

MB-500: Microsoft Dynamics 365: Finance and Operations Apps Developer

Lab 6a - Data Export/Import using Data Entity

Contents

Lab Environment.....	2
Lab Overview	2
Scenario 1	2
Scenario 2.....	2
Exercise 1: Create New Entity for import	3
Task 1: New Data Entity: DDTCustFlyDetailsEntity	3
Task 2: Data Management Setup.....	4
Exercise 2: Import Customer data updating miles.....	6
Task 1: Chain of Command: DDTCustFlyDetailsEntity > mapEntityToDataSource method ..	6
Task 2: Data Management Setup.....	7
Check Output.....	9

Lab Environment

In order to run this lab, you will need:

- An all-in-one demo data VM with
 - o Visual Studio installed, and a Visual Studio subscription
 - o A browser to run the user interface
 - o Lab 5 – Code Extension & Development completed

Lab Overview

- Dependency: Lab 5 – Code Extension & Development should be completed
- Creation and extension of a Data Entity
- Set up Data Management for importing external data into Dynamics 365 for Finance and Operations

Estimated time to complete this lab: 30+ minutes

Scenario 1

- In the DynamicsDevTraining package you need to create a new Data Entity DDTCustFlyDetailsEntity with the table DDTCustFlyDetails
- Set up Data Management to import the Customer Flying Details from the csv file CustFlyDetails.csv

Scenario 2

- The next exercise is for package MyLabAirlines. In this exercise, Excel won't have the field FlyingMiles. During the import process the Flying Miles value should be populated from the table MLAAirportMilesChart
- This needs you to create an extension of the data entity DDTCustFlyDetailsEntity
- Finally, set up Data Management again to import the Customer Flying Details from the csv file CustFlyDetails-nomiles.csv

Exercise 1: Create New Entity for import

Task 1: New Data Entity: *DDTCustFlyDetailsEntity*

1. Open DynamicsDevSolution in Solution Explorer
2. Right click the project and select **Add > New Item**
3. Select Data Entity under **Dynamics 365 Items > Data Model**
4. Create a new data entity DDTCustFlyDetailsEntity
 - a. Primary Data source: DDTCustFlyDetails
 - b. Entity Category: Transaction
 - c. Enable Public API: *checked*
 - d. Public Entity Name: CustFlyDetails
 - e. Public Collection Name: CustFlyDetails
 - f. Enable Data Management Capabilities: *checked*
 - g. Staging Table: DDTCustFlyDetailsStaging
 - h. View Privilege: DDTCustFlyDetailsEntityView
 - i. Maintain Privilege: DDTCustFlyDetailsEntityMaintain
5. Select the following fields in the Data Entity
 - a. CustAccount
 - b. FlyCount
 - c. FlyingDate
 - d. FlyingMiles
6. After creating the Data Entity, open it in the element designer
7. Navigate to **Data Sources > DDTCustFlyDetails > Data Sources**
8. Add a new Data Source
 - a. Name: AirportFrom
 - b. Table: DDTAirport
 - c. Is Read Only: Yes
 - d. Add Relation: DDTCustFlyDetails.FlyFrom = AirportFrom.RecId
 - i. Field = FlyFrom
 - ii. Join Data Source = DDTCustFlyDetails
 - iii. Related Field = RecId

9. Add another new Data Source to DDTCustFlyDetails:
 - a. Name: AirportTo
 - b. Table: DDTAirport
 - c. Is Read Only: Yes
 - d. Add Relation: DDTCustFlyDetails.FlyTo = AirportTo.ReclId
10. Drag AirportCode from both the data sources AirportFrom and AirportTo and drop it under the entity's Fields node in the following order:
 - a. CustAccount
 - b. FlyCount
 - c. FlyingDate
 - d. AirportFrom_AirportCode
 - e. AirportTo_AirportCode
 - f. FlyingMiles
11. Save, then Right click DDTCustFlyDetailsEntity in the Designer and execute "Regenerate staging table"
12. Save and Rebuild the project

Task 2: Data Management Setup

1. From the Dynamics 365 Finance and Operations dashboard, open the **Data Management** workspace
2. If the entity list is being refreshed, wait for a message indicating it's finished. If it doesn't finish within minutes, restart the refresh in the Framework parameters screen under Entity settings and select Refresh entity list.
3. Go to **Framework parameters** and change the following field values under General to Yes:
 - a. Ignore error
 - b. Create error file
 - c. Remove duplicates
4. Save and close the parameters form
5. Select the **Import** tile
6. Enter the following values:

