



Junction Solutions

Where Platform Meets Inspiration

Introduction to the Data Migration Framework (DMF) in
Microsoft Dynamics
WHITEPAPER

Junction Solutions documentation 2012

All material contained in this documentation is proprietary and confidential to Junction Solutions, Inc and subject to the nondisclosure provisions of the applicable Junction Solutions, Inc agreement. This material is for informational purposes only. Junction Solutions, Inc is not liable for any damages in connection with the use of this information.

No part of this documentation may be reproduced, stored in a retrieval system, or transmitted in any form or by any means, including, but not limited to, electronic, graphic, mechanical, photocopying, recording, or otherwise without the prior written permission of Junction Solutions, Inc.

This documentation is subject to change without notice, and Junction Solutions, Inc does not warrant that the material contained in this documentation is free of errors. Any errors found in this document should be reported to Junction Solutions, Inc in writing.

Table of Contents

Introduction	4
Resources	4
Framework overview	4
DATA ENTITIES	5
Framework components	6
SOURCE DATA FORMAT	6
TARGET ENTITIES	9
PROCESSING GROUP.....	10
MAPPING	10
Framework process	14
LOAD SOURCE DATA.....	14
STAGING DATA VALIDATION	15
PROCESS STAGING DATA TO TARGET ENTITIES	17
Installation	20

Introduction

Microsoft Dynamics® AX 2012 has undergone some significant design changes to the application and the table structures. Even though moving to a normalized data base schema allows for a more standardized data structure, it also makes data migration a much more complex topic.

Historically data migration has been done through custom written import scripts created by the consulting team. They typically followed the path of reading in data files; convert/map values and validate incoming data. There was no out of the box migration framework that would provide functionality to import, validate and process records into the Microsoft Dynamics AX entities.

This changed when Microsoft released its Data Migration Framework (DMF) in mid-2012. The new framework provides functionality to load the most common data entities (e.g. Customer, Vendors, Items) utilizing a combination of SQL SSI packages as well as AX business logic.

This document will talk about the basic framework and components of the Data Migration Framework. There will be a second whitepaper describing how to modify existing and creating new entities.

Resources

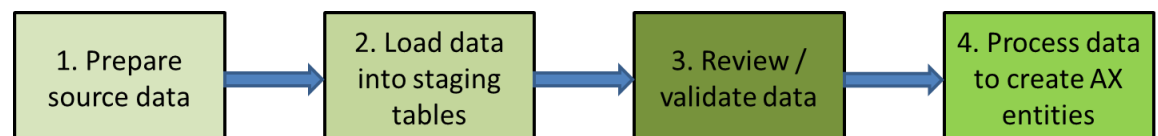
For more information about DMF, refer to the TechNet library in the following location:

- <http://technet.microsoft.com/en-us/library/jj225591.aspx>

Framework overview

The Data Migration Framework (DMF) is a tool provided by Microsoft to make data migration of complex data structures easier and more standardized. Other tools are available, but the DMF is definitely a big step forward to simplify and optimize data migration.

The general flow data elements take through the framework can be described in four steps:



1. Prepare source data:

Through manual or automated processes the source data files have to be created. The framework currently only supports text files as source. Future releases will include support for other input formats; for example ODBC data connections.

2. Load data into a staging table:

The data is loaded into a staging table through a dynamically generated SSIS package.

3. Review/Validate data:

The framework provides the ability to check imported value against existing referenced values to ensure referential integrity.

4. Process data to create AX entities:

In the last step the imported data is processed and the AX internal data records/entities are created.

Each of the steps is described in more detail in the following sections.

Data Entities

The following entities are currently supported out of the box:

- Bill of materials (BOM)
- Contacts
- Contact addresses
- Customer
- Customer addresses
- Dimension
- Employees
- Employee addresses
- Inventory journal
- Ledger balances
- Main account
- Open invoices (accounts payable)
- Open invoices (accounts receivable)
- Open sales orders
- Open purchase orders
- Price disc
- Product
- Project
- Unit of measure
- Vendor
- Vendor address

Framework Components

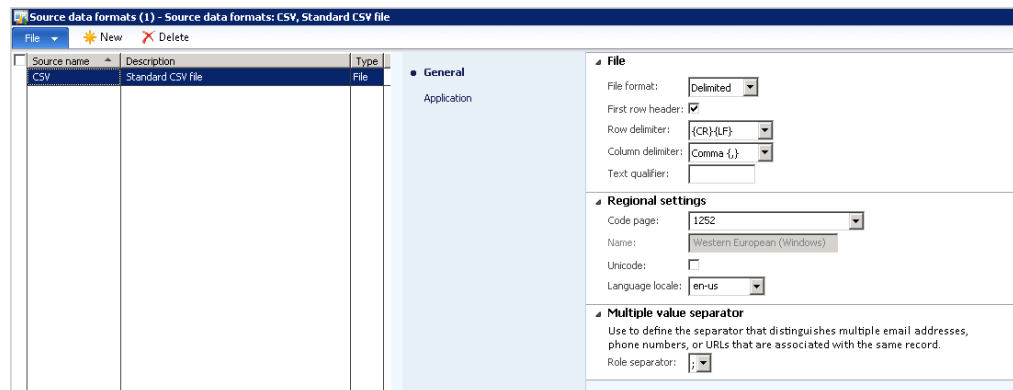
The framework includes the following components which are discussed in more detail in the following sections:

- **Source Data Format:**
Describes the formatting characteristics of the import file structure.
- **Target Entities:**
Defines the final Dynamics AX table records created during the final data processing.
- **Processing Groups:**
Provides an organizational setup for grouping import files to be processed together sequentially.
- **Mapping:**
Outlines the field mappings between the source file and staging tables.

