

## Правила та процедури імпорту даних (Data import templates)

Впровадження системи управлінського обліку на базі програмного  
забезпечення Microsoft Dynamics у Державному підприємстві «Національна  
енергетична компанія «УКРЕНЕРГО»

**ЗАТВЕРДЖЕНО:**

<b>Від ДП «НЕК «УКРЕНЕРГО»</b>  Керівник проектів та програм у сфері нематеріального виробництва департаменту з управління ефективністю (Довіреність №16 від 13.02.2018)  _____ Кравченко Ю.В.	<b>Від ТОВ «СМАРТ БІЗНЕС»</b>  Директор    _____ Руднєв К.М.
---	--

<b>1. ЗАГАЛЬНЕ.....</b>	<b>3</b>
<b>2. ПРОЦЕДУРА ІМПОРТУ ДАНИХ .....</b>	<b>3</b>
<b>3. ОРГАНІЗАЦІЙНІ РОБОТИ З МІГРАЦІЇ.....</b>	<b>4</b>
<b>4. ІНСТРУКЦІЯ З РОБОТИ В МОДУЛІ ІМПОРТУ/ЕКСПОРТУ ДАНИХ .....</b>	<b>5</b>
4.1. Модуль.....	5
4.2. КОНФІГУРАЦІЯ.....	6
4.2.1. <i>Визначення сутності - DMF &gt; Setup &gt; Target entities .....</i>	<i>6</i>
4.2.2. <i>Визначення формату вихідних даних – DMF &gt; Setup &gt; Source data formats.....</i>	<i>6</i>
4.2.3. <i>Визначення формату даних для кожної сутності в інструментальному середовищі .....</i>	<i>7</i>
4.2.4. <i>Визначення формату даних для кожної сутності в інструментальному середовищі .....</i>	<i>8</i>
4.2.5. <i>Визначення групи обробки (processing group).....</i>	<i>8</i>
4.2.6. <i>Налаштування сутності для конфігурації.....</i>	<i>8</i>
4.2.7. <i>Перенесення даних з проміжної таблиці до кінцевого місця призначення ах.....</i>	<i>9</i>
4.2.8. <i>Створення співставлення налаштовуваних сутностей із застосуванням DMF.....</i>	<i>10</i>
4.2.9. <i>Створення із застосуванням “Create a custom entity for migration” .....</i>	<i>10</i>
4.2.10. <i>Створення вручну.....</i>	<i>12</i>

## 1. Загальне

В Microsoft Dynamics AX 2012 налаштовано список обумовлених об'єктів, які можуть імпортуватися в систему із зовнішніх джерел таких як:

- файли-таблиці різного формату;
- текстовий файл;
- база даних;

Кожен об'єкт імпорту\експорту асоціюється з обумовленим файлом-джерелом, класом та функціями його обробки, проміжною таблицею та цільовою таблицею.

Проміжна таблиця являє собою набір даних в табличному вигляді, в яку записуються данні із файлу-джерела. Список полів в проміжній таблиці чітко визначаються при умові використання процедур імпорту\експорту не може бути розширений. Назви полів теж суворо визначенні оскільки є класом об'єкта імпорту\експорту.

Для кожного об'єкта імпорту\експорту створено свій клас та функції обробки, за допомогою яких, данні з проміжної таблиці записуються в базу даних системи :

- В одну цільову таблицю;
- В декілька логічно пов'язаних між собою таблиць.

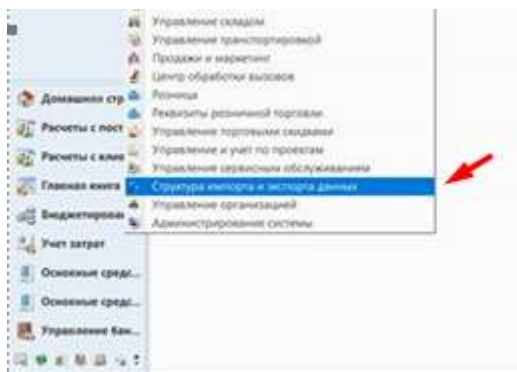
В якості цільових об'єктів, можуть виступати:

- Довідники системи;
- Журнали;

## 2. Визначення

Система - система Microsoft Dynamics AX 2012

Модуль імпорту/експорту даних системи – модуль системи, який призначений для імпорту/експорту даних Системи із сторонніх джерел (вхідний формат даних). Викликається за допомогою меню Системи:



## 3. Процедура імпорту даних

### 3.1. Процедура імпорту даних складається із таких етапів:

- Визначення відповідностей до вхідного формату даних (команда Замовника);
- Визначення вхідного формату даних (команда Виконавця);
- Підготовка структури шаблону даних (команда Виконавця);
- Визначення групи обробки (команда Виконавця);
- Підготовка даних для завантаження відповідно до шаблону даних, перевірок даних для завантаження (команда Замовника);
- Розробка, створення процедури завантаження (команда Виконавця);
- Імпорт даних з файлу-джерела в Цільову таблицю Microsoft Dynamics AX 2012 (команда Замовника та команда Виконавця, яка надає консультації щодо виправлення помилок, налаштовує роботу модуля імпорту/експорту даних системи.);
- Періодично. Очищення проміжних даних (команда Замовника);

### 3.2. Визначення відповідностей до вхідного формату даних.

Надання у форматі Excel, структури бізнес-даних, яка має бути завантажена в систему.

### 3.3. Визначення вхідного формату даних

Файл-джерело може мати наступний тип:

- Файл
- Текстовий з роздільниками
- Текстовий з фіксованою шириною полів
- XML
- Excel.
- Microsoft Dynamics AX 2012
- ODBC конектор.

### 3.4. Підготовка структури шаблону даних

Підготовка структури шаблону даних виконується командою виконавця та замовника у модулі імпорту/експорту даних системи. В шаблоні створюються системні назви колонок із коментарями роз'яснення.

### 3.5. Визначення групи обробки

Об'єкти, які входять в одну групу обробки, обробляються разом. В одну групу необхідно включати наступні типи об'єктів:

- Пов'язанні об'єкти, які повинні бути оброблені разом;
- Об'єкти, які Користувач об'єднав у один пакет. В такому разі необхідно виставити послідовність обробки об'єктів;

Прикладами об'єктів, які можна об'єднати в одну групу обробки, є:

- Клієнти (послідовність = 1);
- Додаткові адреси (послідовність = 2)

В той же час, для кожного з вказаних довідників може бути створена своя група обробки.

### 3.6. Підготовка даних для завантаження відповідно до шаблону даних

Підготовка даних для завантаження відповідно до шаблону даних виконується командою Замовника. Якщо кількість строк записів в шаблоні менше 50, завантаження даних виконується вручну в формі користувача системи.

### 3.7. Розробка, створення процедури завантаження та перевірок даних для завантаження

Для завантаження даних використовуються стандартні об'єкти Модулю імпорту/експорту даних системи з автоматичним перевірками коректності даних. Якщо при завантаженні система генерує помилки, команда Замовника виконує потрібні зміни в даних, та повторює процедуру завантаження. Команда Виконавця всебічно надає консультації щодо виправлення помилок, налаштовує роботу модуля імпорту/експорту даних системи.

### 3.8. Імпорт даних з файлу-джерела в Цільову таблицю Microsoft Dynamics AX 2012

Імпорт складається з таких кроків в модулі імпорту/експорту даних системи:

- Завантаження даних з файлу-джерела в проміжну таблицю;
- Перегляд, перевірка та коригування даних в проміжній таблиці;
- Завантаження даних із проміжної таблиці в цільову.

### 3.9. Очищення проміжних даних

Очищення проміжних даних виконується після завершення завантаження кожного об'єкту. Для цього використовується стандартна процедура – «Очистка промежуточных данных»

## 4. Організаційні роботи з міграції

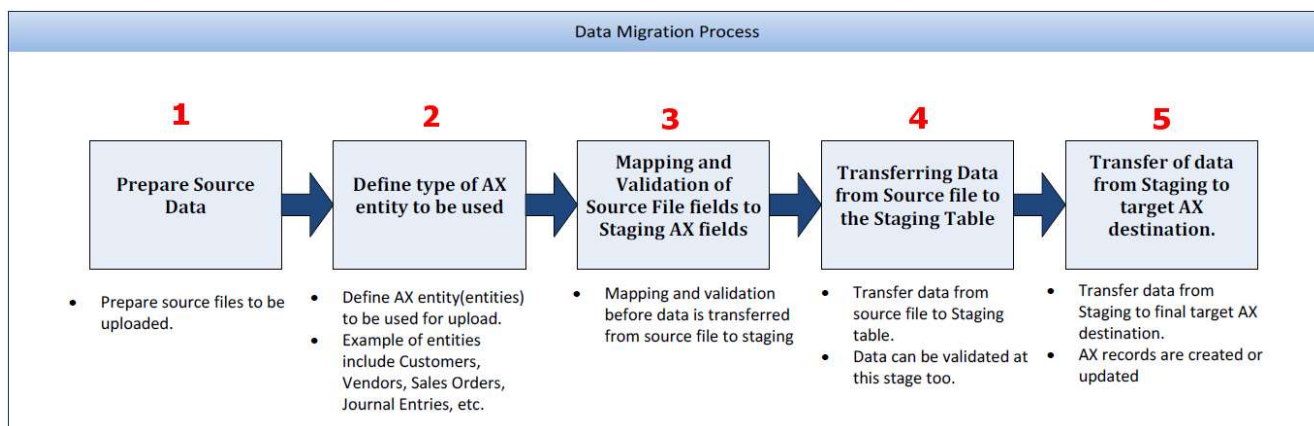
Організаційні роботи з імпорту даних складаються із таких вимог:

