|  |
| --- |
| Guide for updating LCP data in E-PRTR2  Documentation |

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **JOB NUMBER:** | | | **DOCUMENT REF:** | | | |
| **Version** | **Purpose of issue** | **Originated** | **Checked** | **Reviewed** | **Authorised** | **Date** |
| 1.0 | Initial Draft | Morten Hjelmsmark |  |  |  | 05.01.2016 |
|  |  |  |  |  |  |  |
| **ATKINS DANMARK** | | | | | | |

Contents

[1. Summary 2](#_Toc439766356)

[2. E-PRTR setup 3](#_Toc439766357)

[3. Preparation of the Access database 3](#_Toc439766358)

[3.1. Coordinates 3](#_Toc439766359)

[3.2. Primary Keys 3](#_Toc439766360)

[4. Connect to the Access Database in ArcCatalog 4](#_Toc439766361)

[5. Create Geodatabase 7](#_Toc439766362)

[5.1. File geodatabase 7](#_Toc439766363)

[5.2. Import Plant featureclass 7](#_Toc439766364)

[5.3. Import the tables 8](#_Toc439766365)

[5.4. Create Relationship classes 8](#_Toc439766366)

[6. Updating the ArcGIS Server Service 11](#_Toc439766367)

# Summary

Even though it was not the plan LCP data is now integrated into the updated E-PRTR site. So keep in mind that the LCP data setup was meant for demo purpose and hence not prepared for future data updates. Now that the LCP is integrated and we face data updates, we need a guideline.

# E-PRTR setup

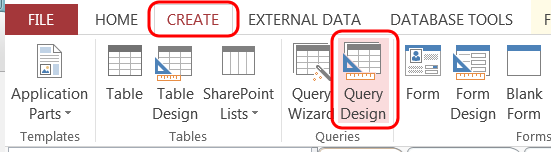
While the E-PRTR data is both served as spring hibernate based REST services and as ArcGIS Server mapservices, the LCP data is fully served as an ArcGIS Server service only. It is off course not good practice to have different setups within the same site, but in this case, the benefit is, that we only need to focus on updating one ArcGIS Server service.

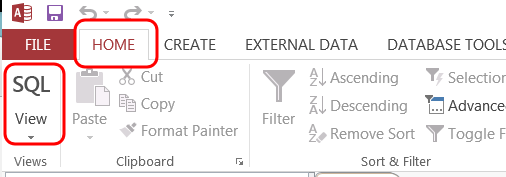
# Preparation of the Access database

The ArcGIS mapservice cannot be based directly on the Access database so we need to move the data into a file geodatabase. The first step is to prepare the Access database for conversion.

## Coordinates

Not all plants have coordinates so the first step is to assure that all non-numerical values in the longitude and latitude columns in the **2\_Plant** table are converted into 0.

Open the Access database and create a new Query:

Then choose SQL view:

Then you can paste this into the editor and run:

UPDATE [2\_Plant] SET [2\_Plant].Longitude = 0 WHERE IsNumeric([2\_Plant].Longitude) = FALSE;

And then this:

UPDATE [2\_Plant] SET [2\_Plant].Latitude = 0 WHERE IsNumeric([2\_Plant].Latitude) = FALSE;

On my Danish computer I also had to convert . to , before I converted the coordinates from string format into number.

UPDATE [2\_Plant] SET [2\_Plant].Latitude = Replace( [2\_Plant].Latitude, ".", ",");

UPDATE [2\_Plant] SET [2\_Plant].Longitude = Replace( [2\_Plant].Longitude, ".", ",");

Then open 2\_Plant in Design view and change the latitude and longitude columns into number with field size Double.

## Primary Keys

When we import the tables into an esri geodatabase the ID column will automatically be converted into esri’s own OBJECTID column. Since we want to base relations to the ID columns we need to duplicate them.