Source Data Format

The source data format allows the user to specify parameters defining the import file. Multiple formats can be setup; however it is recommended that the same data format is used across different files. Support exists to accept both fixed-length and character delimited files.

The source data format allows us to setup various parameters regarding the type of source file that is processed.



Source name	Description	Type
CSV	Standard CSV file	File

General
Application

File
File format: Delimited
First row header:
Row delimiter: {CR}{LF}
Column delimiter: Comma (,)
Text qualifier:

Regional settings
Code page: 1252
Name: Western European (Windows)
Unicode:
Language locale: en-us

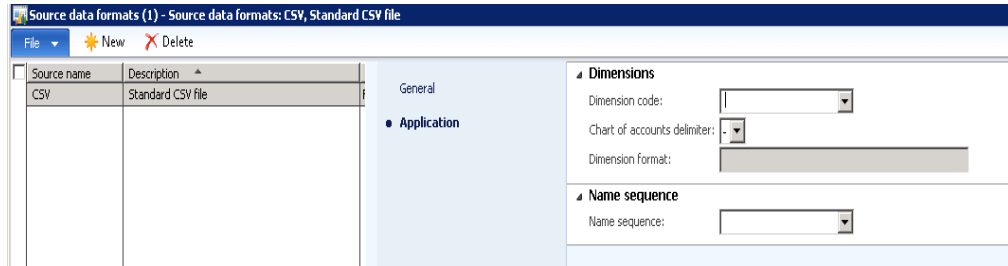
Multiple value separator
Use to define the separator that distinguishes multiple email addresses, phone numbers, or URLs that are associated with the same record.
Role separator: ;

Source Data Formats Form – General Tab

Use the following table for information on the **Source Data Format Form – General** tab:

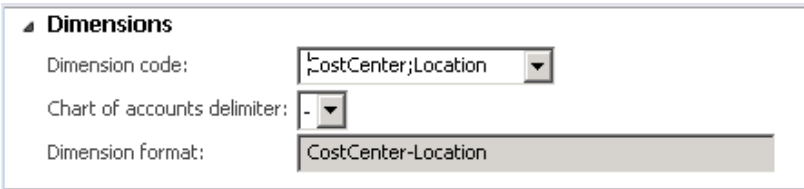
Field	Description
Type	The medium of import; currently “File” is selected by default and cannot be changed.
File format	Defines if the file is character delimited or if every column is fixed length.
First row header	Indicates the first row in the file is header information.
Row/Column delimiter	Character indicating the next column/row in the file.
Text qualifier	<p>Character used to encase text, for example double quotes. Any delimiter symbols found with the text qualifiers will be ignored and parsed as part of the text.</p> <p>Example: Lucky Luke, “123 Street Ave, Suite 5”, California</p> <p>(NOTE: Since the entire address is surrounded by double quotes (the text qualifier), the comma after “Ave.” is not considered a column delimiter)</p>
Regional settings (FastTab fields)	Allows specifying specific language settings if the data is loaded in any other format than default language, date and time format.
Multiple value separator (FastTab)	<p>A source file column can potentially contain multiple values. A typical example of this are multiple phone numbers assigned to a customer or contact. The parameter allows setting the character used to separate them.</p> <p>Important: This character has to be different than the column delimiter to ensure the values are treated as one column. The system currently does not validate this setup.</p>

The **Application** tab contains application specific setup information that determines how imported data elements are processed.



The screenshot shows a software interface for configuring source data formats. A table on the left lists 'Source name' (CSV) and 'Description' (Standard CSV file). The 'Application' tab is active, displaying settings for 'Dimensions' (Dimension code, Chart of accounts delimiter, Dimension format) and 'Name sequence' (Name sequence).

Use the following table for information on the **Source Data Format Form – Applications** tab:

Field	Description
Dimension code	This property allows enabling the financial dimensions used in the source data.
Chart of account delimiter	Selects the character that is used to separate the financial dimensions in the source file. Uses the same constraint as the “Multi value separator”; It must be different than the row delimiter.
Dimension Setup	If the setup below is chosen, a column in the source file would have to contain the following value: Warehouse-Pleasanton 
Name sequence	This parameter is used for the global address book entry to determine the order and format the name is going to be displayed.

Target Entities

The **Target entities** form shows the relation between staging table and the final entity (the Dynamics AX record(s)).