- Визначення складу робочої групи з міграції. Робоча група створюється із представників команди Замовника та Виконавця, знайомі в достатній мірі з роботою історичних систем;
- Попередній план міграції. План міграції по ходу проекту буде неодноразово коригуватися. Корегування відбувається за допомогою запиту на зміну який відображається в календарному план-графіку проекту;
- Періоди дат вивантаження даних з історичних систем, обсяги даних. Періоди зрізу даних для міграцій, дати тестових і підсумкової міграцій. Дану інформацію можна віднести до плану міграції;
- Зміст даних, що підлягають міграції. Довідкові дані, класифікатори, трансакційні дані, залишки, обороти та ін.;
- Перевірка якості, коректності та цілісності даних в процесі міграції і за підсумками. Система автоматично перевіряє цілісність даних при завантаженні. Додаткову перевірку якості та коректності даних виконує група Замовника за долученням групи Виконавця для надання консультацій та налаштування роботи системи;

## 5. Інструкція по роботі модуля імпорту/експорту даних Системи

### 5.1. Модуль

Назва	Визначення
Staging table Проміжна таблиця	тимчасове місце та перехідна локація до переносу даних в остаточне місце призначення в AX
Target table Цільова таблиця	таблиця-отримувач



➤ Підготовка вихідних даних (Prepare Source Data): Файл повинен мати заголовок, що включає назви полів, які позначають кожен стовпчик даних. Якщо у якості заголовка використовувати назви стовпців AX, то DMF намагатиметься зробити автоматичне співставлення. Якщо використовувати різні назви, то потрібно буде співставляти файл і перехідну таблицю вручну.

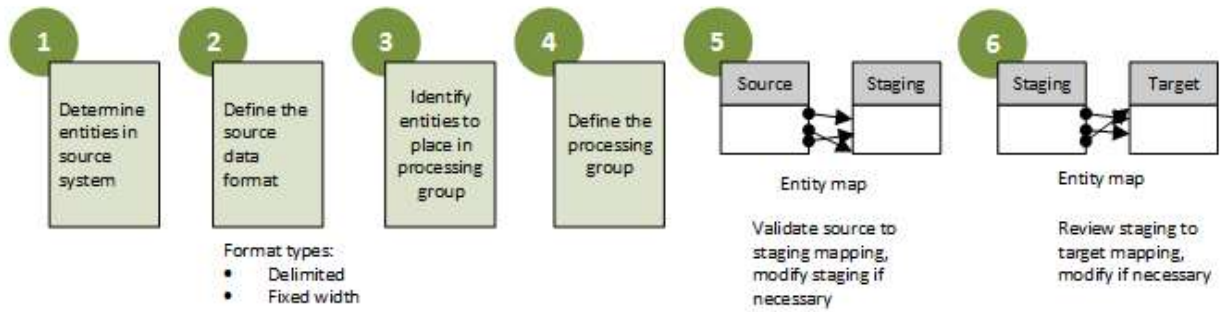
➤ Визначення типу сутності AX, що буде використовуватись (Define type of AX entity to be used). Визначте сутності, які будуть завантажуватись в AX. Наприклад: клієнти, вендори...

➤ Співставлення і валідація полів файла-джерела з проміжними полями AX (Mapping and validation of source file fields to staging AX fields). Інструмент DMF дозволяє співставляти кожне поле з файла-джерела з полем AX у проміжній таблиці. DMF використовує назви полів для співставлення полів у проміжній таблиці.

➤ Перенесення даних з файла-джерела до проміжної таблиці (Transferring data from source file to the staging table). Дані будуть готові для перенесення в проміжну таблицю одразу після співставлення полів файла-джерела з полями AX у проміжній таблиці.

➤ Перенесення даних з проміжної до цільової таблиці AX (Transfer of data from staging to target AX destination). Одразу після перегляду та валідації у проміжній таблиці дані переносяться до цільової сутності AX, і створюються записи AX.

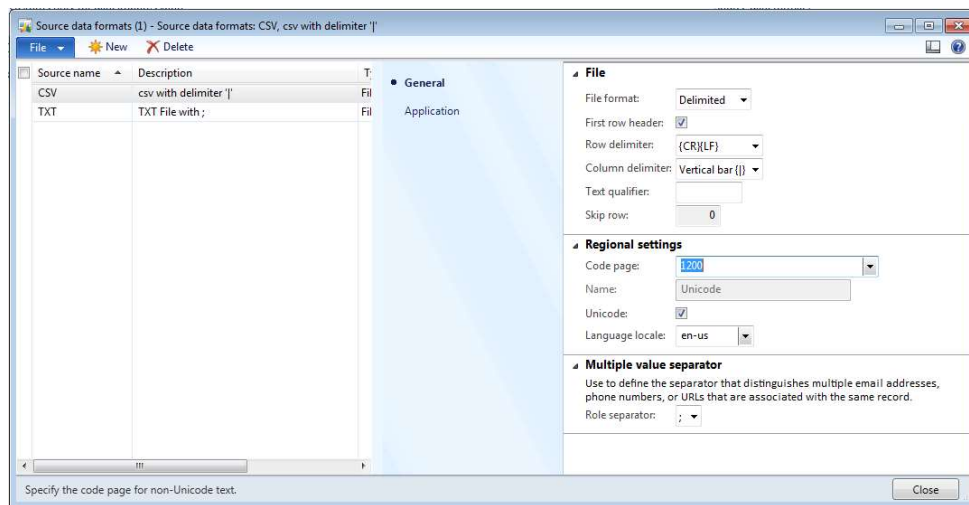
## 5.2. Конфігурація



### 5.2.1. Визначення сутності - DMF > Setup > Target entities

Існує набір визначених наперед сутностей. Ці сутності також включають у себе процедури для переміщення даних з проміжної таблиці до цільової.

### 5.2.2. Визначення формату вихідних даних – DMF > Setup > Source data formats



<b>File format / Формат файлу</b>	Оберіть фіксовану ширину (Fixed) або з роздільниками (Delimited). Формат з роздільниками використовує спеціальні символи для розділення полів. Формат з фіксованою шириною виділяє конкретну кількість символів на кожне поле.	
<b>First row header / Заголовок першого стовп.</b>	Оберіть цей варіант, якщо перший стовпчик ваших файлів з даними містить інформацію заголовку. При форматі з фіксованою шириною можна вказувати символи-роздільники для встановлення значень в рядку заголовка, якщо він є.	
<b>Row delimiter / Роздільник рядка</b>	Оберіть роздільник для рядків	
	{CR}{LF}	Рядок заголовку розділяється комбінацією символів повернення каретки/переведення рядка (carriage return/line feed).
	{CR}	Рядок заголовку розділяється символом повернення каретки.
	{LF}	Рядок заголовку розділяється символом переведення рядка.
	Semicolon {;}	Рядок заголовку розділяється крапкою з комою.
	Colon {:}	Рядок заголовку розділяється двокрапкою.
	Comma {,}	Рядок заголовку розділяється комою.
	Tab {t}	Рядок заголовку розділяється вкладкою (tab).
	Vertical bar { }	Рядок заголовку розділяється вертикальною рискою (vertical bar).
<b>Column delimiter / Розд-ник стовпця</b>	Оберіть роздільник для стовпців. Якщо вам потрібен список доступних значень – дивіться опис параметра <b>Роздільник рядка (Row delimiter)</b>	

<b>Text qualifier/ Кваліфікатор тексту</b>	Введіть роздільник, що використовується, якщо у значенні тексту в рядку є символ, який ви використовуєте як роздільник рядка чи стовпчика. Наприклад, якщо в назві вашої організації є коми, а ви їх використовуєте як роздільники стовпців, то вам потрібно ввести інший роздільник, який вживатиметься для текстових значень. У цьому прикладі краще використовувати символ лапок ("").
<b>Skip row/ Пропустити рядок</b>	Оберіть цю опцію, щоб пропускати рядки з неопрацьованими помилками і продовжити обробку. При виборі даної опції помилки записуються в журнал
<b>Code page</b>	Вкажіть кодову сторінку для тексту, який не відповідає стандарту Unicode.
<b>Unicode</b>	Оберіть цю опцію, щоб використовувати стандарт Unicode.
<b>Language locale</b>	Вкажіть регіон для надання мовної інформації при впорядкуванні, а також для встановлення форматів дати та часу.
<b>Multiple value separator</b>	На вибір: введіть роздільник, який використовуватиметься для полів з багатьма значеннями, що пов'язані з одним записом, як то електронні адреси, номери телефонів чи URL.
<b>Dimension code</b>	На вибір: вкажіть список фінансової аналітики, яка є складовою файла-джерела. Коди для фінансової аналітики беруться з Microsoft Dynamics AX.
<b>Chart of account: delimiter</b>	Роздільник між значеннями фінансової аналітики.
<b>Dimension format</b>	Встановлений наперед формат виміру, що ґрунтується на фінансовій аналітиці та вказаному вами роздільнику.
<b>Name sequence</b>	Вкажіть послідовність імен за замовчуванням для сторін, які створюються під час міграції.

