



Maximise Data Migration

User Guide

Release 22.1

@Copyright 2022 Version 1 – All Rights Reserved

Contents

1. Overview 5

1.1. About Maximise Data Migration 5

1.1. Target Audience 5

1.2. Related Documents 5

1.3. Terminology 5

1.4. Out of Scope 6

2. Maximise Data Migration Overview 7

2.1. Architecture 7

2.2. Extract 7

2.3. Transform 7

2.4. Load 7

3. Maximise Data Migration 9

3.1. Configuration 9

3.2. Maximise Object Validation 9

3.3. Maximise Extract 11

3.4. Maximise Transform 13

3.5. Maximise Load 17

3.6. Maximise Import 18

4. Maximise Custom Extensions 20

4.1. What are They? 20

4.2. Pre-requisites 20

4.3. How to extend and call custom code? 20

4.4. Integrate custom extensions with Maximise DM 22

**Version Control**

|  |  |  |  |
| --- | --- | --- | --- |
| Date | Version | Changed By | Reason for Change |
| 29-APR-2022 | 0.1 | Pallavi Kanajar | No previous version |
| 03-MAY-2022 | 0.2 | Praveen Nair | Review |
| 18-May-2022 | 0.3 | Praveen Nair | Review |
|  |  |  |  |

**Circulation List**

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

# Overview

## About Maximise Data Migration

Maximise Data Migration is Version 1’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 the most common data migration scenarios. Maximise Data Migration 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 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. Generate load files and perform import to Fusion Cloud all through a very simple easy to use user interface.

For project teams, Maximise Data Migration accelerator also allows extension capabilities to capture customer specific requirements.

## Target Audience

This guide is intended for anyone interested in using Maximise Data Migration on Oracle cloud data migration projects.

## Related Documents

* [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)

## Terminology

Some of the terminology used in this document and across Maximise Data Migration is listed 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 |

## Out of Scope

* 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
* Transformation and Load is out of scope for Maximise
* Customer specific customisations and mapping rules will not be covered under Maximise DM
* Data cleansing is outside the scope of Maximise Data Migration and should be performed before the Extract task

# Maximise Data Migration Overview

## Architecture

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

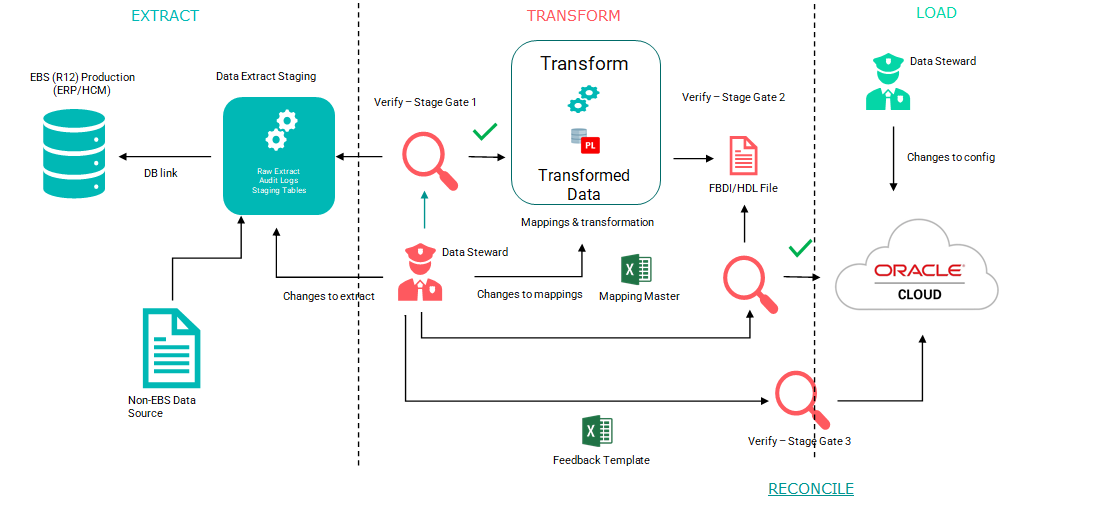


Figure 1 Maximise DM Architecture

## Extract

* 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.

* <<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

Data is moved, between STG and XFM, by transform routines, using <<Customer/Functional Consultant>> defined mappings, when initiated on a per entity basis. This includes technical “standard” mapping and transformation, alongside verification and validation checks to the newly configured Oracle Cloud (if any are defined in the Maximise Data Migration accelerator)

## 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 Data Migration

This section outlines the required configuration for Maximise Data Migration accelerator.