Type	Entity name	Staging table	Entity class	Target entity
	Asset	DMFAssetEntity	DMFAssetEntityClass	DMFAssetTargetEntity
	BOM	DMFBOMEntity	DMFBOMEntityClass	DMFBOMTargetEntity
	BOM version	DMFBOMVersionEntity	DMFBOMVersionEntityClass	DMFBOMVersionTargetEntity
	Contact address	DMFContactPersonAddressEntity	DMFContactPersonAddressEntityClass	DMFContactPersonAddressTargetEntity
	Contact person	DMFContactPersonEntity	DMFContactPersonEntityClass	DMFContactPersonTargetEntity
	Customer	DMFCustomerEntity	DMFCustomerEntityClass	DMFCustomerTargetEntity
	Customer address	DMFCustomerAddressEntity	DMFCustomerAddressEntityClass	DMFCustomerAddressTargetEntity
	Dimension	DMFDimensionAttributeValueEntity	DMFDimensionEntityClass	DMFDimensionAttributeValueTargetEntity
	Employee	DMFEmployeeEntity	DMFEmployeeEntityClass	DMFEmployeeTargetEntity
	Employee address	DMFEmployeeAddressEntity	DMFEmployeeAddressEntityClass	DMFEmployeeAddressTargetEntity
	Inventory journal	DMFInventJournalEntity	DMFInventJournalEntityClass	DMFInventJournalTargetEntity
	Ledger balance	DMFLedgerJournalEntity	DMFLedgerBalanceEntityClass	DMFLedgerJournalTransEntity
	ImBOM	DMFBOMEntity	DMFBOMEntityClass	DMFBOMTargetEntity
	Main account	DMFMainAccountEntity	DMFMainAccountEntityClass	DMFMainAccountTargetEntity
	Price disc	DMFPriceDiscAdmTransEntity	DMFPriceDiscAdmTransEntityClass	DMFPriceDiscAdmTransTargetEntity
	Product	DMFProductEntity	DMFProductEntityClass	DMFProductTargetEntity
	Project	DMFProjTableEntity	DMFProjTableEntityClass	DMFProjTableTargetEntity
	Purchase order header	DMFPurchTableEntity	DMFPurchTableEntityClass	DMFPurchTableTargetEntity
	Purchase order line	DMFPurchLineEntity	DMFPurchLineEntityClass	DMFPurchLineTargetEntity
	Sales order header	DMFSalesTableEntity	DMFSalesTableEntityClass	DMFSalesTableTargetEntity
	Sales order line	DMFSalesLineEntity	DMFSalesLineEntityClass	DMFSalesLineTargetEntity
	Sales table history	DMFjshistSalesTableEntity	DMFjshistSalesTableEntityClass	DMFjshistSalesTableEntity
	Units	DMFUnitOfMeasureEntity	DMFUnitOfMeasureEntityClass	DMFUnitOfMeasureTargetEntity
	Vend invoice journal	DMFVendInvoiceJournalEntity	DMFVendInvoiceJournalEntityClass	DMFVendInvoiceJournalTargetEntity
	Vendor	DMFVendorEntity	DMFVendorEntityClass	DMFVendorTargetEntity
	Vendor address	DMFVendorAddressEntity	DMFVendorAddressEntityClass	DMFVendorAddressTargetEntity

The staging table is a raw intermediate collection of data imported from the source data file. The end result can be an entity that exists as multiple records (e.g. customer and addresses). The staging table is a flat representation of all potential data elements. When processing the staging data, the system splits the data and creates the necessary AX records.

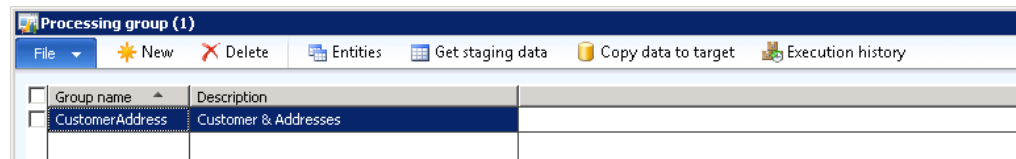
The linkage between the two elements is done through a mapping that can be automatically generated or manually added through the “Modify target mapping” form. Since the mappings are rarely changed, this will not be covered in the current document.

The entity class can be used to execute additional business logic during processing the staging data. A typical example is the creation of the necessary global address book entries when creating a customer. Entity classes and their functionality will be covered in detail in the next document looking at creating/modifying existing entities.

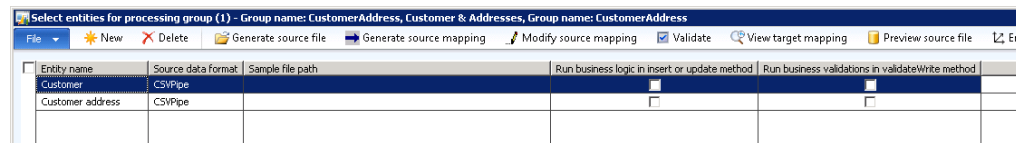
Processing Group

Once all the setup has been completed, a processing group can be created. Each processing group can contain one or multiple entities that will be processed together. An example would be to process customer and address information in one run.

The processing group form gives the user the options to load source data into the staging table(s), validate imported data and process/re-process the staged data into the target entities.



It is important to notice that entities will be processed in the order they were added to the processing group. This has to be taken into account when importing dependent entities such as “the customer has to exist before addresses can be imported”.



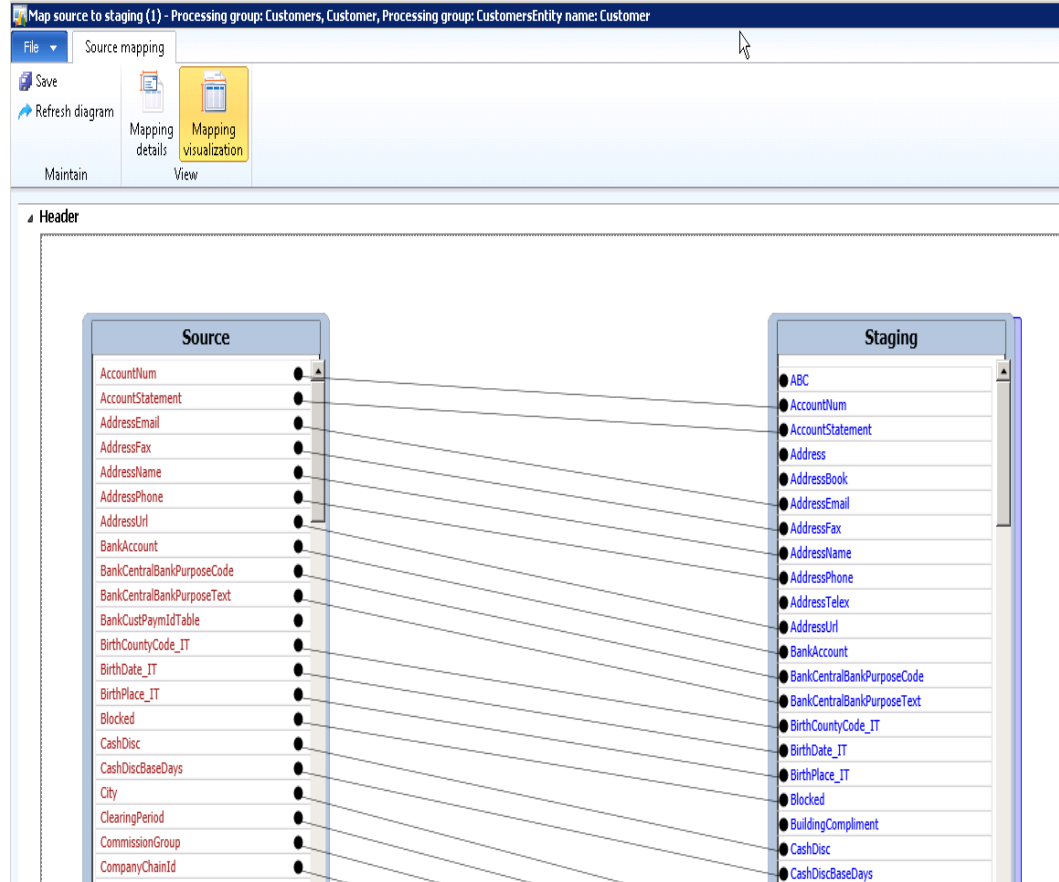
Each entity requires the source data format to be specified. This ensures that the source data is properly read and processed.