### 5.2.3. *Визначення формату даних для кожної сутності в інструментальному середовищі*

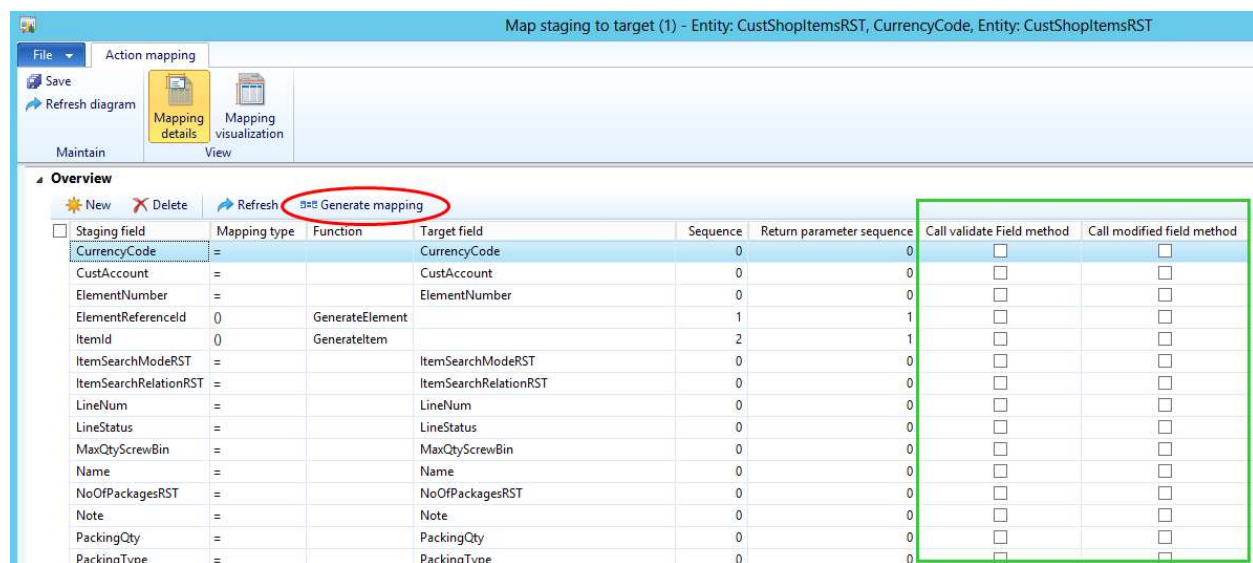
Сутності використовуються для того, щоб вказати, яку саме сутність DMF можна використати (наприклад: вендори, клієнти...) для обробки, порядок, в якому оброблятимуться сутності, якщо вибрано більше, ніж одну, джерело даних для використання, файл-джерело для використання при обробці, а також чи буде враховуватись бізнес-логіка AX при завантаженні. Користувач також використовуватиме цю сутність для співставлення і валідації вихідних полів з полями проміжної таблиці AX.

Ви можете створити три типи сутностей:

- **Entity (сутність):** співставляє окрему сутність.
- **Table (таблиця):** дозволяє імпортувати дані з джерела прямо до цільової таблиці AX без використання перехідної таблиці чи застосування будь-якої бізнес-логіки.
- **Composite entity (складна сутність):** дозволяє створити одну сутність, яка складається з кількох пов'язаних сутностей. Складна сутність використовується при імпорті з джерела, що містить дані з різних пов'язаних таблиць в одному файлі.

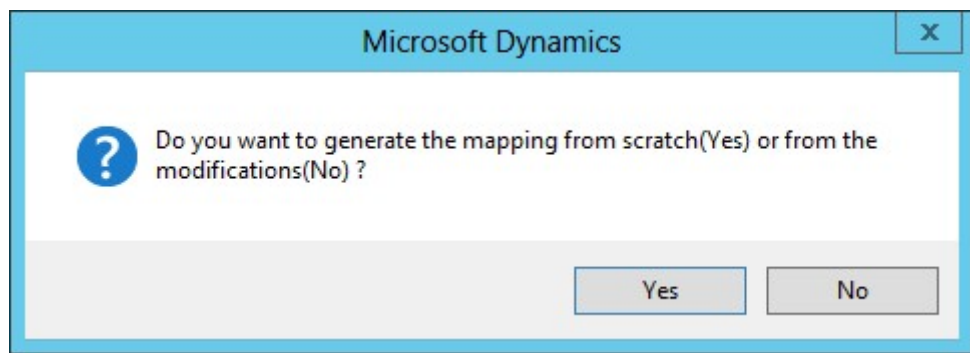
### Внесення змін в цільове співставлення сутності

1.5 Оберіть цільову сутність, яка буде змінюватись, і натисніть **Modify target mapping**.



2. Натисніть **Generate mapping** для повторної генерації полів та співставлення





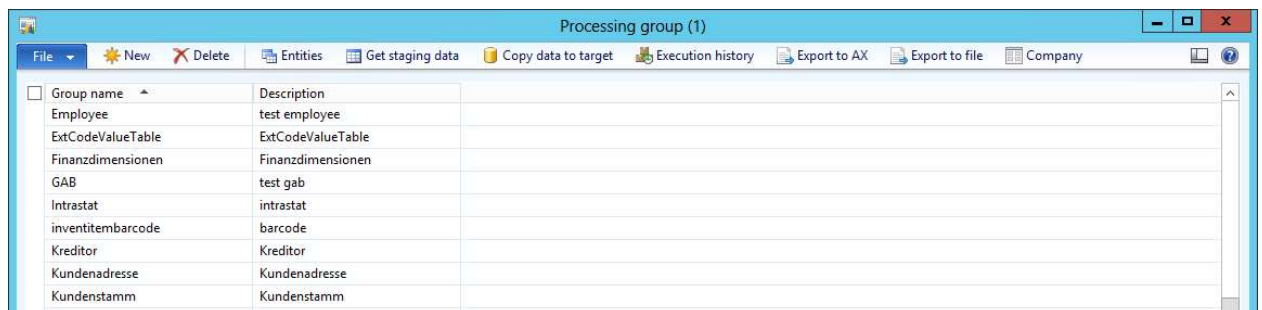
#### 5.2.4. *Визначення формату даних для кожної сутності в інструментальному середовищі*

Форматом даних може бути файл з фіксованою шириною чи з роздільниками, вихідні дані ODBC або таблиця Microsoft Dynamics AX.

#### 5.2.5. *Визначення групи обробки (processing group)*

##### **Data import export framework > Common > Processing group**

Групи обробки використовуються в DMF у якості основного драйвера, який дозволяє користувачам налаштовувати сутності, переносити дані до проміжної таблиці, робити валідацію та копіювання даних з проміжної таблиці до таблиці-отримувача AX. Такі групи обробки можуть використовуватись для повторного імпорту даних та для ведення журналу історії транзакцій та помилок.

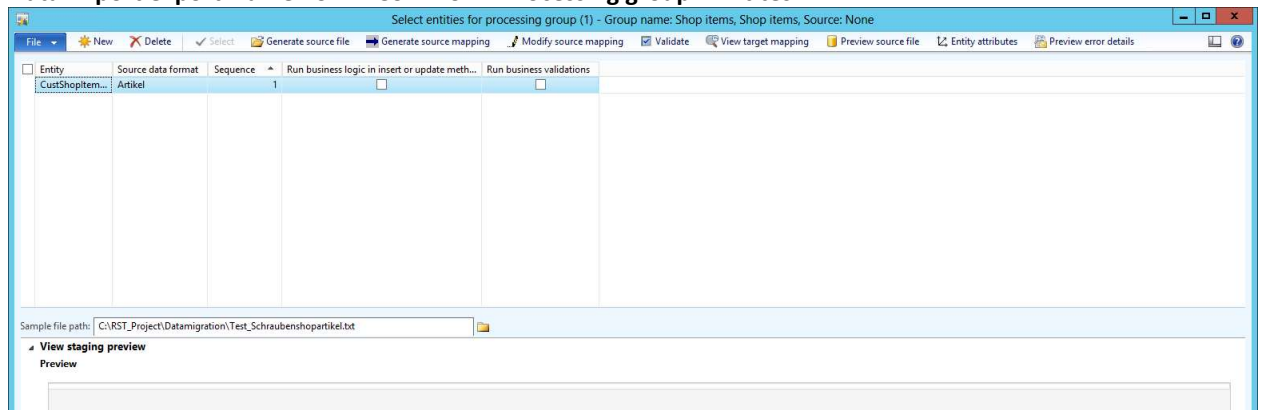


✓ Створіть нову групу обробки

- **Entities (сутності):** для визначення сутностей, які оброблятимуться
- **Get staging data (отримати дані в перехідну таблицю):** дані готові для копіювання до перехідної таблиці. Ця кнопка використовується для того, щоб почати процес. В повідомленні вказується, що дані скопійовано до перехідної таблиці.
- **Copy data to target (скопіювати дані в цільову таблицю):** копіює дані з перехідної таблиці до цільової
- **Execution history (історія виконання):** показує історію виконання
- Експортувати в AX
- Експортувати у файл
- Компанія