*1\_BasicData*

Add the new column by running this in the Query:

ALTER TABLE 1\_BasicData ADD COLUMN BasicID INT

And then we set the values by running this:

UPDATE [1\_BasicData] SET [1\_BasicData].BasicID = [1\_BasicData].ID;

*2\_Plant*

Add the new column by running this in the Query:

ALTER TABLE 2\_Plant ADD COLUMN PlantID INT

And then we set the values by running this:

UPDATE [2\_Plant] SET [2\_Plant].PlantID = [2\_Plant].ID;

I think this is it; the next step is to create a file geodatabase and import the LCP data from the Access database.

# Connect to the Access Database in ArcCatalog

Accessing an Access database from ArcCatalog is a bit tricky so here is a guide:

|  |  |
| --- | --- |
| First we need to find the OLD DB connection tool to be able to connect to the Access database.  In the **Customize** tab, choose **Toolbars** and then click **Customize** | In the **Customize** window choose **Commands** tab and write ‘**Add Ole DB**’ in the search box. Drag the tool into an existing toolbox. |
| Choose the **Microsoft OLE DB Provider for ODBC Drivers Next>>** | Select **Use connection string** and click **Build** |
| Choose **Machine Data Source** tab and click **New** | Assure that **User data Source** is selected and click **Next>>** |
| Choose **Microsoft Access Driver** and click **Next>** | Click **Finish** and a new window will open |
| Click **Select** and browse to the Access file, then set a name for the data source and click **OK** | The data source is now listed as a machine data source, click **OK** |
| The Access Database is probably not security enabled so you can leave the login empty and click  **OK** | Now the connection string is set and you can test the connection, if it succeeds click **OK** |
| Update the **Database Connections** folder in ArcCatalog and a new **OLE DB Connection.odc** will appear – now rename it to avoid issues later on. |  |

# Create Geodatabase

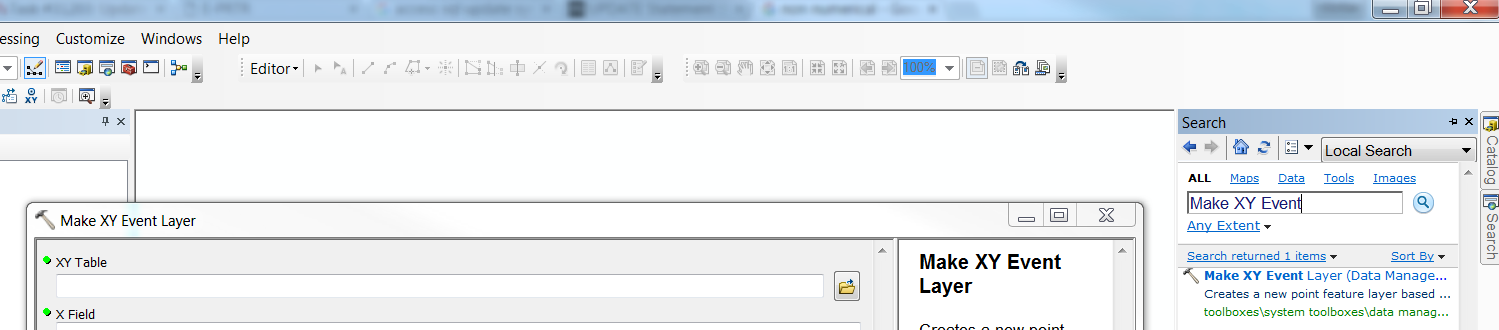
In this section we create a new File Geodatabase, import the 2\_Plant into a point feature class, import the rest of the tables and then we create the relationship classes.