Mapping

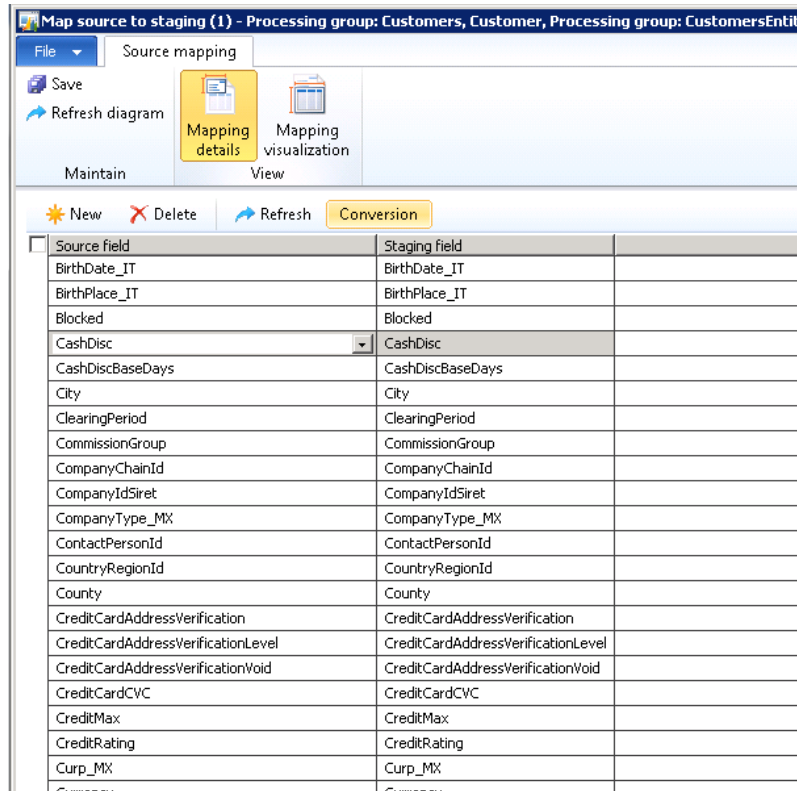
The next and most critical step is to have the correct mapping between the source data file and the staging table in AX. The mapping will ensure all values are properly imported into the correct table fields.

If the source file already exists, it can be selected as sample file and the “Generate source mapping” function can be used. This function will read the first row of the file and generate the mapping based upon the field names. Any file that is used as sample file therefore has to include header information.

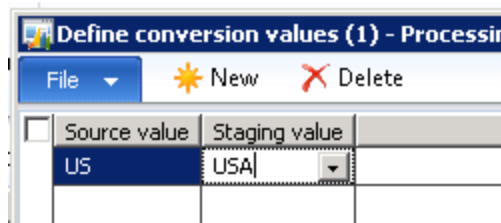
If the function cannot find a match based upon the given structure no mapping is generated and the user has to manually create the mapping by using the “Modify source mapping” function.



Through simple drag and drop the relation between the source and staging entity can be established. If a user prefers a table view of the existing/missing mappings the detailed view can be opened by selecting "Mapping details".

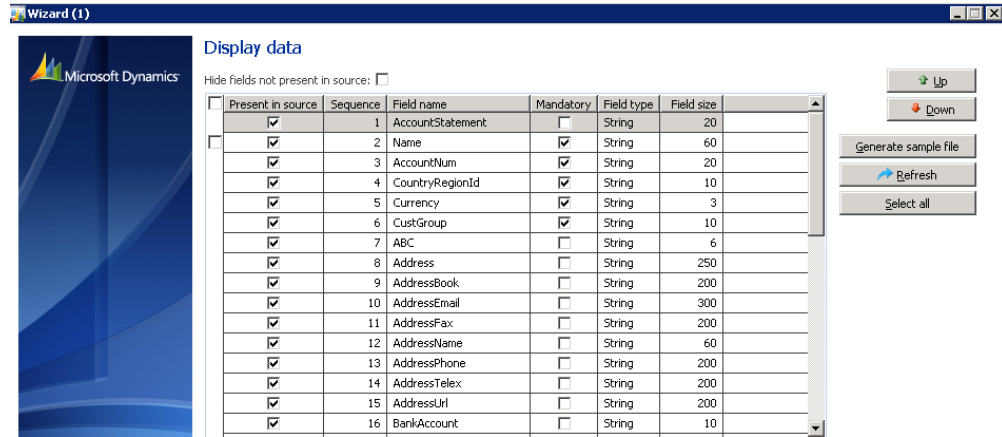


Additionally, the detailed view allows setting up simple value conversions that are applied during import; e.g. convert the country value of US to USA.