#### 5.2.6. *Налаштування сутності для конфігурації*

##### **Data import export framework > Common > Processing group > Entities**



✓ Створіть нову сутність

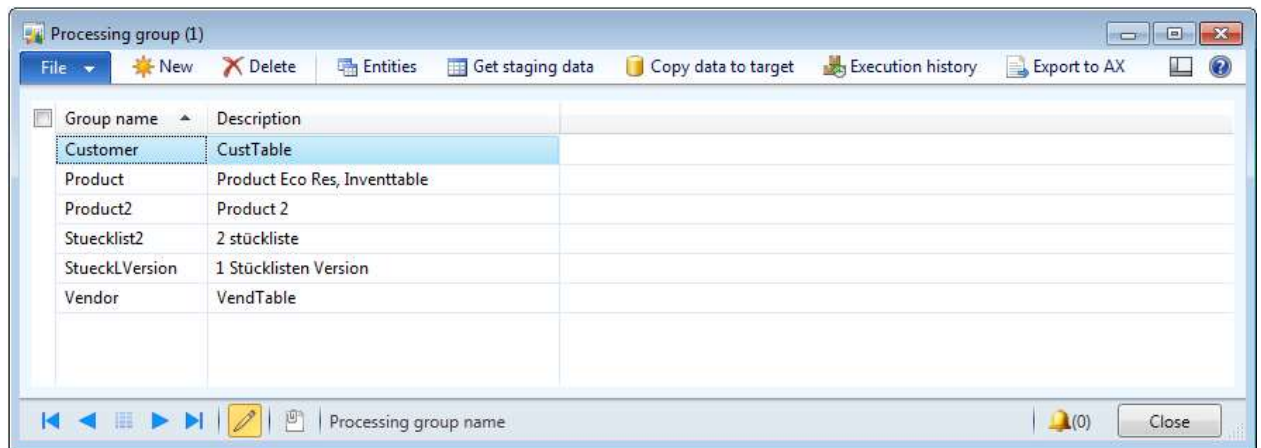
- **Entity:** оберіть сутність, яка використовуватиметься



- **Source data format (формат вихідних даних):** цей формат вихідних даних конфігурується в налаштуваннях і вказує тип вихідних даних, що буде завантажено
- **Run business logic (запустити бізнес-логіку):** два прапорці для запуску перевірки бізнес-логіки AX
- **Sample file path (шлях до файлу-прикладу):** шлях до файлу
- ✓ **Generate source mapping:** співставляє файл-джерело з перехідною таблицею. Якщо у якості заголовку використовувати назви стовпців AX, DMF намагатиметься робити співставлення автоматично. Якщо використовувати різні назви, то прийдеться співставляти файл з перехідною таблицею вручну.
- ✓ **Modify source mapping:** для перегляду і внесення змін вручну в співставлення з джерела до проміжної таблиці.
- ✓ **Validate:** процес валідації забезпечує коректне співставлення всіх полів з файла-джерела з проміжною таблицею.
- ✓ **View target mapping:** ця опція може використовуватись для валідації співставлення з проміжної до кінцевої таблиці-отримувача AX.
- ✓ **Preview source file:** користувач може попередньо переглянути файл-джерело до того, як він буде скопійований до проміжної таблиці

#### 5.2.7. *Перенесення даних з проміжної таблиці до кінцевого місця призначення ax*

Виберіть "Copy data to target" (скопювати дані до цільової таблиці).



### 5.2.8. Створення співставлення налаштовуваних сутностей із застосуванням DMF

Коли ви створюєте налаштовувану сутність, ви повинні створити проміжну таблицю, проект, запит, клас та функції. Ви повинні завжди створювати відповідний клас та функції вручну, але можна використовувати один з двох методів для створення проміжної таблиці, проекту, та запиту:

- 1) Використовуйте Create a custom entity для майстра імпорту/експорту даних.
- 2) Створіть проміжну таблицю та запит вручну.

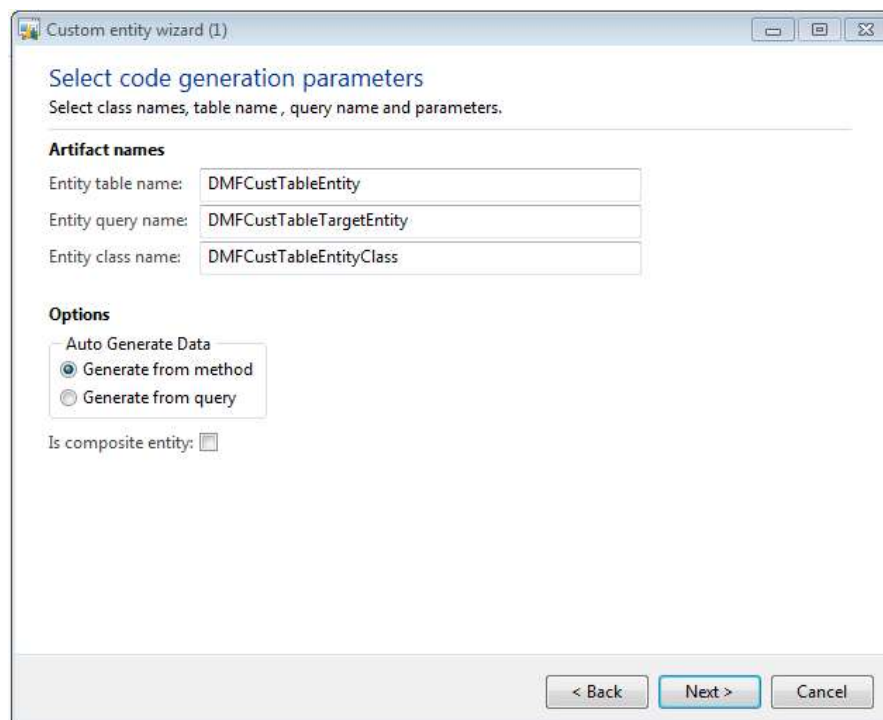
Назва	Визначення
<b>Query</b>	запит – це визначення цільової таблиці. Якщо потрібно для співставлення – можна додавати зв'язки
<b>Table</b>	Таблиця з доданим суфіксом “Entity” – це проміжна таблиця, яка використовується для співставлення
<b>EDT</b>	зовнішній ключ, який використовує RecId. Згенерований EDT – це строкове поле з кодом
<b>Class</b>	клас - це там, де можна закласти бізнес-логіку. Наприклад, потрібно вирішити EDT з RecId

### 5.2.9. Створення із застосуванням “Create a custom entity for migration”

- 1) Data Import/Export Framework > Common > Create a custom entity for data import/export [Data migration framework > Common > Create a custom entity for migration] (2012)



- 2) Оберіть таблицю, пов'язану з сутністю, яку ви бажаєте перемістити, і натисніть Next .
- 3) У вікні “Select code generation parameters” майстер налаштувань запропонує назви для проміжної таблиці, запиту та класу.



4) На сторінці “Wizard complete” натисніть “Finish”. Модуль Data Import/Export Framework відкриє Application Object Template (AOT) та створить проект для відповідного елементу.

5) Якщо таблиця, яку Ви обрали, використовує розширений тип даних (EDT), система запитає чи бажаєте ви додати зв'язок ForeignKey з EDT до нової проміжної таблиці. Для створення взаємозв'язку з ForeignKey натисніть Yes.

### 5.2.10. Створення вручну

#### Створення проміжної таблиці (Staging Table)

1) У АОТ створіть проміжну таблицю для відповідної сутності з наступними властивостями таблиці:

Властивість	Значення
SaveDataPerCompany	NO
SupportInheritance	NO
TableType	Regular
ConfigurationKey	DMF
ValidTimeStateFieldType	None

2) Створіть поля з наступними властивостями:

Назва поля	EDT	Enum
DefinitionGroup	DMFDefinitionGroupName	
IsSelected	DMFIsSelected	NoYes
TransferStatus		DMFTransferStatus
ExecutionId	DMFExecutionId	



3) Створіть групи полів з наступними властивостями:

Назва групи полів	Мітка	Поле
ExecutionList	Execution list	DefinitionGroup, IsSelected, TransferStatus, ExecutionId
Enabled	Enabled	OPTIONAL. Fields from the staging table which you want to make part of template by default ВИБІРКОВО. Поля з проміжної таблиці, які Ви бажаєте зробити частиною шаблону по замовчуванню.
<<FunctionName_Sequence>> Ex.: GeneratePostalAddress_2	<<Description of function>>	Fields from staging table which are the source for the specified method (fields that are used in the method) Поля з проміжної таблиці, які є джерелом для певного методу (поля які використовуються для такого методу)

**ПРИМІТКА:** Група полів “Generate” повинна бути створена, навіть якщо в ній не буде полів. Якщо у певному класі є метод “Generate”, таблиця DMF повинна мати прив’язану групу полів (незалежно від того пуста вона, чи ні). Наприклад: таблиця “DMFProductEntity” має групу полів з назвою “GenerateItemGroupCompany\_2”, яка є пустою, а відповідний клас “DMFProductEntityClass” має метод з назвою “GenerateItemGroupCompany”.