## Configuration

Create a standalone Database (Cloud DBAAS) with the configuration outlined below:

|  |  |
| --- | --- |
| Database Parameters | Description |
| DB Type | Standalone DB |
| Shape | VM. Standard2.2 or higher |
| OCPU Count | Minimum 2 |
| Network Bandwidth | 2 GBPS |
| Memory | 30 GB |
| Local Disk | Block Storage Only |
| Capacity type | 500GB Free Storage with On-demand configuration |
| Database System Version | 19.7.0.0.0 Standard Edition or higher |
| Character Set,  National Character Set | AL32UTF8, AL16UTF16 |
| Database Workload | OLTP |
| Pluggable Database Name | MXDM\_PDB1 |

## Maximise Object Validation

The table below provides a list of table names and description for each. These tables are created as part of the Maximise Data Migration 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 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 Finance Extract and Transformation. |
| XXMX\_FUSION\_LOAD\_GEN\_PKG | Maximise Generic Package for Load file generation. |

## Maximise Extract

The section below details the pre-requisites for performing Extract and the different modes available to extract data.

### Pre-requisite

* Make sure database link is operating

***Note****: This is mandatory for EBS customers only*

* All Maximise database objects are in Valid Status
* All Staging tables, Transformation Tables are installed. Details of objects can be obtained from the **XXMX\_MIGRATION\_METADATA** table
* All configuration objects, covered under section 3.2 are installed and valid
* The Maximise tables below should be populated for the relevant business entities, for Extract
  + XXMX\_MIGRATION\_METADATA
  + XXMX\_MIGRATION\_PARAMETERS
  + XXMX\_CORE\_PARAMETERS
  + XXMX\_LOOKUP\_VALUES
  + XXMX\_STG\_TABLES
  + XXMX\_XFM\_TABLES
* For non-EBS customers, data file should be placed in the Oracle path defined in the directory SOURCE\_DATAFILE. The path for this directory should be maintained in the XXMX\_MIGRATION\_PARAMETERS table. By default, the Maximise team will configure the path as ‘/tmp’ for the customer. The implementation team can change it as per their requirement

### Extract Mode – PLSQL

The sections below highlight the steps required to extract data using SQL Developer for both Finance and HCM.

#### Finance and SCM

* Populate the following parameters in the XXMX\_MIGRATION\_PARAMETERS table and enable or disable, as required.
  + Operating Unit (PARAMETER\_CODE=’ORGANIZATION\_NAME’)
  + Business group (PARAMETER\_CODE=’BUSINESS\_GROUP\_NAME’)
  + GL Ledger Names (PARAMETER\_CODE=’LEDGER\_NAME’)
* List of Maximise provided parameters are attached here for reference. Implementation teams can enable them or disable them appropriately



* In SQL Developer, change buffer size to 20000.



***Note:*** *FILE\_SET\_ID is mandatory for non-EBS customers only*

#### HCM

* Populate the following parameters in the XXMX\_MIGRATION\_PARAMETERS table and enable or disable, as required.
  + Person Types (PARAMETER\_CODE=PERSON\_TYPE’)
  + Business Groups (PARAMETER\_CODE=’BUSINESS\_GROUP\_NAME’)
  + Payroll (PARAMETER\_CODE=’PAYROLL\_NAME’)
  + Migration Dates (PARAMETER\_CODE=’MIGRATION\_DATE\_FROM’ and ‘MIGRATION\_DATE\_TO’)
* In SQL Developer, change buffer size to 20000



***Note:*** *FILE\_SET\_ID is mandatory for non-EBS customers only*

### Extract Mode – VBCS

The sections below highlight the steps required to extract data using the Maximise DM user interface.