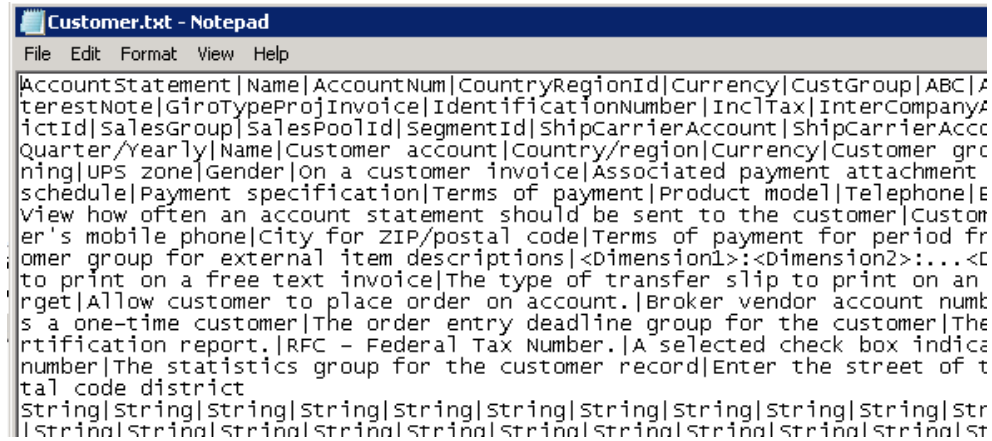


If there is no template or source file available, the "Generate source file" function will start a wizard to generate a basic template.

The Wizard guides the user through a few quick steps to determine which fields should be included in the template.



Once the fields are selected, the “Generate sample file” button will generate a text file that can be imported into Excel to serve as a template base:



	A	B	C	D	E	F	G	
1	AccountNum	CustGroup	Currency	CountryRegionId	LanguageId	Name	InvoiceAccount	Pay
2	Customer account number.	Customer group	Currency	Country	Language	Name	Invoice account	Ter
3	Customer account number.	Customer group that the customer is attached to.	The customer's company currency for invoicing	Primary address - Country.	Default language of vendor.	Reference field in a different table	The alternate customer account selected to receive	The payi cust
4	UserType	UserType	UserType	UserType	UserType	UserType	UserType	Use
5	Yes	Yes	Yes	No	No	No	No	No
6								
7								

Framework Process

The previous section described the required setup to process a source file into the target entities. The following section describes the actual processing steps:

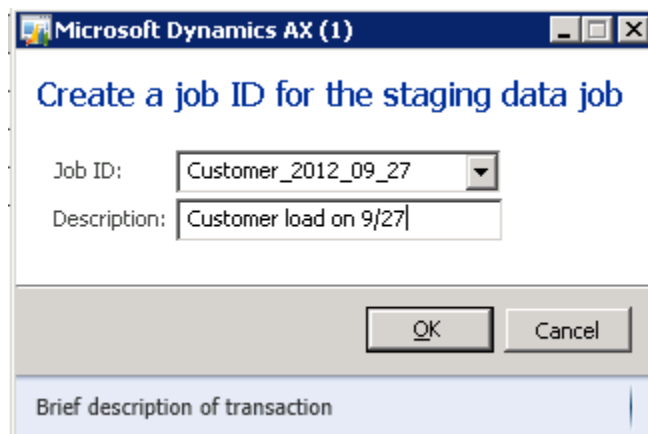
- **Load Source Data:**
Loads the source data file into the intermediate staging tables.
- **Staging Data validation:**
Validates the staging table records for referential integrity.
- **Process staging data into target entities:**
Executes the final processing for converting staging data into actual Dynamics AX table records.

Load Source Data

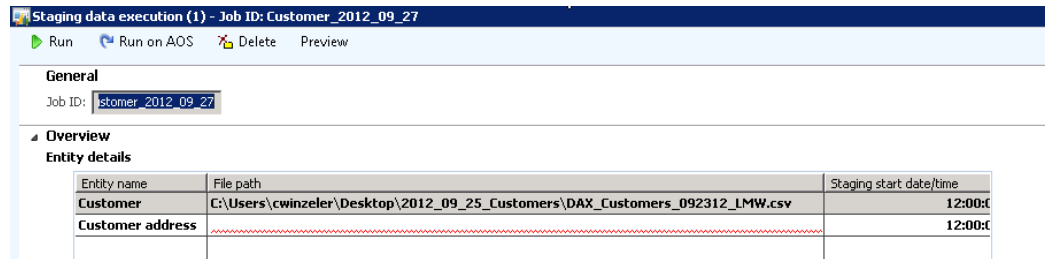
After the processing group setup is completed, the framework is ready to load the source data into the staging table(s).

It is possible to load data for the same entity in multiple iterations; an example would be to load US customers first, then Canadian. For this purpose every time a data file is loaded a Run ID is assigned. The Run ID is assigned to every record and allows for the described separation.

Through the “Get staging data” function the Run ID as well as the specific file used for the import can be selected.



The screenshot shows a dialog box titled "Microsoft Dynamics AX (1)" with the main heading "Create a job ID for the staging data job". It contains two input fields: "Job ID:" with a dropdown menu showing "Customer_2012_09_27" and "Description:" with a text box containing "Customer load on 9/27". At the bottom, there are "OK" and "Cancel" buttons. A footer bar at the very bottom contains the text "Brief description of transaction".

