX\_STG\_TABLES
* XXMX\_STG\_TABLE\_COLUMNS
* XXMX\_XFM\_TABLES
* XXMX\_XFM\_TABLE\_COLUMNS

## 7.2. Non-Batching

* Verify that Batch\_Load column is ‘N’ in the **XXMX\_MIGRATION\_METADATA** table.
* Ensure that the below tables are populated.
* XXMX\_STG\_TABLES
* XXMX\_STG\_TABLE\_COLUMNS
* XXMX\_XFM\_TABLES
* XXMX\_XFM\_TABLE\_COLUMNS

# 8. Technical Specification

This section outlines the technical specifications for batching and non-batching.

## 8.1. Batching

## 8.1.1. Extract

During the extraction phase, batching for the extraction process is enabled and initiated if **BATCH\_FLAG** is set to 'Y' based on the data from **xxmx\_migration\_metadata** for the corresponding Business Entity and Sub entity.

* To determine the desired batch count of records for extracting data in batches, the batch count is retrieved from the OIC lookup named **Lookup\_Data\_Migration\_Extracts**, specifically from the extract\_batch\_count field.
* The extraction process triggers **xxmx\_utilities\_pkg.batch\_extract\_data**, considering parameters such as the application suite, business entity, sub-entity, batch count, and phase. After the extraction is completed and data being transferred to STG tables.
* During Extraction for first sub entity sequence from xxmx\_migration\_metadata table, if the flag is set to ‘Y’, it calls **batch\_extract\_data** procedure from **xxmx\_utilities\_pkg** to batch the first sub entity sequence.
* Once the formation of batches is completed for the first sub\_entity\_seq, the 'Load\_batch' column for finance and 'Batch\_name' column for HCM will be updated with the batches in the STG tables. Details of batches is stored in **XXMX\_DM\_FILE\_BATCH** table.
* For other sub entity sequences during their extraction, it checks if sub\_entity\_seq1 is batched, only then batches the remaining sub\_entity sequences according to the common\_load\_column set in xxmx\_xfm\_tables.
* If sub\_entity\_seq1 is not batched, an exception is thrown in the xxmx\_module\_message table, indicating that batching must be completed for sub\_entity\_seq1 before proceeding further. Until this is done, further batching cannot be processed.
* The table below provides a list of tables, procedures, and description for each detailing their functionalities.

|  |  |
| --- | --- |
| Key | Description |
| BATCH\_EXTRACT\_DATA | This procedure handles the batching of extracted data. |
| XXMX\_DM\_FILE\_BATCH | This table stores information such as the table name, batch name, batch column, and sequence batch of the sub\_entity\_sequence. |
| XXMX\_MIGRATION\_METADATA | Table holds details of Business Entity and Sub Entity. Extract packages, Staging, Transformation and External Tables. Client can enable and disable any business entity or sub entity for Extract, Transform or Load. |
| XXMX\_MODULE\_MESSAGES | This table holds log messages for each phase – Extract, Transform and Load |
| XXMX\_UTILITIES\_PKG | Maximise Core utility package |
| XXMX\_DYNAMIC\_SQL\_PKG | Maximise Package for Transformation, Extract and Load |
| XXMX\_FIN\_STG\_EXTRACT\_PKG | Maximise Generic Package for Finance Extract and Transformation. |
| XXMX\_HCM\_STG\_EXTRACT\_PKG | Maximise Generic Package for HCM Extract and Transformation. |
| XXMX\_FUSION\_LOAD\_GEN\_PKG | Maximise Generic Package for Load file generation. |
| LOAD\_BATCH | This column holds the batched records for Finance in stg and xfm tables. |
| BATCH\_NAME | This column holds the batched records for HCM in stg and xfm tables. |
| BATCH\_LOAD | This column holds the flag for batching and non-batching in xxmx\_migration\_metadata table. |

## 8.1.2. Transform

The Transform approach remains consistent, with data being moved from STG to XFM through transform routines in alignment with the batching processes carried out during the transform phase.

## 8.1.3. Load

Following the completion of the extract and transform stages, the subsequent phase entails batching for the load process.

1. **Load Process:**

* The process starts by loading data into temporary tables ('xxmx\_csv\_file\_temp' for finance and 'xxmx\_hdl\_file\_temp' for HCM), and then recording this information in the 'XXMX\_LOADFILE\_STATUS\_LOG' table with a Load file ID and status set to 'R'. Subsequently, the data prepared in these temporary tables is ready for file generation on the server.