4) Створіть первинний індекс для таблиці:

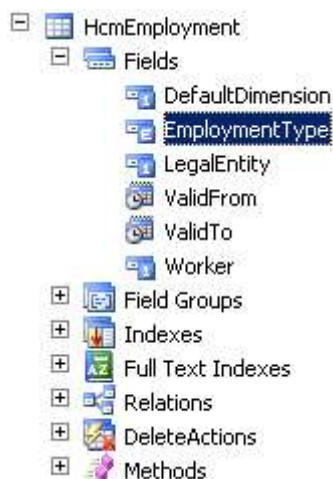
Назва індексу	Властивості	Поле (Поля)
<<any name>> Ex.: Idx	AllowDuplicates (HI)  AlternateKey (TAK)	DefinitionGroup, ExecutionId, поля з проміжної таблиці для визначення унікальності  DefinitionGroup, ExecutionId, поля з проміжної

- 5) Визначте взаємозв'язки між проміжною таблицею та цільовою таблицею

#### Створення полів, які містять перелічення (Enum fields), в цільовій сутності

Поле, яке містить перелічення у цільовому елементі, представлене у рядку проміжної таблиці. Ви маєте створити новий тип розширених даних для типу рядка відповідної довжини, що має вміщувати перелічення у відповідному рядку.

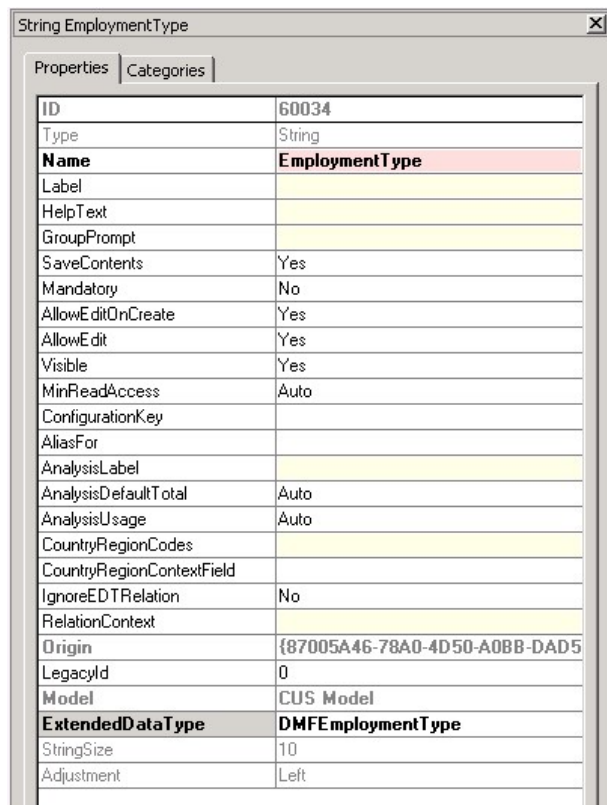
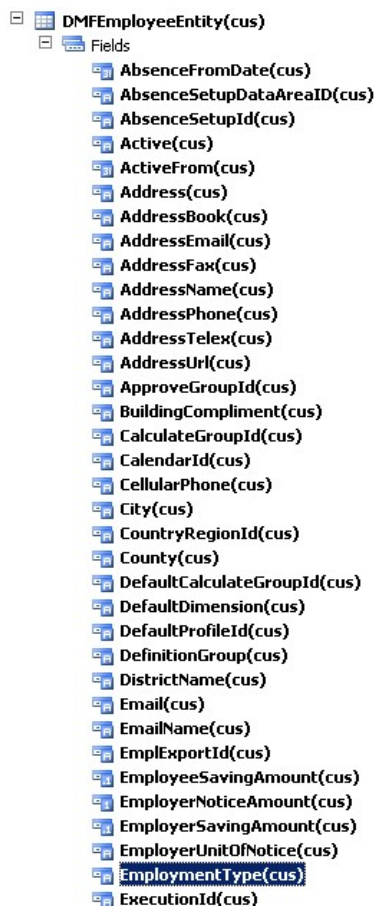
Наприклад:



Enum EmploymentType	
Properties   Categories	
ID	6
Type	Enum
<b>Name</b>	<b>EmploymentType</b>
Label	
HelpText	
GroupPrompt	
SaveContents	Yes
<b>Mandatory</b>	<b>Yes</b>
AllowEditOnCreate	Yes
<b>AllowEdit</b>	<b>No</b>
Visible	Yes
MinReadAccess	Auto
ConfigurationKey	
AliasFor	
AnalysisLabel	
AnalysisDefaultTotal	Auto
AnalysisUsage	Auto
CountryRegionCodes	
CountryRegionContextField	
IgnoreEDTRelation	No
RelationContext	
Origin	{09A26A59-8A77-42C4-B109-1664D}
<b>LegacyId</b>	<b>6</b>
Model	Foundation
ExtendedDataType	
<b>EnumType</b>	<b>HcmEmploymentType</b>

HcmEmployment.EmploymentType  
DMFEmployeeEntity. EmploymentType

= Enum HcmEmploymentType  
= Type String → EDT = DMFEmploymentType



#### RefRecId

Якщо цільова таблиця містить поля, які є RecIds з інших таблиць, ви повинні конвертувати вихідний ключ у RecId. Для цього є два варіанти:

- ✓ Додайте джерело даних так, аби до запиту цільової сутності можна було додати таблицю, на яку посилається ключ
- ✓ Створіть функцію

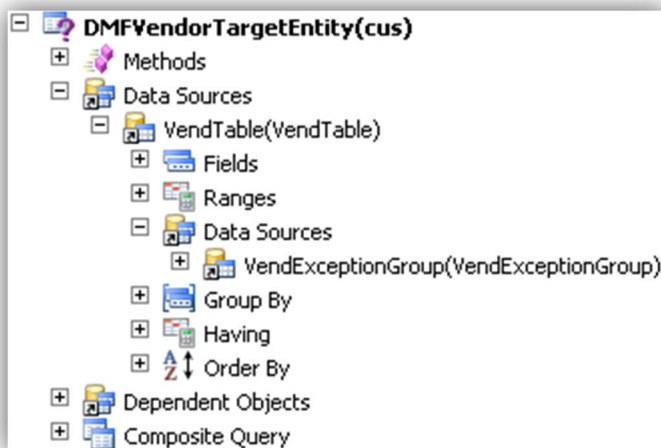
#### 5.2.10.1. Додавання джерела даних

Ви можете додати джерело даних, аби до запиту цільової сутності можна було додати таблицю, на яку посилається ключ.

Наприклад: цільова таблиця VendTable містить VendExceptionGroup, яка є RecId, що походить з таблиці VendExceptionGroup.



В такому випадку проміжна таблиця повинна включати VendExceptionGroup. Ми додаємо таблицю до запиту як VendTable datasource. Такий зв'язок повинен вказуватись вручну.



Проміжне поле DMFVendorEntity.VendExceptionGroup потрібно співставити з запитом цільового поля для DMFVendorTargetEntity. DS:VendExceptionGroup.VendExceptionGroup

➔ **Внесіть зміни до запиту!!! Такий запит повинен включати нові поля!**

#### 5.2.10.2. Створення функції

Створіть метод в класі сутності з метою конвертації рядка у RecId. Наприклад, метод DMFCustomerEntity.GenerateCompanyIdNAF:

```
public container GenerateCompanyIdNAF(boolean _stagingToTarget = true)
{
    CompanyNAFCode companyNAFCode;
    container res;

    if (_stagingToTarget)
    {
        select firstOnly1 RecId from companyNAFCode where companyNAFCode.CompanyIdNAF == entity.CompanyIdNAF;
        res = [companyNAFCode.RecId];
    }
    else
    {
        if (target.CompanyNAFCode)
        {
            select firstOnly1 CompanyIdNAF, RecId from companyNAFCode where companyNAFCode.RecId == target.CompanyNAFCode;
        }

        res = [companyNAFCode.CompanyIdNAF];
    }
    return res;
}
```

Коли ви створюєте метод функції, ви повинні створити групу полів для проміжної таблиці, а поля повернення даних у цільовому об'єкті потрібно вказати в методі **getReturnFields** класу сутності.

```
case methodStr(DMFCustomerEntityClass, GenerateCompanyIdNAF) :
    con += [fieldstrToTargetXML(fieldStr(CustTable, CompanyNAFCode))];
    break;
```

#### Створення класу

- 1) В AOT створіть клас для сутності з наступними властивостями:
  - ✓ [DMFAttribute(true)]
  - ✓ DMFClassName extends DMFEntityBase
  - ✓ Зазначте об'єкт для проміжної таблиці з назвою "entity" (сутність).
  - ✓ Зазначте об'єкт для основної таблиці для цільової сутності з назвою "target".
  - ✓ Наступний приклад показує, як зазначається DMFBEMATableEntityClass.



```
[DMFAttribute(true)]
public class DMFBEMTableEntityClass extends DMFEntityBase
{
    DMFBEMTableEntity entity;
    BEMTable target;
}
```

- 2) Створення **нового** методу:
- ✓ Взяти проміжну таблицю як джерело параметрів
  - ✓ Сутність значення повинна ініціалізуватись з параметра. Наступний приклад показує новий метод DMFBEMTableEntity

```
public void new(DMFBEMTableEntity _entity)
{
    entity = _entity;
}
```

- 3) Створення **конструкційного** методу:
- ✓ Проміжна таблиця береться як джерело параметрів
  - ✓ Використовуючи параметр, створити та повернути об'єкт поточного класу. Наступний приклад показує **конструкційний** метод DMFBEMTableEntity

```
public static DMFBEMTableEntityClass construct(DMFBEMTableEntity _entity)
{
    DMFBEMTableEntityClass entityClass = new DMFBEMTableEntityClass(_entity);
    return entityClass;
}
```

- 4) Створення методу **setTargetBuffer**:
- ✓ Цільовий елемент може мати численні джерела даних. Одним з джерел даних є основна таблиця, в якій представлена сутність. У методі **setTargetBuffer** параметр **\_dataSourceName** відображає джерела даних, які наявні у запиті цільової сутності
  - ✓ В залежності від джерела даних, можете виникнути потреба в ініціалізації локального примірника таблиці для відповідного джерела даних. Такий локальний примірник можна використовувати у функціях, необхідних для міграції даних. Цільовий об'єкт має бути ініціалізований основною таблицею, яка відображає цільову сутність.

```
public void setTargetBuffer(Common _common, Name _dataSourceName = "")
{
    switch (_common.TableId)
    {
        case tableNum(BEMTable) :
            target = _common;
            break;
    }
}
```

- 5) Встановлення властивості класу **RunOn** на значення **CalledFrom**

Напишіть функції для імпорту та експорту даних

Співставити дані з проміжної таблиці з цільовою таблицею можна двома шляхами:

1) Призначте поля: поле з проміжної таблиці присвоюється полю в цільовій таблиці. В такому випадку типи даних для проміжного та цільового полів повинні бути однаковими.