#### Generate STG Data

* Select the **Application Suite** and **Business Entity** to extract the data from the source EBS tables

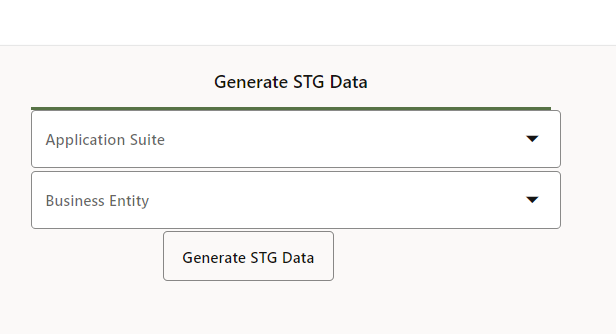


Figure 2 Maximise UI-Generate STG Data

* For non-EBS customers, new parameter will be visible to pass Mandatory file\_set\_id from Source Data File
* The VBCS API routines in turn calls OIC workflow processes to launch the PLSQL Extract
* The result of the processing can be reviewed by querying the staging tables



#### Upload STG Data

Select the Application Suite, Business Entity, Iteration and Migration Set ID to generate the extracted data file. The new file will be placed in the pre-configured FTP path and can be used for reconciliation purposes.

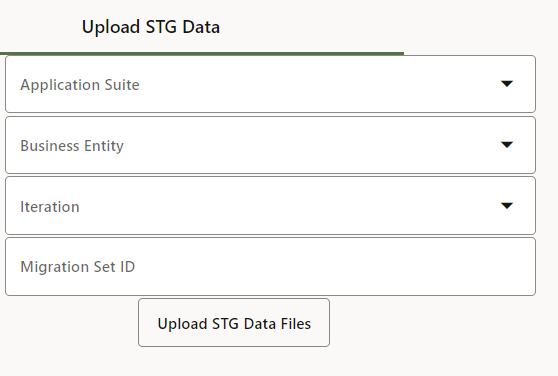


Figure 3 Maximise UI-Upload STG Data

## Maximise Transform

### Pre-requisite

Download the mapping master spreadsheet to capture all simple and complex transformations. This is required for Maximise DM to perform transformations on extracted data.



* Merge all the mapping to Master Tab using the macro “**MERGE\_SHEETS**”
* **Navigation:** *View > Macros > View Macros > Select Merge sheets > Run*

Graphical user interface, application, table, Excel

Description automatically generated

Figure 4 Maximise Mapping Spreadsheet

* Create a new workbook for the **Master** tab. Save the new workbook as a csv file
* Right-click on the **Master** tab and select ***Move or Copy…***

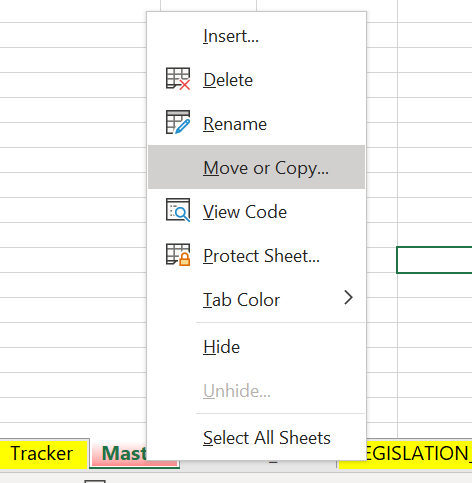


Figure 5 Maximise Mapping-New Worksheet

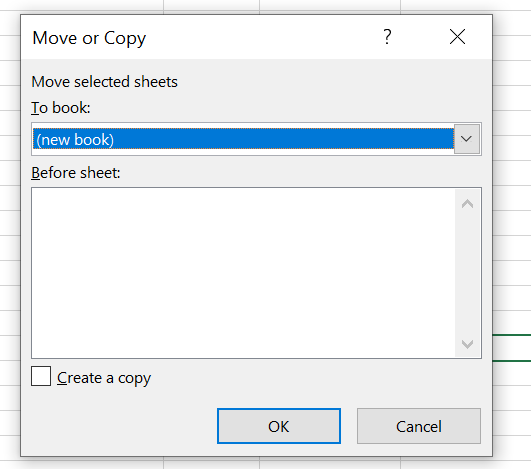


Figure 6 Maximise Mapping-Copy

* Click **OK**
* Save the document as a **CSV** file

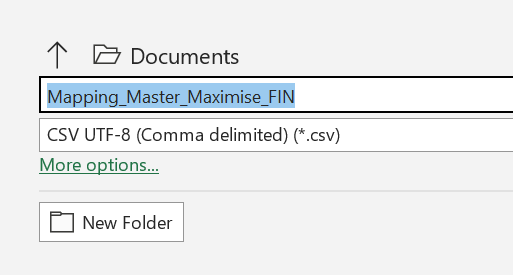


Figure 7 Maximise Mapping-New File

* Generated **CSV** should be placed in the pre-defined FTP server path

***UserName :*** *<OIC Username>*

***Password:*** *<OIC Password>*

***File Path:*** */home/user/<OIC Username>/client\_csv*

* Generated **CSV** file can be loaded to OIC Integration - to map the data in the Maximise tables **XXMX\_SIMPLE\_TRANSFORMS** and **XXMX\_MAPPING\_MASTER**

