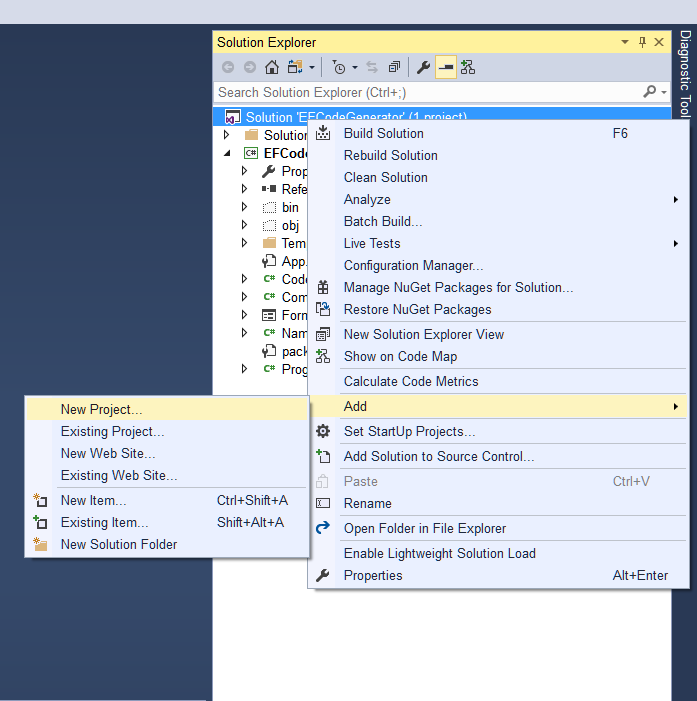
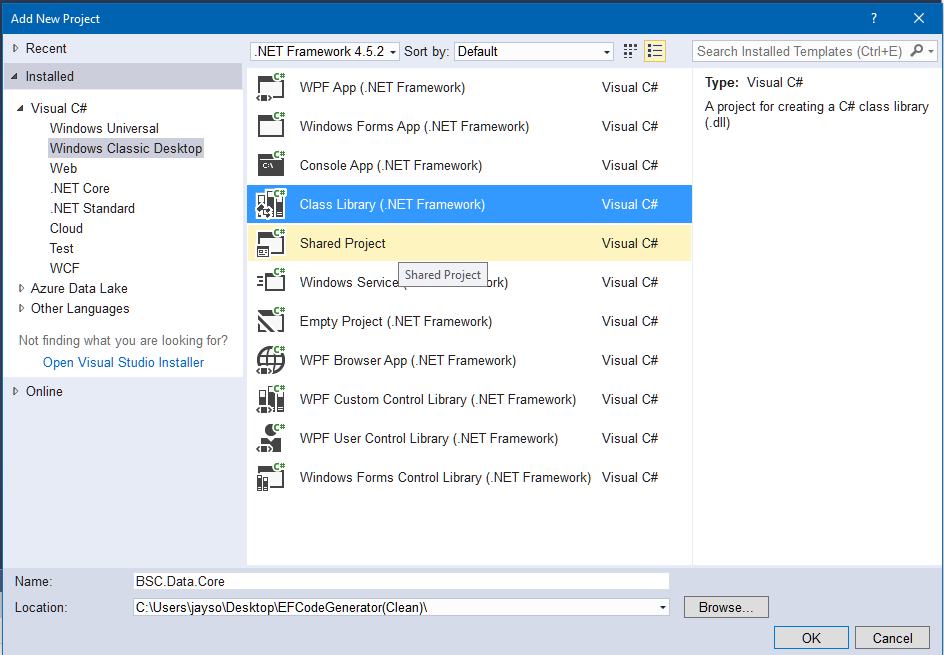
How to setup a new DAL Project?