2) Напишіть функцію для трансформації значень полів з проміжної до цільової таблиці. Ви можете написати функції X++ для трансформації і співставлення даних з проміжної до цільової таблиці.

Написання функції

Визначені вами функції імпорту та експорту даних повинні виконувати наступні дії:

1) Введення (Джерело): цілий проміжний запис доступний як локальна змінна для класу, то ж немає потреби переносити до цього класу якісь параметри.

2) Виведення (Ціль): в результаті виконання функції в цільовій функції встановлюються нульові або множинні поля. Тип значення, що повертається функцією – це контейнер, який може містити нульові або більше значення, які встановлюються для цільового об'єкту.

Послідовність значень, які повертаються певною функцією, визначається у методі **getReturnFields**.

Метод addStagingLink

Метод addStagingLink використовується для встановлення взаємозв'язку між проміжною та цільовою таблицями, коли він не встановлюється шляхом використання властивості взаємозв'язку проміжної таблиці.

Цільовий запит та проміжний запис є доступними у методі, тому діапазон між цільовим та проміжним об'єктами може додаватись шляхом використання коду.

Наприклад: DMFEmployeeEntityClass

```
public Query addStagingLink(Query query, TableId _entityTableId, Common _staging)
{
    QueryBuildDataSource qbd;
    qbd = query.dataSourceTable(tableNum(HcmWorker));
    qbd.addRange(fieldNum(HcmWorker,PersonnelNumber)).value(_staging.(fieldNum(DMFEmployeeEntity,PersonnelNumber)));
    return query;
}
```

Метод getReturnFields

Метод getReturnFields використовується для зазначення виведених за замовчуванням даних (default output) або цільових полів для функцій, що використовуються для міграції даних.

Параметри включають:

- ✓ `_entity`: entity name
- ✓ `_name`: function name

Функція повинна повернути контейнер, а також назву джерела даних в запиті цільової сутності, з якою повинен виконуватись метод, та назву поля джерела даних у запиті цільової сутності, яка повинна ініціалізуватись при виконанні функції.

```

public static container getReturnFields(Name _entity, MethodName _name)
{
    DataSourceName dataSourceName = queryDataSourceStr(DMFEmployeeTargetEntity, HcmEmploymentDetail);
    Container con = [dataSourceName];

    Name fieldstrToTargetXML(Fieldname _fieldName)
    {
        return DMFTargetXML::findEntityTargetField(_entity, dataSourceName, _fieldName).XMLField;
    }

    switch (_name)
    {
        case methodStr(DMFEmployeeEntityClass, GenerateReasonCode) :
            con += [fieldstrToTargetXML(fieldStr(HcmEmploymentDetail, TransitionReasonCode))];
            break;

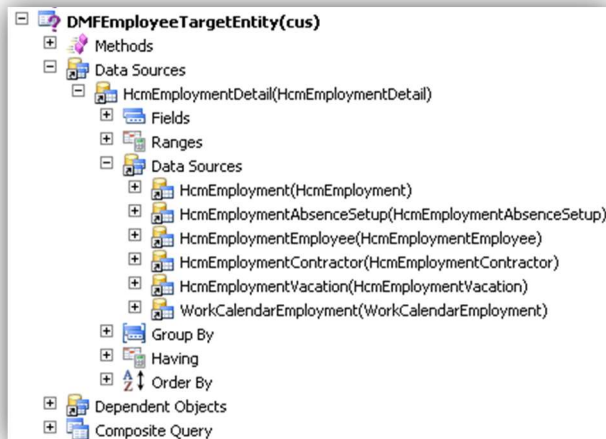
        //change datasource
        case methodStr(DMFEmployeeEntityClass, GenerateDefaultDimension) :
            dataSourceName = queryDataSourceStr(DMFEmployeeTargetEntity, HcmEmployment);
            con = [dataSourceName];
            con += [fieldstrToTargetXML(fieldStr(HcmEmployment, DefaultDimension))];
            break;

        default :
            con = conNull();
    }

    return con;
}

```

Це дасть зрозуміти інструменту міграції, до якої таблиці та поля має йти значення, повернене від метода генерування.



## Імпорт налаштовуваної сутності

Після того як створено код, здійсніть наступні кроки для використання налаштовуваної сутності:

- 1) Data import/export framework > Setup > Target entities
- 2) Data import/export framework > Periodic > Processing group

## Внести зміни до цільової сутності

- 1) Знайдіть запис з сутністю, яку бажаєте використовувати. Натисніть “Modify target mapping”.

Entity type	Entity	Entity name	Staging table	Entity class	Target entity	Set based
Entity	Customer	Customer	DMFCustomerEntity	DMFCustomerEntityClass	DMFCustomerTargetEntity	<input type="checkbox"/>
Entity	Customer address	Customer address	DMFCustomerAddressEntity	DMFCustomerAddressEntityClass	DMFCustomerAddressTargetEntity	<input type="checkbox"/>
Entity	Customer address	Customer address SHM	DMFCustomerAddressEntity	DMFCustomerAddressEntityClass	DMFCustomerAddressTargetEntity	<input type="checkbox"/>
Entity	Customer	Customer SHM	DMFCustomerEntity	DMFCustomerEntityClass	DMFCustomerTargetEntity	<input type="checkbox"/>

- 2) Натисніть “Generate mapping”: з цієї функцією вставляються всі нові поля і оновлюється співставлення.

Staging field	Mapping type	Function	Target field	Sequence	Return parameter sequence	Call validateField method	Call modifiedField method
*vpm*	=		CreationDateWPM	0	0	<input type="checkbox"/>	<input type="checkbox"/>
CreationDateWPMSHM	=		CreationDateWPM	0	0	<input type="checkbox"/>	<input type="checkbox"/>
ForecastCustGroupIdWPM...	=		ForecastCustGroupIdWPM	0	0	<input type="checkbox"/>	<input type="checkbox"/>
InvoiceMethodWPMSHM	=		InvoiceMethodWPM	0	0	<input type="checkbox"/>	<input type="checkbox"/>
Name2WPMSHM	()	GenerateParty		1	13	<input type="checkbox"/>	<input type="checkbox"/>
SalesBackorderWPMSHM	=		SalesBackorderWPM	0	0	<input type="checkbox"/>	<input type="checkbox"/>
SalesUnitIdWPMSHM	=		SalesUnitIdWPM	0	0	<input type="checkbox"/>	<input type="checkbox"/>

## Деталі співставлення (якщо ви додаєте всі нові поля вручну)

Для нормальних полів, які мають той самий тип перелічення (Enum type) у таблиці DMF та в цільовій таблиці, вставте “Staging field” у перший стовпчик, а “Target field” у четвертий стовпчик. В цьому випадку тип співставлення буде “=”.

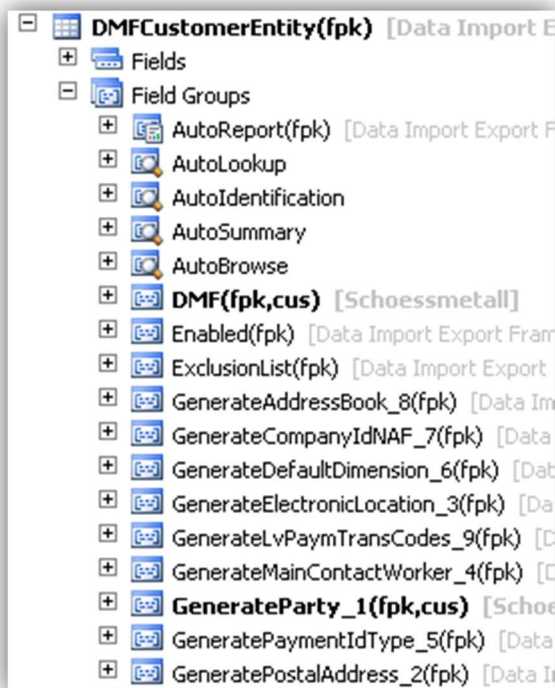
У стовпчику “Staging field”, одне поле можна порівнювати більше ніж один раз, але (у цьому випадку) значення “Target field” повинно бути іншим. Наприклад (сутність Product):