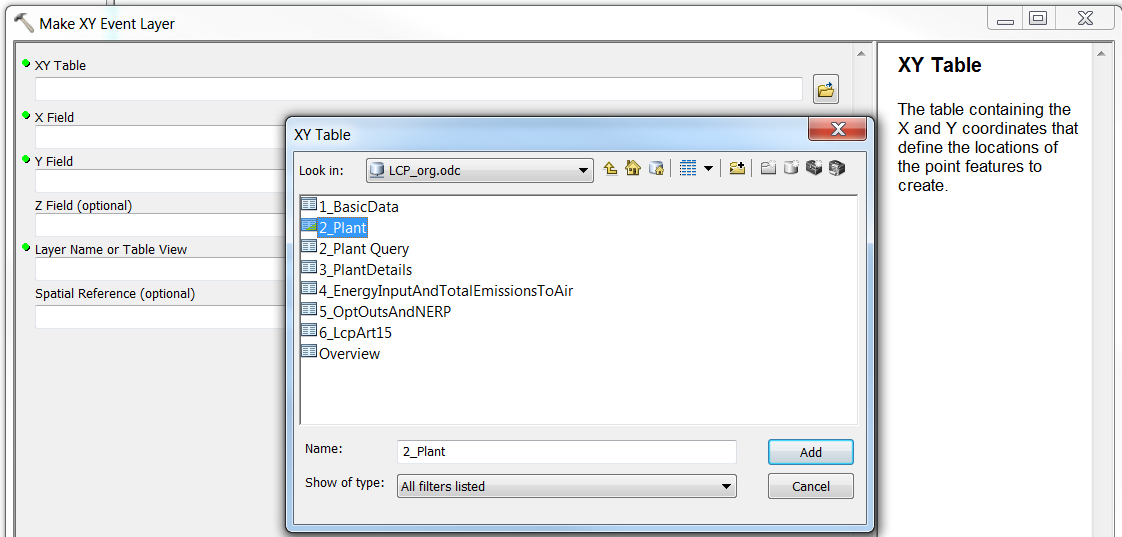
## File geodatabase

|  |  |
| --- | --- |
| In AcrCatalog add a new **Folder Connection** to where you want to save the new file GeoDatabase | In the folder right click and choose **New** and then **File GeoDatabase** and give the file geodatabase a proper name |

## Import Plant featureclass

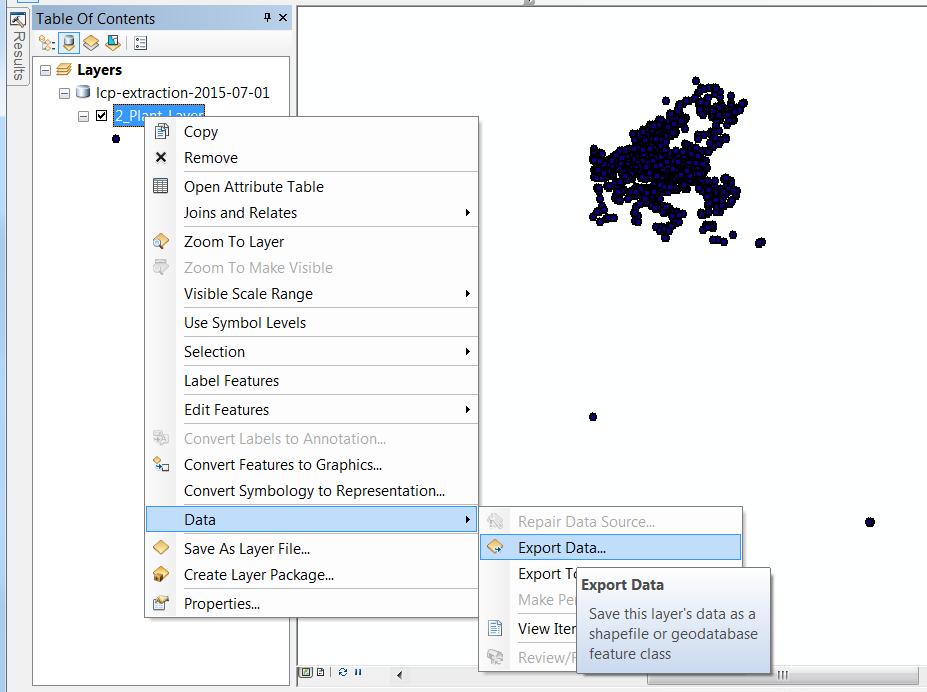
This part is done in ArcMAP where you search, find and open the tool **Make XY Event layer.**