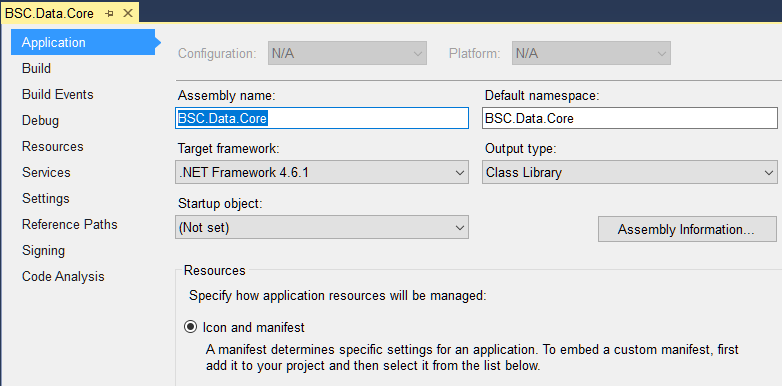
1. Open **EFCodeGenerator.sln** from Visual Studio 2017.
2. Right click on EFCodeGenerator solution and add a new **Class Library (.NET Framework)** project template.



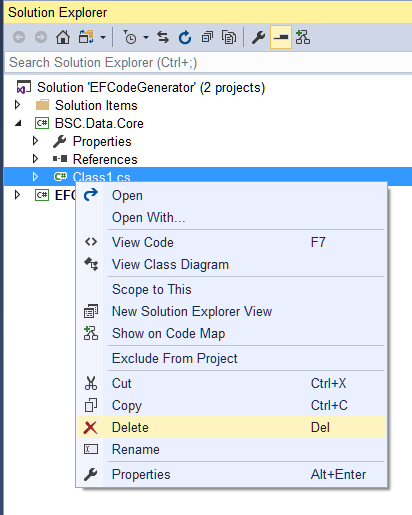
1. Give the project a name and click OK (Ex. BSC.Data.Core)



1. Ensure you are target .NET Framework is 4.5.1 or later or make the target framework to match with the EFCodeGenerator project which is 4.6.1.



1. Delete the default class generated (Class1.cs).

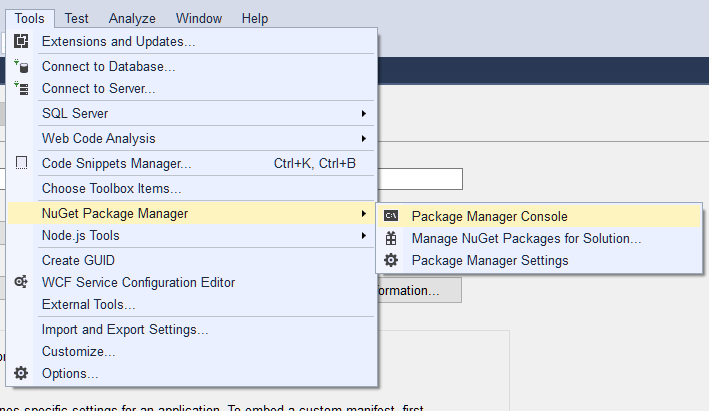


1. Install Entity Framework Core

To use EF Core, install the package for the database provider(s) you want to target.

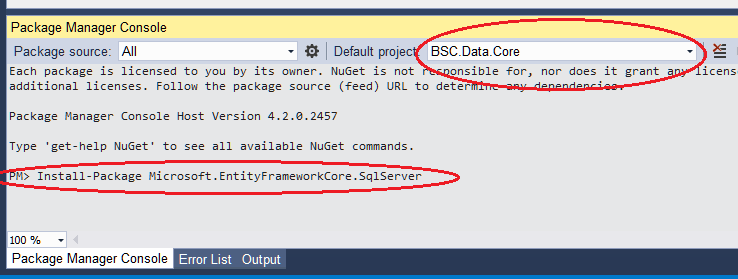
Note: This walkthrough uses SQL Server.