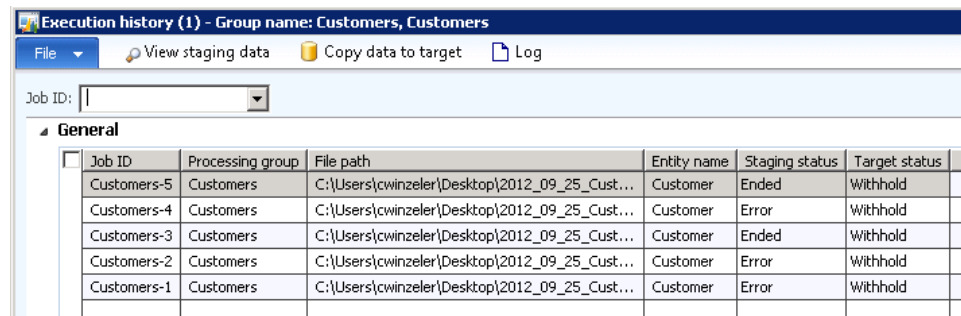


There are two options to process the file. The “Run” option will import the file immediately and use the client’s local security policy & resources. The “Run on AOS” allows adding the import job to the batch queue and execute it at a later point and time. It is important to mention is that the AOS has to have visibility to the file location as well as the AOS user has to have the proper permission to read from the location.

Staging Data Validation

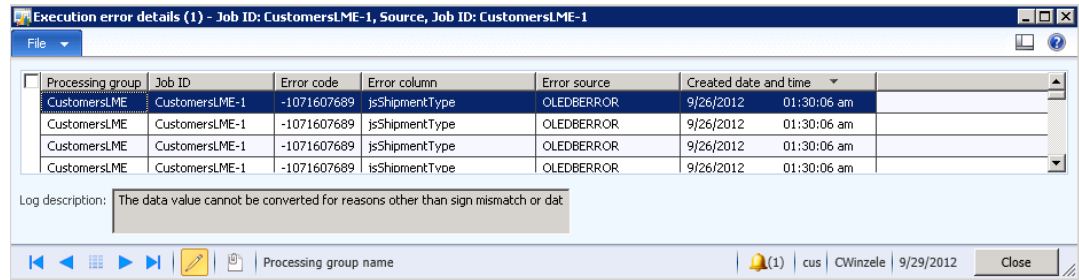
Every time a new job is executed, a new record in the execution history is created. The execution history can be reviewed from the processing group and will enable the following tasks:

- **View the log of the import:** Allows review of potential errors
- **View the staging data:** Allows validation of imported records
- **Copy the data to target:** Allows you to process the staged data to the target entity



If errors occurred during the import the staging status will show “Error” and through the log function the user can review any potential issues. However, there are certain errors that are not logged in the current version. Some of these include duplicate primary keys or an invalid file format. These errors have to be resolved in the source file before reloading the data.

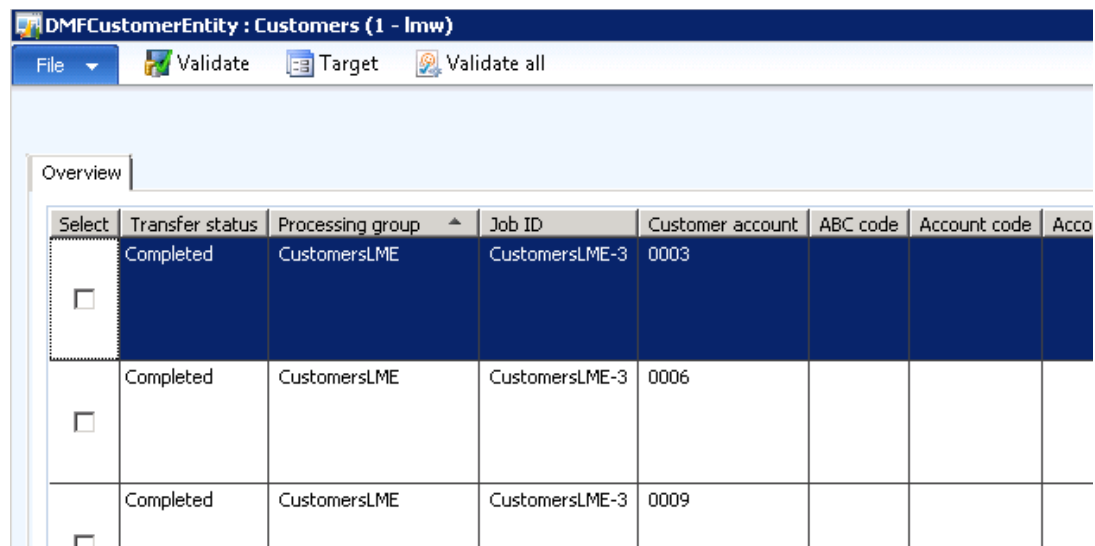
Example of an error that may occur: A Base Enum value that cannot be converted. This error is logged.



Processing group	Job ID	Error code	Error column	Error source	Created date and time
CustomersLME	CustomersLME-1	-1071607689	jsShipmentType	OLEDBERROR	9/26/2012 01:30:06 am
CustomersLME	CustomersLME-1	-1071607689	jsShipmentType	OLEDBERROR	9/26/2012 01:30:06 am
CustomersLME	CustomersLME-1	-1071607689	jsShipmentType	OLEDBERROR	9/26/2012 01:30:06 am
CustomersLME	CustomersLME-1	-1071607689	jsShipmentType	OLEDBERROR	9/26/2012 01:30:06 am