- a. Group Name: Customer Flying Details Import
 - b. Select Add file (visible in the Enhanced view)
 - c. Source data format: CSV
 - d. Entity name: Customer Fly Details (DDTCustFlyDetailsEntity) – if this is not available, wait for the message saying the entity list has been refreshed
 - e. Keep all other default values
 - f. Upload and add from resource folder: CustFlyDetails.csv
7. Select **View map** icon in the Selected entities grid and make sure the source-staging mapping is correct, then close the mapping visualization
8. Select **Customer Fly Details** entity link (in the Selected entities grid) followed by **Modify Target Mapping** button and check if all mappings are correct between Staging table and Target. Switch between Mapping details and Mapping visualization
9. Select field FLYCOUNT and uncheck the checkbox “Call validate Field method”
10. Close the mapping and the target entities screens
11. Save and select **Import** button

Exercise 2: Import Customer data updating miles

Task 1: Chain of Command: DDTCustFlyDetailsEntity > mapEntityToDataSource method

1. Open MyLabAirlines in Solution Explorer
2. Right click project and select **Add > New Item**
3. Select Class under **Dynamics 365 Items > Code**
4. Create a new class MLACustFlyDetailsEntity_Extension
5. Write the following code to create a chain of command for Data Entity DDTCustFlyDetailsEntity:

```
[ExtensionOf(tableStr(DDTCustFlyDetailsEntity))]  
  
final class MLACustFlyDetailsEntity_Extension  
  
{  
  
}
```

6. Write the following code in that class:

```
public void mapEntityToDataSource(DataEntityRuntimeContext _entityCtx, DataEntityDataSourceRuntimeContext  
_dataSourceCtx)  
  
{  
  
    next mapEntityToDataSource(_entityCtx, _dataSourceCtx);  
  
    if (_dataSourceCtx.name() == DataEntityDataSourceStr(DDTCustFlyDetailsEntity, DDTCustFlyDetails))  
  
    {
```

```
DDTCustFlyDetails dsCustFlyDetails = _dataSourceCtx.getBuffer();

DDTCustFlyDetailsEntity deCustFlyDetailsEntity = _entityCtx.getEntityRecord();

    dsCustFlyDetails.flyingMiles =
MLAAirportMilesChart::getMiles(deCustFlyDetailsEntity.AirportFrom_AirportCode,
deCustFlyDetailsEntity.AirportTo_AirportCode);

    }

}
```

Task 2: Data Management Setup

1. From the Dynamics 365 Finance and Operations dashboard, open the **Data Management workspace**
2. Select the **Import tile**
3. Enter the following values
 - a. Group Name: Customer Flying Details Import without Miles
 - b. Add file; Source Data format: CSV
 - c. Entity name: Customer Fly Details (DDTCustFlyDetailsEntity)
 - d. Keep all other default values
 - e. Upload from resource folder: CustFlyDetailsWOMiles.csv
 - f. Close that window
4. Select the View map icon in the Selected entities grid and make sure the source-staging mapping is correct
5. Add the column FlyingMiles in the Staging Table in Mapping details and check the Auto default check box

Mapping visualization		Mapping details					
		+ New Delete Save Default value Conversion Query criteria					
<input checked="" type="checkbox"/>	Auto-generat...	Auto default	Source field	Staging field	Ignore blank ...	Text qualifier	Use enum label
<input type="checkbox"/>		<input type="checkbox"/>	FROMAIRPORT	AirportFrom_AirportCode	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
<input checked="" type="checkbox"/>		<input checked="" type="checkbox"/>	Default	FlyingMiles	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
<input type="checkbox"/>		<input type="checkbox"/>	TOAIRPORT	AirportTo_AirportCode	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
<input type="checkbox"/>		<input type="checkbox"/>	CUSTACCOUNT	CUSTACCOUNT	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
<input type="checkbox"/>		<input type="checkbox"/>	FLYCOUNT	FLYCOUNT	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
<input type="checkbox"/>		<input type="checkbox"/>	FLYINGDATE	FLYINGDATE	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>

6. Select the **Default value** button and enter some temporary number (e.g. 99) to fill in the staging table
7. Go back to the previous screen, save and select **import**

Check Output

- Follow the steps mentioned; you should be able to see the new imported records in the Flying Details tab of the Customer form.