ItemId	=		EcoResStorageDimensionGroupItem_ItemId
ItemId	=		EcoResTrackingDimensionGroupItem_ItemId
ItemId	=		InventItemGroupItem_ItemId
ItemId	=		InventItemInventSetup_ItemId
ItemId	=		InventItemPurchSetup_ItemId
ItemId	=		InventItemSalesSetup_ItemId
ItemId	=		InventItemSetupSupplyType_ItemId
ItemId	=		InventModelGroupItem_ItemId
ItemId	=		InventTable_ItemId
ItemId	=		InventTableModuleInvent_ItemId
ItemId	=		inventTableModulePurch_ItemId
ItemId	=		InventTableModuleSales_ItemId
ItemId	=		ProductNumber

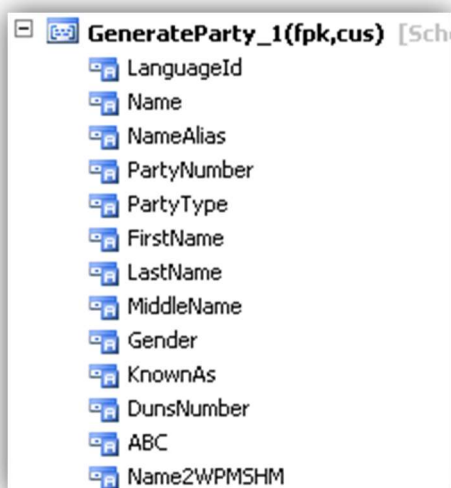
Зверніть увагу, що у наведеному вище скріншоті є поле з назвою “Name2WPMSHM”.

У цьому випадку це поле заповнене функцією. Назва функції зазначена у стовпчику “Function”, а типом співставлення (“Mapping type”) є “()”. Метод “GenerateParty” відноситься до класу “DMFCustomerEntityClass”. У цьому випадку у стовпчиках наведені значення:

- 1- Послідовність (Sequence): у таблиці “DMFCustomerEntity” поля для цього методу зазначені у FieldGroup “GenerateParty\_1”. Звідси береться утиліта нумерації цих груп полів. Номер 1 той самий, що зазначений у стовпчику “Sequence”.



2- Послідовність параметрів, що повертаються (Return parameter sequence): цей номер є номером поля у групі полів. Повернення функції здійснюється у відповідності до групи полів.








```
return [party.parmLanguageId(), party.parmName(), party.parmNameAlias(), party.parmPartyNumber(),
enum2str(party.parmPartyType(), party.parmFirstName(), party.parmLastName(), party.parmMiddleName(),
party.parmGender(), party.parmKnownAs(), dunsNumber.DunsNumber, party.parmABC(), party.parmName2WPM());
```

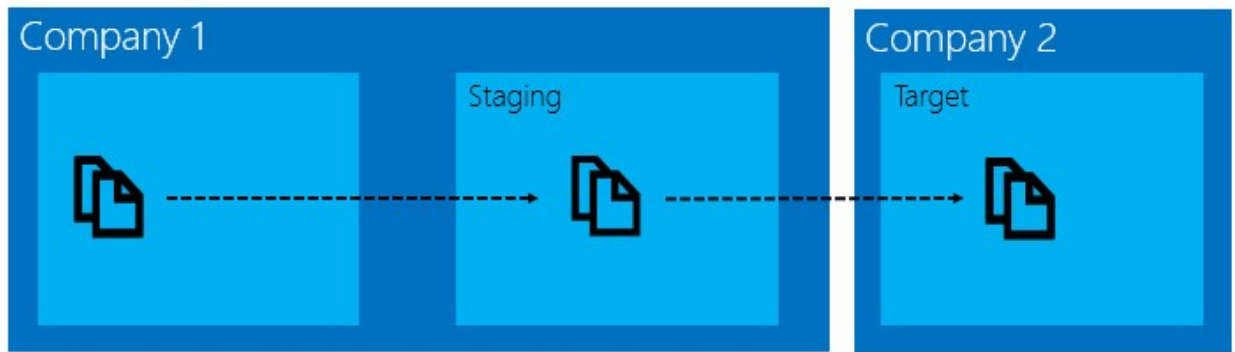
Група обробки (Processing Group)

1) Якщо є якийсь запис для певного кейсу – створіть новий. У прикладі (нижче) налаштована сутність Customer є частиною групи – description “Customer - CustTable”, а налаштована сутність “Customer address” має новий запис “Customer address – Customer address SHM”.

Processing group (1)		
File New Delete Entities Get staging data Copy data to target Execution history Export to AX		
Group name	Description	
Customer	CustTable	
Customer address	Customer address SHM	
Product	Product Eco Res, Inventtable	
Product2	Product 2	
Stuecklist2	2 stueckliste	
StuecklVersion	1 Stuecklisten Version	

2) Натисніть на “Entities” і додайте нову налаштовувану сутність.

Select entities for processing group (1) - Group name: Customer, CustTable					
File  New  Delete  Select  Generate source file  Generate source mapping  Modify source mapping  Validate					
<input type="checkbox"/> Sequence	Entity name	Source data format	Run business logic in insert or update method	Run business validations	
<input type="checkbox"/> 2	CustomerSHM	TXTPipe	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	



- 1) Визначте формат **Data Import/Export Framework > Setup > Source data formats**:

Source name	Description	Type
AX	AX	AX
ODBC	ODBC	ODBC
CSV	csv with delimit...	File
TXT	TXT File with ;	File
TXTPipe	TXT with	File

**Application**

**Dimensions**

Dimension code:

Chart of accounts delimiter:

Dimension format:

**Name sequence**

Name sequence:

- 2) Створіть нову групу обробки **Data Import/Export Framework > Common > Processing group** (Наприклад: ExportCust)
- 3) Натисніть "Entities" та введіть новий запис:
  - a. Entity name = Customer
  - b. Source Data Format = AX (той, що ви створили)
- 4) Натисніть "Select" і вкажіть критерії, які бажаєте використовувати для експорту даних

Sequence	Entity name	Source data format	Run business logic in insert or update method	Run business validations
1	Customer	AX	<input type="checkbox"/>	<input type="checkbox"/>

**DMFCustomerTargetEntity (1 - shm)**

Select query:  Query used

Table: Customers

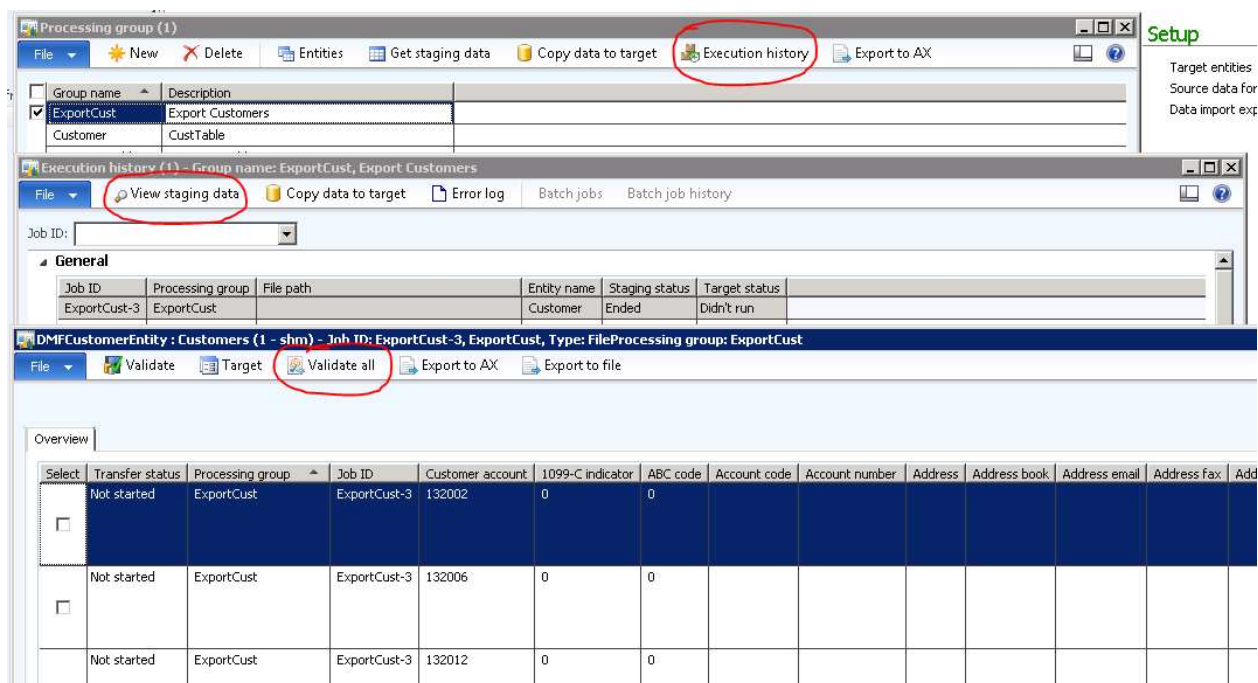
Range:  Sorting:

Table	Derived table	Field	Criteria
Customers	Customers	Customer account	132002..132062