2. **Load File Generation in DB Server:**

* Load file generation on the DB server is managed by a scheduled integration in OIC named **'INTDM999015 DM Write Load File to DB Server (1.0)'**, running every 10 minutes. This integration selects Load file IDs marked with 'R' in the 'XXMX\_LOADFILE\_STATUS\_LOG' table and processes each ID sequentially. As it performs operations for each ID, it marks status it as 'i' in log table, indicating that it's in progress.
* The integration checks the 'batch\_load' flag in the 'xxmx\_migration\_metadata' table. If the flag is set to 'Y' (indicating batching), it invokes the 'XXMX\_WRITE\_ORACLEDB\_SCH' procedure, which internally calls the **'XXMX\_WRITE\_DBCS**' procedure from the 'xxmx\_utilities\_pkg' package.
* This procedure extracts line content from the 'xxmx\_csv\_file\_temp' table for finance or 'xxmx\_hdl\_file\_temp' table for HCM, writes the data to the DB file server in batches, and compresses the files using the **'xxmx\_zip\_file**' procedure from the **'xxmx\_utilities\_pkg'** package. The generated files are placed in a specific path provided in the log table.
* Once the file generation is complete, the integration updates the status column for the particular load\_file\_id to 'G' (generated) in the 'XXMX\_LOADFILE\_STATUS\_LOG' table.

3. **Moves File from DB server to FTP:**

* This process remains the same for both batching and non-batching scenarios.
* After the load file is generated, the next step involves transferring it from the DB server to the FTP. This task is accomplished through a scheduled integration in OIC named 'INTDM999017 Moves Load File from DB Server to FTP', which executes every 10 minutes.
* The integration selects load\_file\_ids with a 'G' status from the log table. It iterates through each load\_file\_id, retrieving the files from the DB server using a specific path and downloading the zip files. In the FTP server, if any old files exist for the particular Business Entity or Sub-entity folder path, the integration archives these old files into an archival folder. Subsequently, it uploads the downloaded zip files to the FTP server, deletes the files from the DB server, and updates the status to 'C' in the 'XXMX\_LOADFILE\_STATUS\_LOG' table which means load is completed and zipfiles are available in FTP server in particular business entity/sub entity folder.
* The table below provides a list of tables, status, integrations, and description for each detailing their functionalities during load process.

|  |  |
| --- | --- |
| Key | Description |
| Status 'R' | This status in **xxmx\_loadfile\_status\_log** table means Load data is generated in the temp tables and is ready to generate the file |
| Status 'i' | This status in **xxmx\_loadfile\_status\_log** table means that record is being processed and is in progress. |
| Status 'G' | This status in **xxmx\_loadfile\_status\_log** table means file is generated in DB server. |
| Status 'C' | This status in **xxmx\_loadfile\_status\_log** table means load data file is moved to the FTP server and same can be checked in the ftp folder. |
| INTDM999004 Data Migration Load File Generation V2 | Integration to populate load data in the XXMX\_CSV\_FILE\_TEMP/XXMX\_HDL\_FILE\_TEMP |
| INTDM999015 DM Write Load File to DB Server | Integration to write the Load data populated in database table into DB server as a Zip file. |
| INTDM999017 Moves Load File from DB Server to FTP | Integration to move Load File which is generated in DB server to FileZilla. |
| XXMX\_WRITE\_DBCS | This procedure writes the line content from 'xxmx\_csv\_file\_temp/hdl' table into the DB file server. |
| xxmx\_zip\_file | Procedure which zips the batched files in DB server |
| XXMX\_LOADFILE\_STATUS\_LOG | Table which holds the status of the Load process. |

## 8.2. Non-Batching

## 8.2.1 Extract

The non-batching process mirrors the approach used in the previous version of Maximise, where no batching procedures are invoked in this exception case. In this scenario, the Load\_Batch column for finance and Batch\_Name column for HCM are left null, indicating the absence of created batches. Consequently, files are then visible in a sequential single-file format within the FTP folder.

## 8.2.2. Transform

The Transform approach remains consistent, with data being moved from STG to XFM through transform routines in alignment with the non-batching processes carried out during the transform phase.

## 8.2.3. Load

After the extract and transform stages, the next step is the load process.

1. **Load Process:**

* The process starts by loading data into temporary tables ('xxmx\_csv\_file\_temp' for finance and 'xxmx\_hdl\_file\_temp' for HCM), and then recording this information in the 'XXMX\_LOADFILE\_STATUS\_LOG' table with a Load file ID and status set to 'R'. Subsequently, the data prepared in these temporary tables is ready for file generation on the server.