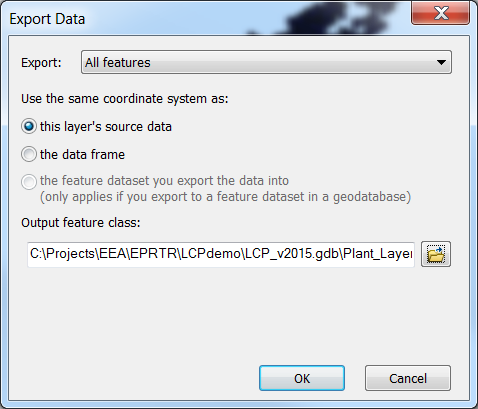


For the XY Table browse to the 2\_Plant table by using the ODBC Database Connection we just created before.

Choose **Latitude** for the *X Field* and **Longitude** for the *Y Field*, keep the suggested name for the layer and choose **GCS\_WGS\_1984** for the *Spatial Reference.*

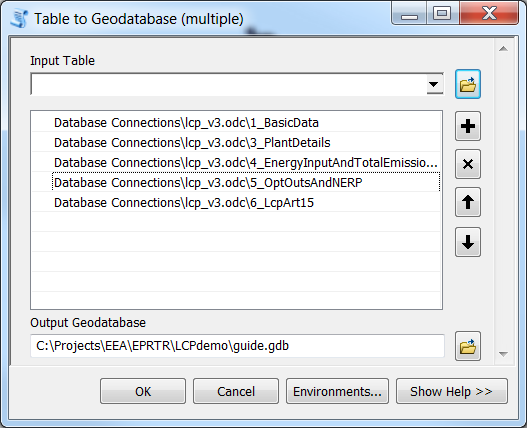
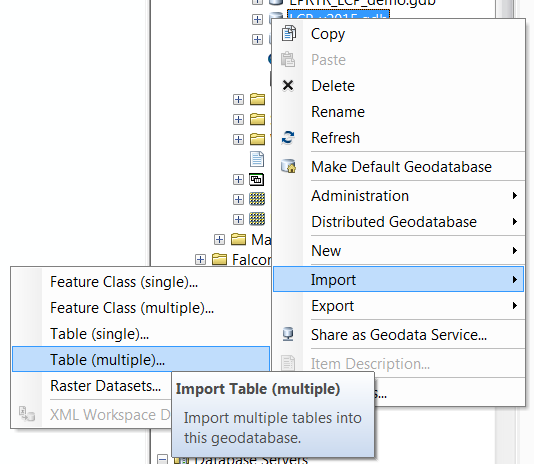
Now validate that the points / facilities are correct.

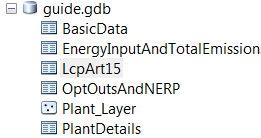
**If the data is correct, then you right click the layer, choose Data and then Export Data.

In the dialog choose **All features**, choose **this layer’s source data** and browse to the file geodatabase and save as **Plant\_Layer**

## Import the tables

In the Catalog window navigate to the file geodatabase, right click, choose **Import** and then click **Table (Multiple)**. Select the five tables through our ODBC connection; 1\_BasicData, 3\_PlantDetails, 4\_EnergyInputAndTotalEmissionsToAir, 5\_OptOutsAndNERP and 6\_LcpArt15.



Then you should rename the tables by removing the initial part; T[number]\_ (T1\_), so that the list looks like the one on the right.

## Create Relationship classes

The relationship classes are for querying the tables through the ArcGIS Server service. With these relationship classes we just recreate the relationships from the Access database.