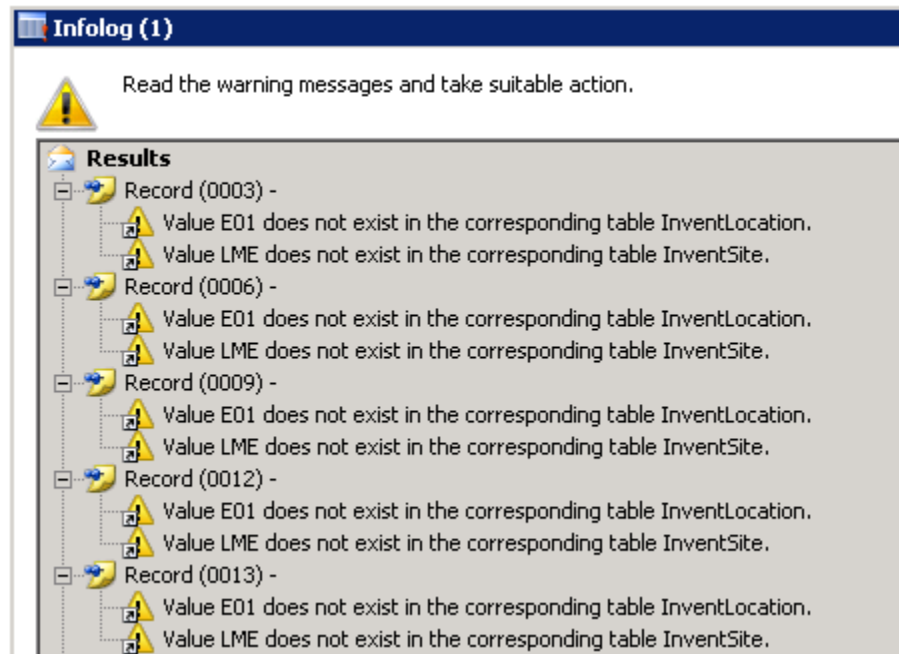
Log description: The data value cannot be converted for reasons other than sign mismatch or dat

Other than validating the format of the data, there are no referential integrity checks executed during the import. Reviewing and validating the actual data is an additional step and can be done through the “View staging data” function.



Select	Transfer status	Processing group	Job ID	Customer account	ABC code	Account code	Acco
<input type="checkbox"/>	Completed	CustomersLME	CustomersLME-3	0003			
<input type="checkbox"/>	Completed	CustomersLME	CustomersLME-3	0006			
<input type="checkbox"/>	Completed	CustomersLME	CustomersLME-3	0009			

The “Validate all” function will loop through all the records and validate each field that has a link to related table. The result is presented in an info log to the user.



There are three possibilities to correct errors.

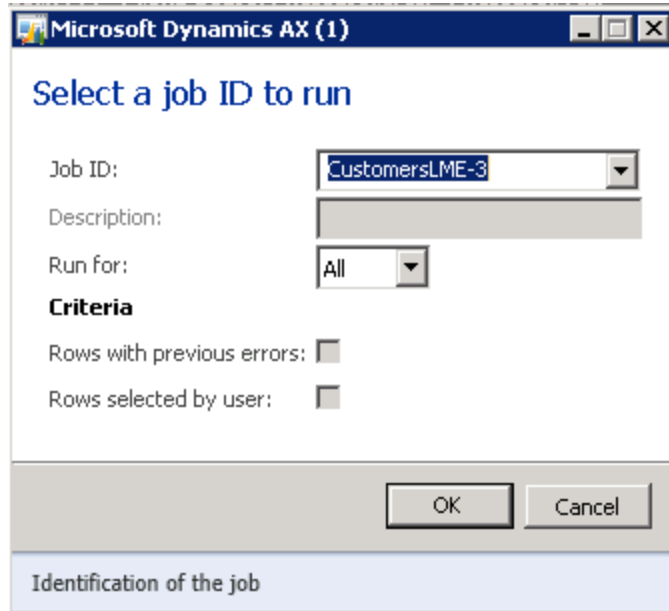
1. The data elements can be corrected in the source file. This is advisable if the file is going to be used to load the data into another environment or to load the data at a later stage again.
2. The data can be manually corrected on the “View staging data” file. This is helpful if there are only a few issues to correct and or the number of records is limited.
3. The staging data can be mass updated through custom AX jobs. As the staging tables are standard AX data tables they can be modified through code like any other entity.

It is important to notice that the framework will not prevent the user from processing staging data to target entities even though there are errors in the imported data. If the data is not carefully reviewed and validated the potential issue of referential integrity arises.

Process staging data to target entities

If there are no errors remaining the staging data can be processed to the target entities. The “Copy data to target” function will complete the process by reading the staging data and creating the actual AX records.

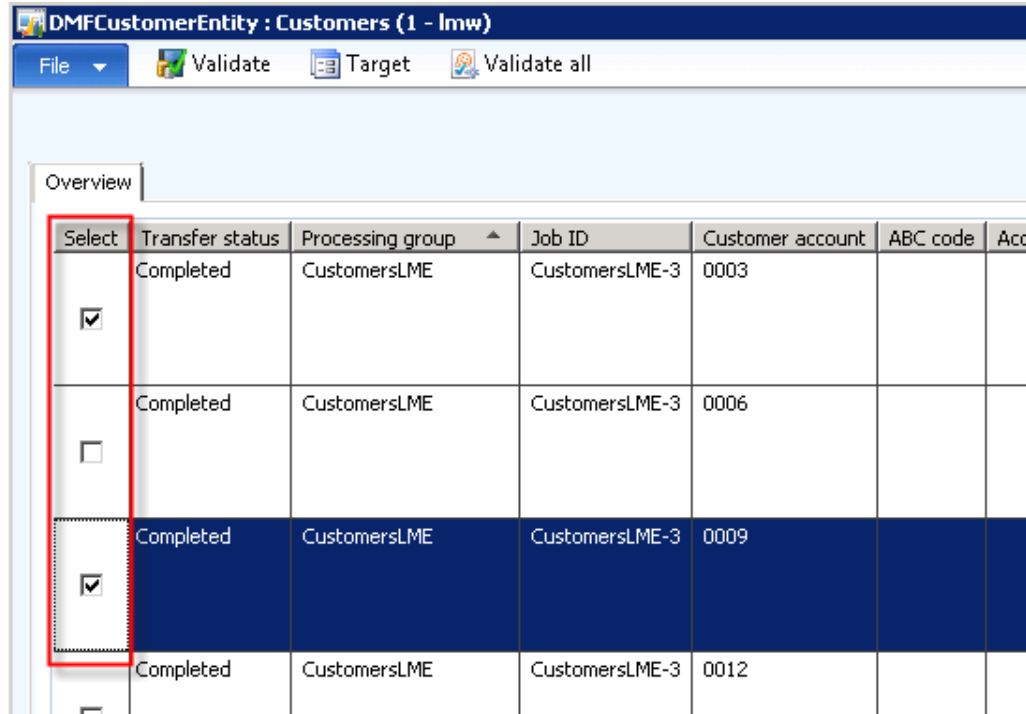
The staging data is processed based upon the assigned Job ID during the import. This allows processing the parts of the staging data that are ready for processing, but not the ones still worked on. For example all US customers are reviewed and ready; but Canadian customers are still being validated.