OIC Integration Name is “***INTDM999005 Data Migration Mapping Process***”

### Transform from PLSQL

* After completing the mapping, both the Simple and Complex Transformation data are moved to the respective Maximise tables
* Get Migration\_Set\_ID from Staging tables using the following call:



***Note:*** *FILE\_SET\_ID is mandatory for non-EBS customers only*

* Transformed data will be inserted into Maximise XFM Tables.

### Transform from VBCS

#### Generate XFM Data

* Select the Application Suite, Business Entity to Transform the data from Staging tables
* Requirement is staging should have only one Migration\_Set\_ID

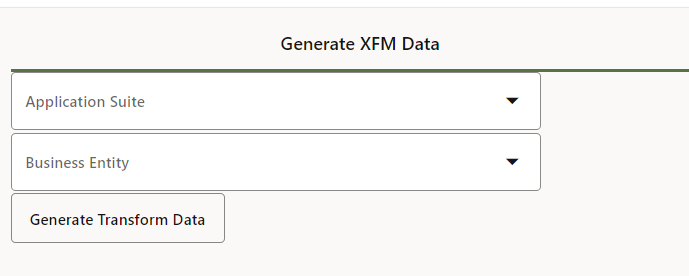


Figure 8 Maximise Transform-Generate XFM Data

* For Non-EBS customers new parameter will be visible to pass mandatory File\_Set\_Id from Source Data File.
* The VBCS API routines in turn calls OIC workflow processes to launch the PLSQL Transformation
* Once the Transformation process is successful (from PLSQL or VBCS), check the respective transformation tables to verify if data is populated. The list of transformation tables can be obtained from the **XXMX\_MIGRATION\_METADATA** table



#### Upload XFM Data

Select the Application Suite, Business Entity, Iteration and Migration Set ID to generate the transformed data file. This file is then placed in the pre-configured OIC FTP Path location, for reconciliation purposes.

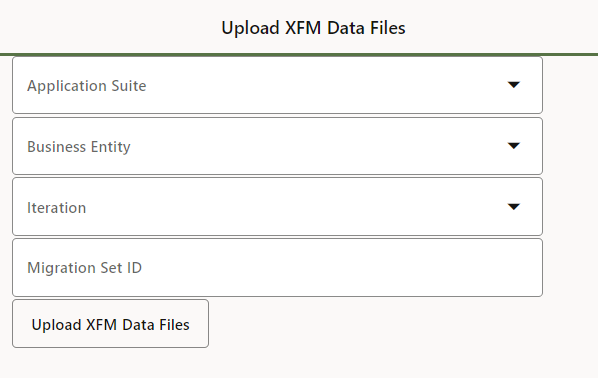


Figure 9 Maximise Transform-Upload XFM Data File

## Maximise Load

### Pre-requisite

* File path is setup in the **XXMX\_FILE\_LOCATIONS** table
* Populate the **XXMX\_XFM\_TABLES, XXMX\_XFM\_TABLE\_COLUMNS** for columns *fusion\_template\_field\_name, field\_delimiter, mandatory* and *Include\_in\_outbound\_file*

***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 Migration.
* Table **XXMX\_FUSION\_BUSINESS\_UNITS** must be populated for all Finance migration.
* Table **XXMX\_DM\_FUSION\_DAS** must be populated for Journal Migration.