*BasicData\_Plant*

|  |  |
| --- | --- |
| Browse to the file geodatabase in the Catalog window, right click and choose **New** and then **Relationship Class..** | The name of this class should be **BasicData\_Plant**, select **BasicData** as *origin table* and select **Plant\_Layer** as *destination*. Then click **Next** |
| Since we’re only querying data we choose a **Simple** relationship, click **Next** | We don’t need messages, click **Next** |
| Choose **1-M(one to many)** relationship, click **Next** | We don’t want to append attributes to the origin table, click **Next** |
| The *primary key* for the **BasicData** table is the **BasicID** column we created earlier, the *foreign key* for the **Plant\_Layer** table is **FK\_BasicData\_ID**, click **Next** | Click Finish |

Here are the settings for the last four relationship classes:

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Name | Origin | Destination | Relationship | Primary key | Foreign key |
| Plant\_EIATETA | Plant\_Layer | EnergyInputAndTotalEmissionsToAir | 1-M | PlantID | FK\_Plant\_ID |
| Plant\_ LcpArt15 | Plant\_Layer | LcpArt15 | 1-M | PlantID | FK\_Plant\_ID |
| Plant\_OOANERP | Plant\_Layer | OptOutsAndNERP | 1-M | PlantID | FK\_Plant\_ID |
| Plant\_ PlantDetails | Plant\_Layer | PlantDetails | 1-M | PlantID | FK\_Plant\_ID |

Alternatively, you can run these commands in the ArcMAP or ArcCatalog Python dialog:

arcpy.CreateRelationshipClass\_management(r'C:\Projects\EEA\EPRTR\LCPdemo\LCP\_2016v1.gdb\BasicData',r'C:\Projects\EEA\EPRTR\LCPdemo\LCP\_2016v1.gdb\Plant\_Layer',"BasicData\_Plant","SIMPLE","PlantLayer","Basic","NONE","ONE\_TO\_MANY","NONE","BasicID","FK\_BasicData\_ID")

arcpy.CreateRelationshipClass\_management(r'C:\Projects\EEA\EPRTR\LCPdemo\LCP\_2016v1.gdb\Plant\_Layer',r'C:\Projects\EEA\EPRTR\LCPdemo\LCP\_2016v1.gdb\EnergyInputAndTotalEmissionsToAir',"Plant\_EIATETA","SIMPLE","EnergyInputAndTotalEmissionsToAir","PlantLayer","NONE","ONE\_TO\_MANY","NONE","PlantID","FK\_Plant\_ID")

arcpy.CreateRelationshipClass\_management(r'C:\Projects\EEA\EPRTR\LCPdemo\LCP\_2016v1.gdb\Plant\_Layer',r'C:\Projects\EEA\EPRTR\LCPdemo\LCP\_2016v1.gdb\LcpArt15',"Plant\_LcpArt15","SIMPLE","LcpArt15","PlantLayer","NONE","ONE\_TO\_MANY","NONE","PlantID","FK\_Plant\_ID")