1. Go to Tools ‣ NuGet Package Manager ‣ Package Manager Console

****

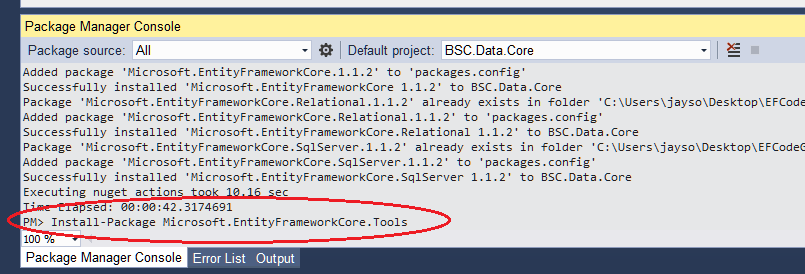
* Run **Install-Package Microsoft.EntityFrameworkCore.SqlServer**

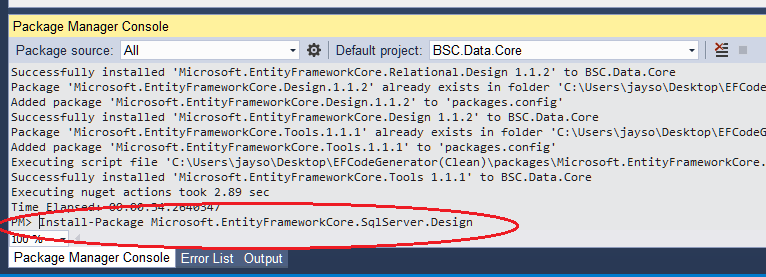
Make sure that the target project defaults to the DAL project created.

****

To enable reverse engineering from an existing database we need to install a couple of other packages too.

* Run **Install-Package Microsoft.EntityFrameworkCore.Tools**
* Run **Install-Package Microsoft.EntityFrameworkCore.SqlServer.Design**

****

****

1. Reverse engineer your model

Now it's time to create the EF model based on your existing database.

* Tools –> NuGet Package Manager –> Package Manager Console
* Run the following command to create a model from the existing database and set the output directory to Models

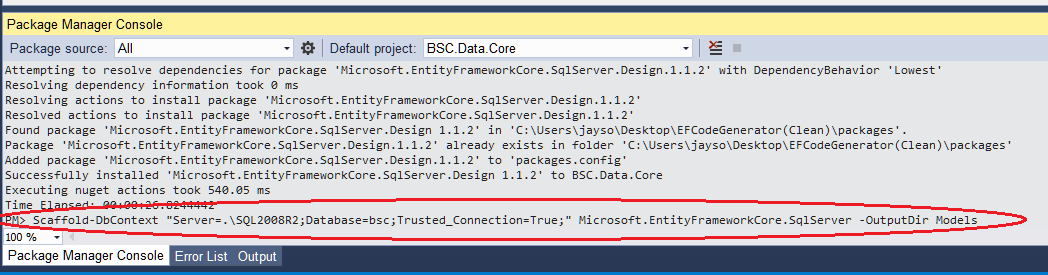
**Scaffold-DbContext "Server=localhost;Database=bsc;Trusted\_Connection=True;" Microsoft.EntityFrameworkCore.SqlServer -OutputDir Models**

Or add a –Project parameter to specify the target project

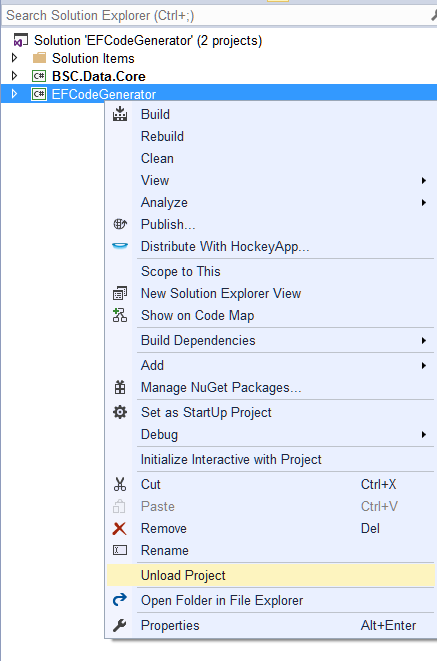
**Scaffold-DbContext "Server=localhost;Database=bsc;Trusted\_Connection=True;" Microsoft.EntityFrameworkCore.SqlServer -OutputDir Models -Project BSC.Data.Core**