### Load from PLSQL

Execute the below package in SQL Developer to generate the load file



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

### Load from VBCS

#### Generate Load File

* To generate the Load File, populate the parameters for Application Suite, Business Entity, Sub Entity, and Iteration

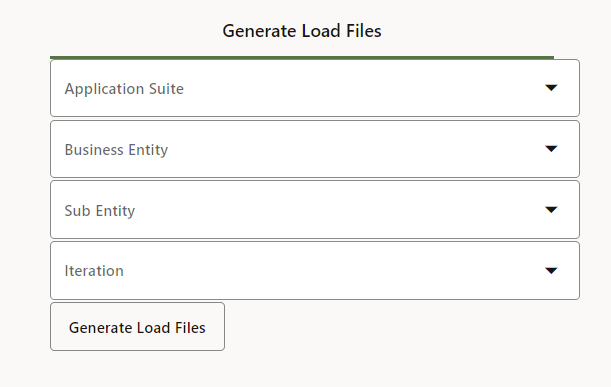


Figure 10 Maximise Load-Generate Load Files

* HCM data will be created in **XXMX\_HDL\_FILE\_TEMP** and finance data will be created in the **XXMX\_CSV\_FILE\_TEMP** table
* .**CSV** or .**DAT** file is created in the pre-defined FTP location

#### Load File to Fusion Interface

* This step is applicable for Finance only. Populate the parameters - Application Suite, Business Entity, Sub Entity, and Iteration.

***Note:*** *This step is not applicable for HCM (Fusion HDL for HCM processes both load and import)*

* This step will load the Interface tables in Fusion for Finance , SCM and PPM.

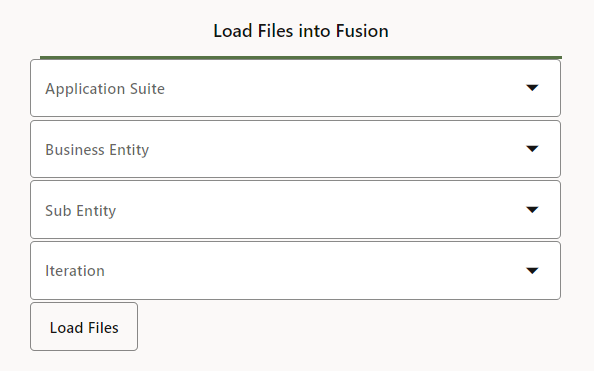


Figure 11 Maximise Load-Load Files To Fusion

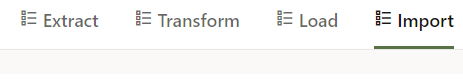
## Maximise Import

### Pre-requisite

* Table **XXMX\_DM\_ESS\_JOB\_DEFINITIONS** must be setup for capturing import parameters
* Fusion connection details must be working, and Fusion user must have the required data access
* Import functionality is automated only in VBCS

### Import from VBCS

Import works for both Finance and HCM. The data is imported to Fusion Base Tables from this screen.



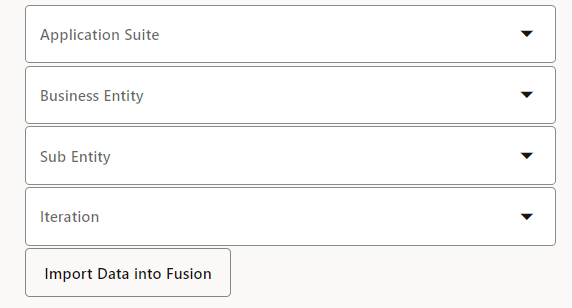


Figure 12 Maximise Import-Import to Fusion

# Maximise Custom Extensions

## What are They?

Custom extensions in Maximise Data Migration, provide the Version 1 implementation team functionality to extend Maximise Data Migration to capture customer specific requirements for validation etc. The implementation teams can add custom rules to existing Extract and Transform routines.

Custom extensions can be added to existing business entities/sub entities in Maximise DM.

This section will not cover how to add new business entities/sub entities in Maximise DM.

## Pre-requisites

* The custom code units should be part of a package definition. Custom Code for extension should be added in a new package maintained by Delivery team only.
* The procedure should follow a specific set of parameters, as per Maximise definition to integrate with Maximise
* XXMX\_CUSTOM\_EXTENSIONS and XXMX\_CUSTOM\_SUB\_EXTENSIONS should be populated for business entity or sub entity custom logic extension. Shown in section 4.4.

## How to extend and call custom code?

Create PL/SQL code unit to extend the existing extract and transform code

### Sample Extension - Extract

* Create the custom package in XXMX\_CORE Schema for extension on Business entity



Figure 13 Extract extension sample for a business Entity

* Create the custom package in XXMX\_CORE Package for extension on Sub entity



Figure 14 Extract extension sample for a business Sub-Entity

### Sample Extension - Transform

If complex transformation rules are required, define PL/SQL code units with the rules and integrate it with Maximise

**Note:** *We recommend keeping the code units grouped in a package for ease of maintenance*

* Create the custom package in XXMX\_CORE schema for extension on Business entity.