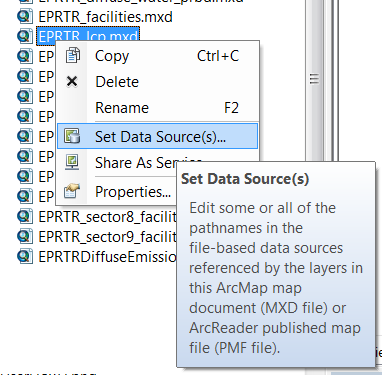
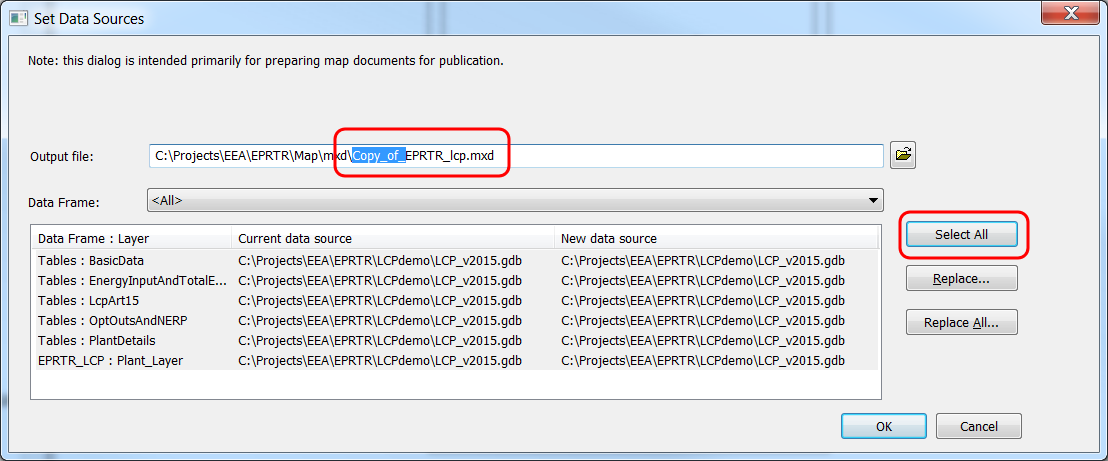
arcpy.CreateRelationshipClass\_management(r'C:\Projects\EEA\EPRTR\LCPdemo\LCP\_2016v1.gdb\Plant\_Layer',r'C:\Projects\EEA\EPRTR\LCPdemo\LCP\_2016v1.gdb\OptOutsAndNERP',"Plant\_OOANERP","SIMPLE","OptOutsAndNERP","PlantLayer","NONE","ONE\_TO\_MANY","NONE","PlantID","FK\_Plant\_ID")

arcpy.CreateRelationshipClass\_management(r'C:\Projects\EEA\EPRTR\LCPdemo\LCP\_2016v1.gdb\Plant\_Layer',r'C:\Projects\EEA\EPRTR\LCPdemo\LCP\_2016v1.gdb\PlantDetails',"Plant\_PlantDetails","SIMPLE","PlantDetails","PlantLayer","NONE","ONE\_TO\_MANY","NONE","PlantID","FK\_Plant\_ID")

# Updating the ArcGIS Server Service

The last part is to update the service. Get the map document (MXD) from SVN: <https://svn.eionet.europa.eu/repositories/EPRTR/trunk/Website/Map/mxd/EPRTR_lcp.mxd>.

Now you need to redirect the Plant layer and tables to point to the new file geodatabase, which can be done in two ways;

1. In ArcCatalog navigate to the map document, right click and choose Set Data Source(s)

You don’t need to create a new map document, so you can delete the *Copy\_of\_* part of the output name. Then click **Select All** and then **Replace**. Here you can change the old file geodatabase path with the new, click **replace** and then **OK**.

1. Open the map document with ArcMAP. Then you need to set the datasource for each layer and table manually. If the existing file geodatabse cannot be reached, then you get the possibility to use repair datasource. With repair datasource, you just need to set the data source for the first layer and ArcMAP will set the rest.

Now we need to share the map document. First, you need to assure that the map document points at the correct file geodatabase. Then share as Service (ArcMAP: File -> Share As -> Service; ArcCatalog: Right click document -> Share As Service).

|  |  |
| --- | --- |
| Choose Overwrite an existing service, click Next | Connect to the EEA test ArcGIS Server (Contact Sebastien Petit [Sebastien.Petit@eea.europa.eu](mailto:Sebastien.Petit@eea.europa.eu) for credentials). Then select the **EPRTR\_LCP** in the list and click **Continue** |
| There should be no reason to change the settings, so click **Analyse**. If no errors are reported, then click **Publish**. Accept that data will be copied to the server. |  |

Now you can test that the service works and when accepted contact Sebastien Petit [Sebastien.Petit@eea.europa.eu](mailto:Sebastien.Petit@eea.europa.eu) to get him to upgrade the service from the test environment to production.