2. **Load File Generation in DB Server:**

* Load files are generated on the DB server through a scheduled integration in OIC named 'INTDM999015 DM Write Load File to DB Server (1.0)', running every 10 minutes. This integration selects Load file IDs marked with 'R' and iterates through each ID, setting its status to 'i' in the table which means in progress.
* For each load\_file\_id, depending on the batch\_load flag(which should be ‘N’) from the xxmx\_migration\_metadata table for the business entity and sub-entity, the system calls the procedure 'XXMX\_LOADDB\_ORACLEDB\_SCH.' This procedure then invokes 'XXMX\_LOADDB\_FILE' for each sub entities under particular BusinessEntity, writes line content from 'xxmx\_csv\_file\_temp' for finance and 'xxmx\_hdl\_file\_temp' for HCM into a file in Db server.
* Afterwards, it records details such as file name, server path, business entity, and load file ID, and updates the status to 'C' in XXMX\_LIST\_DB\_LOADFILE table once each sub entity file is written inside the DB server
* The integration subsequently checks the status of each file. If it is 'C', then it calls the 'XXMX\_ZIP\_FILE\_NONBATCHING' procedure from 'xxmx\_utilities\_pkg' to zip these files in Db server.
* Finally, updates the status to 'G' in 'XXMX\_LOADFILE\_STATUS\_LOG,' indicating that the file has been generated in the DB server for load\_file\_id.

3. **Moves File from DB server to FTP:**

* The process remains the same for both batching and non-batching scenarios.
* After the load file is generated, the next step involves transferring it from the DB server to the FTP. This task is accomplished through a scheduled integration in OIC named 'INTDM999017 Moves Load File from DB Server to FTP', which executes every 10 minutes.
* The integration selects load\_file\_ids with a 'G' status from the log table. It iterates through each load\_file\_id, retrieving the files from the DB server using a specific path and downloading the zip files. In the FTP server, if any old files exist for the particular Business Entity or Sub-entity folder path, the integration archives these old files into an archival folder. Subsequently, it uploads the downloaded zip files to the FTP server, deletes the files from the DB server, and updates the status to 'C' in the 'XXMX\_LOADFILE\_STATUS\_LOG' table which means load is completed and zipfiles are available in FTP server in particular business entity/sub entity folder.

|  |  |
| --- | --- |
| Key | Description |
| Status 'R' | This status in **xxmx\_loadfile\_status\_log** table means Load data is generated in the temp tables and is ready to generate the file |
| Status 'i' | This status in **xxmx\_loadfile\_status\_log** table means that record is being processed and is in progress. |
| Status 'G' | This status in **xxmx\_loadfile\_status\_log** table means file is generated in DB server. |
| Status 'C' | This status in **xxmx\_loadfile\_status\_log** table means load data file is moved to the FTP server and same can be checked in the ftp folder. |
| INTDM999004 Data Migration Load File Generation V2 | Integration to populate load data in the XXMX\_CSV\_FILE\_TEMP/XXMX\_HDL\_FILE\_TEMP |
| INTDM999015 DM Write Load File to DB Server | Integration to write the Load data populated in database table into DB server as a Zip file. |
| INTDM999017 Moves Load File from DB Server to FTP | Integration to move Load File which is generated in DB server to FileZilla. |
| XXMX\_LIST\_DB\_LOADFILE | This table updates the files present in the directory to the table |
| XXMX\_ZIP\_FILE\_NONBATCHING | Procedure which zips the files in DB server |
| XXMX\_LOADFILE\_STATUS\_LOG | Table which holds the status of the Load process. |
| XXMX\_LOADDB\_FILE | This procedure writes the line content from 'xxmx\_csv\_file\_temp/hdl' table into the DB file server. |

# 9. Batching and Non-Batching Snippets

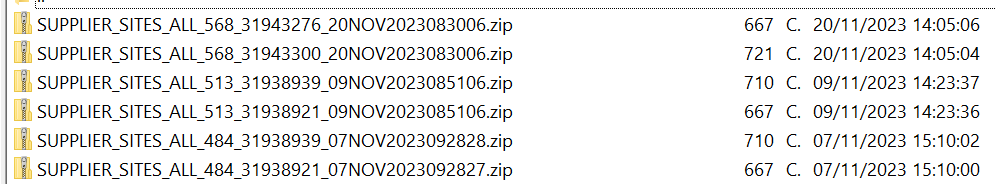
## Batching

* For an example, the files can be observed on the FTP server in the following format,

1. **Business entity level File Generation Example.**

Format: BusinessEntity\_ALL \_LoadFileId\_ BatchName \_DateMonthYearTimestamp

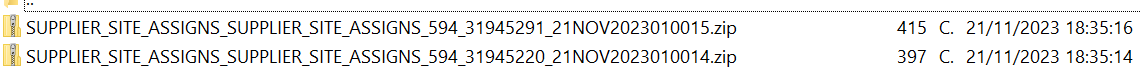
**Example:**

****

**2. SubEntity Level File Generation Example:**

Format: BusinessEntity \_SubEntity \_LoadFileId\_ BatchName \_DateMonthYearTimestamp

**Example:**

****

## 9.2. Non-Batching

* For an example, the files can be observed on the FTP server in the following format,

1. **Business entity level File Generation example.**

Format: BusinessEntity \_ALL\_ LoadFileId \_DateMonthYearTimestamp

Example:



**2. SubEntity Level File Generation Example:**

Format: BusinessEntity \_SubEntity\_ LoadFileId\_DateMonthYearTimestamp