To re-scaffold the entire migration, use the -Force parameter.

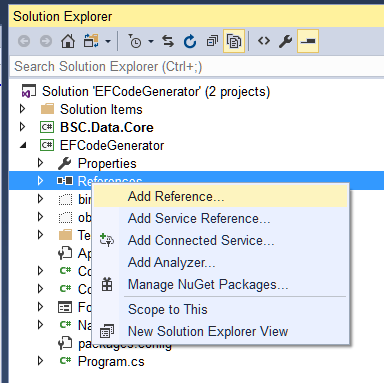
**Scaffold-DbContext "Server=localhost;Database=bsc;Trusted\_Connection=True;" Microsoft.EntityFrameworkCore.SqlServer -OutputDir Models –Force**

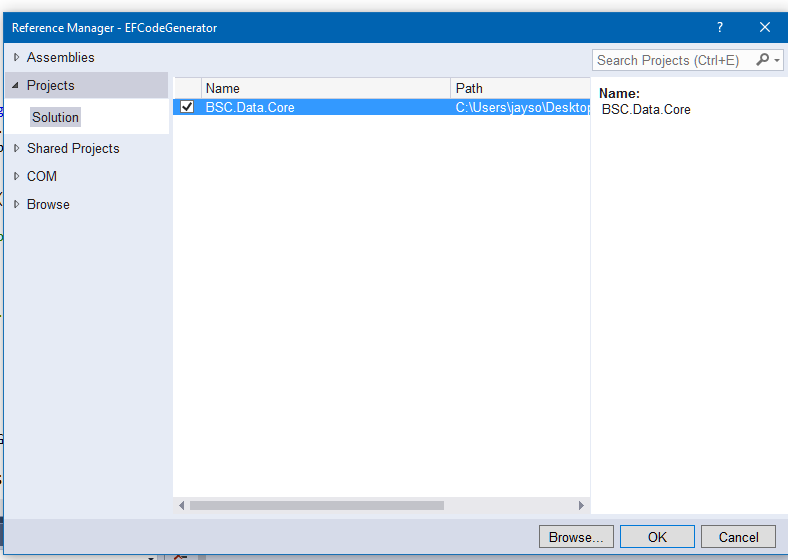
****

Note: If there is an error scaffolding the DB, please unload the EFCodeGenerator project first and reload it again after running the script.



1. Compile and build the solution.
2. Add a new reference to the EFCodeGenerator project with the created DAL dll.





1. Update the **Common.cs** code in the EFCodeGenerator project with the lines below:
2. Line 1

* Add a using statement for the DAL project (i.e. using BSC.Data.Core.Models;)

1. Line 18, 32 and 56

* Replace the DBContext keyword in the code with the name of the new context class (i.e. bscContext)
* Uncomment underlying lines of code for the DBContext

//Type type = typeof(DbContext);

//if (type != null)

//{

// \_defaultNamespace = type.Module.Name.Replace(".dll", string.Empty);

// \_modelsNamespace = type.Namespace;

// \_contextClass = type.FullName;

//}

Type type = typeof(bscContext);

if (type != null)

{

\_defaultNamespace = type.Module.Name.Replace(".dll", string.Empty);

\_modelsNamespace = type.Namespace;

\_contextClass = type.FullName;

}

1. Run and build the EFCodeGenerator.