Buttons: Add, Remove, Reset, OK, Cancel

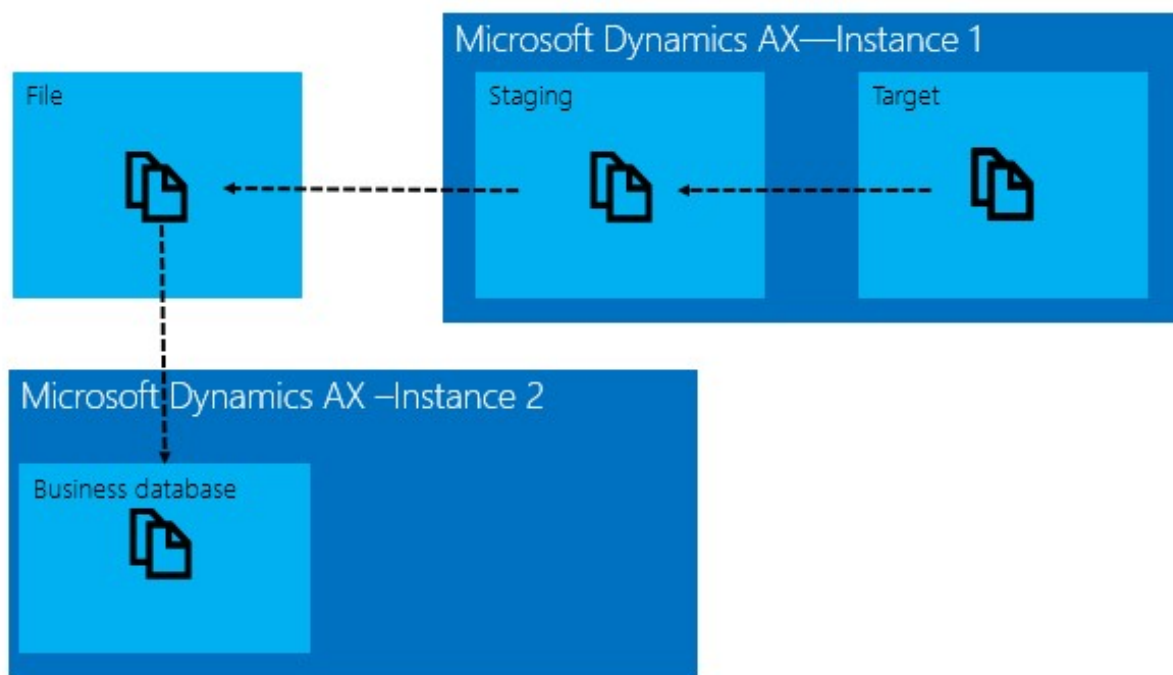
- 5) Закрийте форму **Data Import/Export Framework > Common > Processing group**, натисніть "Get staging data".
- 6) Натисніть "Execution history" у формі **Data Import/Export Framework > Common > Processing group**, потім на "View staging data" та валідуйте дані (Validate all).



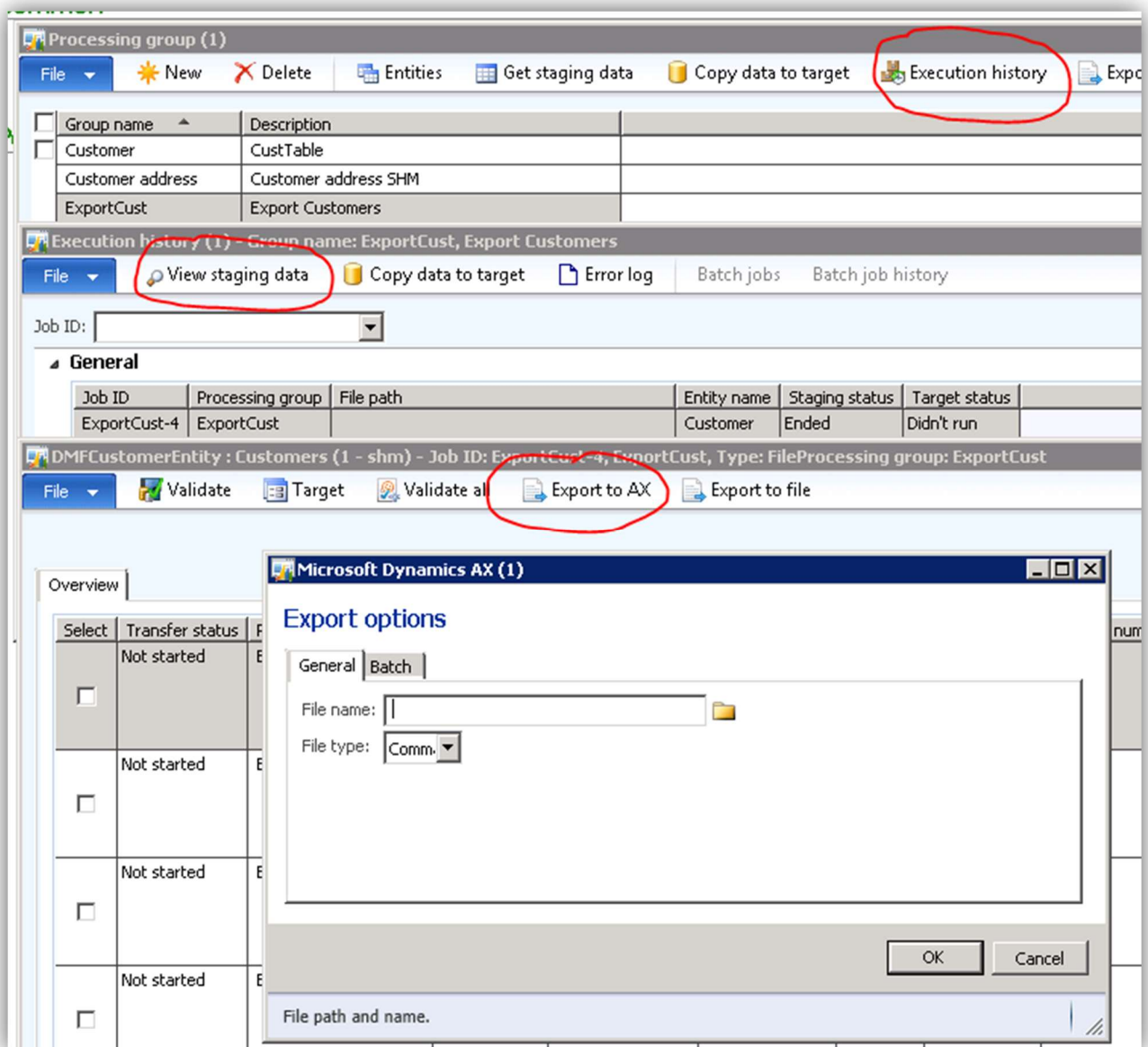


7) Змініть компанію та в **Data Import/Export Framework > Common > Processing group** натисніть “Copy data to target”

Переміщення даних між екземплярами Microsoft Dynamics AX



- 1) Виконайте кроки 1-6 з попереднього параграфу “Переміщення даних між компаніями в Microsoft Dynamics AX”.
- 2) Натисніть “Export to AX”



3) Введіть назву файлу і натисніть Ок.

4) Для імпорту даних до іншої системи:

a. **System administration > Common > Data export/import > Import**

b. На вкладці **General** оберіть файл даних, який бажаєте імпортувати

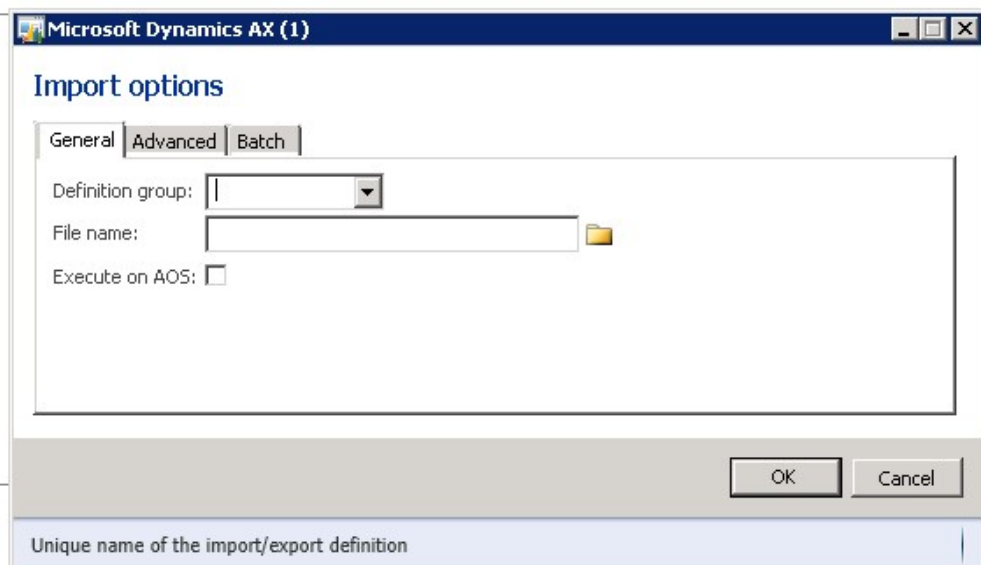
c. На вкладці **Batch** оберіть групу пакету (batch group), з якою ви бажаєте, щоб оброблявся імпортований файл.

## Common

- Users
  - Users
  - Online users
  - External web users
  - User relations
  - User groups
  - User profiles
  - User requests
- Data export/import
  - Definition groups
  - Export to
  - Import

## Reports

- Licensing
- Database





Щоб отримати більше інформації, скористуйтеся наступним посиланням:  
<http://technet.microsoft.com/en-us/library/jj225591.aspx>