Additionally it is possible to only process a subset of a specified job by changing the “Run for” property to “Criteria”.

- **Rows with previous errors:** Only records that had errors during the import will be processed.
- **Rows selected by user:** Only records selected in the “View staging data” will be processed.

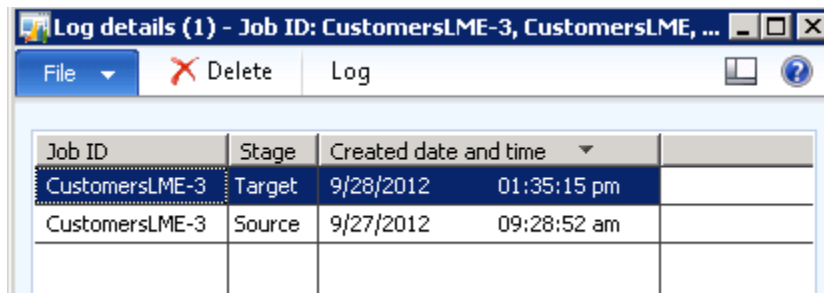
Leaving the property at “All” will process all staged records for the selected Job ID.



Select	Transfer status	Processing group	Job ID	Customer account	ABC code	Acc
<input checked="" type="checkbox"/>	Completed	CustomersLME	CustomersLME-3	0003		
<input type="checkbox"/>	Completed	CustomersLME	CustomersLME-3	0006		
<input checked="" type="checkbox"/>	Completed	CustomersLME	CustomersLME-3	0009		
<input type="checkbox"/>	Completed	CustomersLME	CustomersLME-3	0012		

A second form similar to the form used during the import of the staging data will be presented. The user has the ability to process the data immediately or send the function to the batch queue where it can be executed on the AOS at a later time.

After the function is complete, the processed records are available in AX. Any errors during the import are recorded in the execution history log and can be reviewed there.

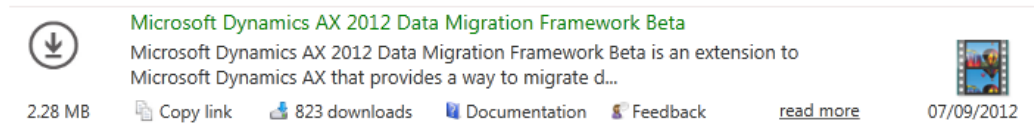


Job ID	Stage	Created date and time
CustomersLME-3	Target	9/28/2012 01:35:15 pm
CustomersLME-3	Source	9/27/2012 09:28:52 am

Installation

The Data Migration Framework (DMF) is available for download on Microsoft's Information source portal. The current framework is only available in a beta version; a release date for the final version has not yet been provided.

<http://informationsource.dynamics.com>



The screenshot shows a download card for the "Microsoft Dynamics AX 2012 Data Migration Framework Beta". It includes a download icon, the title, a brief description, and several action links: "Copy link", "823 downloads", "Documentation", "Feedback", and "read more". The file size is listed as "2.28 MB" and the date is "07/09/2012".

Once the executable is downloaded it has to be extracted into an appropriate folder. It is important to run the setup.exe with administrator privileges on the AOS server.

The installation procedure will:

- Create a folder that contains the AX model used to add the necessary elements to AX
- Install the required DLLs in the client and AOS bin folder
- Create various demo files for testing/review

If the DMF should be used from a different client, it is possible to copy the following DLLs into the client bin folder (C:\Program Files (x86)\Microsoft Dynamics AX\60\Client\Bin):

- Microsoft.Dynamics.AX.DMF.Mapper.dll
- Microsoft.Dynamics.AX.DMF.PreviewGrid.dll
- Microsoft.Dynamics.AX.DMF.SSISHelper.dll

To install the required objects in AX, the model created must be installed using AXUtil or PowerShell:

```
AXUtil: axutil import /file: "C:\Program Files\Microsoft Dynamics AX 2012 Data Migration Framework (Beta)\DataMigrationFramework.axmodel"  
PowerShell: Install-AXModel -File "C:\Program Files\Microsoft Dynamics AX 2012 Data Migration Framework (Beta)\DataMigrationFramework.axmodel"
```

After the AOS is restarted, the following tasks have to be executed:

- Synchronize the data base
- Compile the AX application
- Compile into .Net Framework CIL (full compile; not incremental)

It is noteworthy that the provided model is installed into the USR layer for easier separation. Once all data migration tasks have been completed the DMF model can be uninstalled to ensure all application objects as well as staging records are removed.