* Create the custom package in XXMX\_CORE Schema for extension on Sub entity.



## Integrate custom extensions with Maximise DM

### Extract

To integrate business entity extensions with Maximise, create an entry in the XXMX\_CUSTOM\_EXTENSIONS table as follows:



To integrate business sub-entity extensions with Maximise, create an entry in the XXMX\_CUSTOM\_EXTENSIONS table as follows:



### Transform

To integrate business entity extension with Maximise, create an entry in the table XXMX\_CUSTOM\_EXTENSIONS



To integrate business sub-entity extension with Maximise, create an entry in the table XXMX\_CUSTOM\_EXTENSIONS



# Maximise Add New Business Entity or Sub Entity

## What are They?

* If Delivery teams wants to integrate new business entity into Maximise, which is not part of the Maximise Toolkit. Please follow the steps below.

## How to Integrate?

* Delivery team can create the extract package and required staging and Transformation tables.
* Make sure staging and transformation tables have columns needed for the FBDI/HDL . It can also have extra columns if needed by business.
* Extract package should be in the format of the package “**xxmx\_fin\_<sub\_entity>\_ext\_pkg.sql**” provided in the toolkit for FIN and SCM application.
* Extract package should be in the format of the package “**xxmx\_hcm\_<sub\_entity>\_ext\_pkg.sql**” provided in the toolkit for HCM application.
* Once the tables and packages are ready. Make an entry in the table xxmx\_migration\_metadata for the extract shown as below.

**INSERT INTO xxmx\_core. xxmx\_migration\_metadata**

**(**

**metadata\_id**

**, application\_suite**

**, application**

**, business\_entity\_seq**

**, business\_entity**

**, sub\_entity\_seq**

**, sub\_entity**

**, entity\_package\_name**

**, sql\_load\_name**

**, stg\_procedure\_name**

**, stg\_table**

**, simple\_xfm\_performed\_by**

**, xfm\_procedure\_name**

**, xfm\_table**

**, file\_gen\_performed\_by**

**, file\_gen\_procedure\_name**

**, data\_file\_name**

**, data\_file\_extension**

**, file\_group\_number**

**, enabled\_flag**

**, file\_gen\_package**

**)**

**VALUES**

**(**

**xxmx\_migration\_metadata\_ids\_s. NEXTVAL**

**, <application\_suite> --'HCM'**

**, <application> --'HR'**

**, <BusinessEntity\_sequence> --for new business entity start from 100 onwards**

**, <BusinessEntity Name > --'BANKS\_AND\_BRANCHES'**

**, <SubEntity\_Sequence> -- 1, 2 or 3**

**, <SubEntity Name > --'BANKS'**

**, <Extract Package> --'xxmx\_banks\_and\_branches\_pkg'**

**, NULL**

**, <Procedure Name>**

**, <Staging table Name> --'XXMX\_BANKS\_STG'**

**, 'OIC' (HardCode)**

**, NULL**

**, <Transformation Table Name> --'XXMX\_BANKS\_XFM'**

**, 'OIC' --(HardCode)**

**, <Custom Load file generation procedure > --'banks\_gen'**

**, <Fusion File Name > --'Bank.dat'**

**, <Fusion File Extension> --'dat'**

**, 1 --(Hardcode)**

**, 'Y'-- (HardCode)**

**, <Custom Load file Generation Package > --'xxmx\_hcm\_hdl\_file\_gen\_pkg');**

**Note: You can also reach out to Maximise to guide you with this process.**

# Maximise Release Process

## Bit Bucket

Maximise Release Installation scripts will be placed in Bitbucket under production release folder in Bitbucket.

[https://git.version1.com/projects/VESA/repos/maximise-dm-production-release](https://git.version1.com/projects/VESA/repos/maximise-dm-production-release/browse/Install/EXTENSION/xxmx_create_arch_dbi_script.sql)

Follow the steps in the document “Maximise\_Installation\_Guide.docx” for installation.

# Maximise Support

## How to raise support tickets?

For Maximise Related Queries, Issues, Enhancements please log a ticket in the below link

<https://dev.azure.com/Version1InnovationLabs/Maximise%20Data%20Migration/_workitems/create/Issue>



[**www.version1.com**](http://www.version1.com)

Graphical user interface, text, application, email, Teams

Description automatically generated