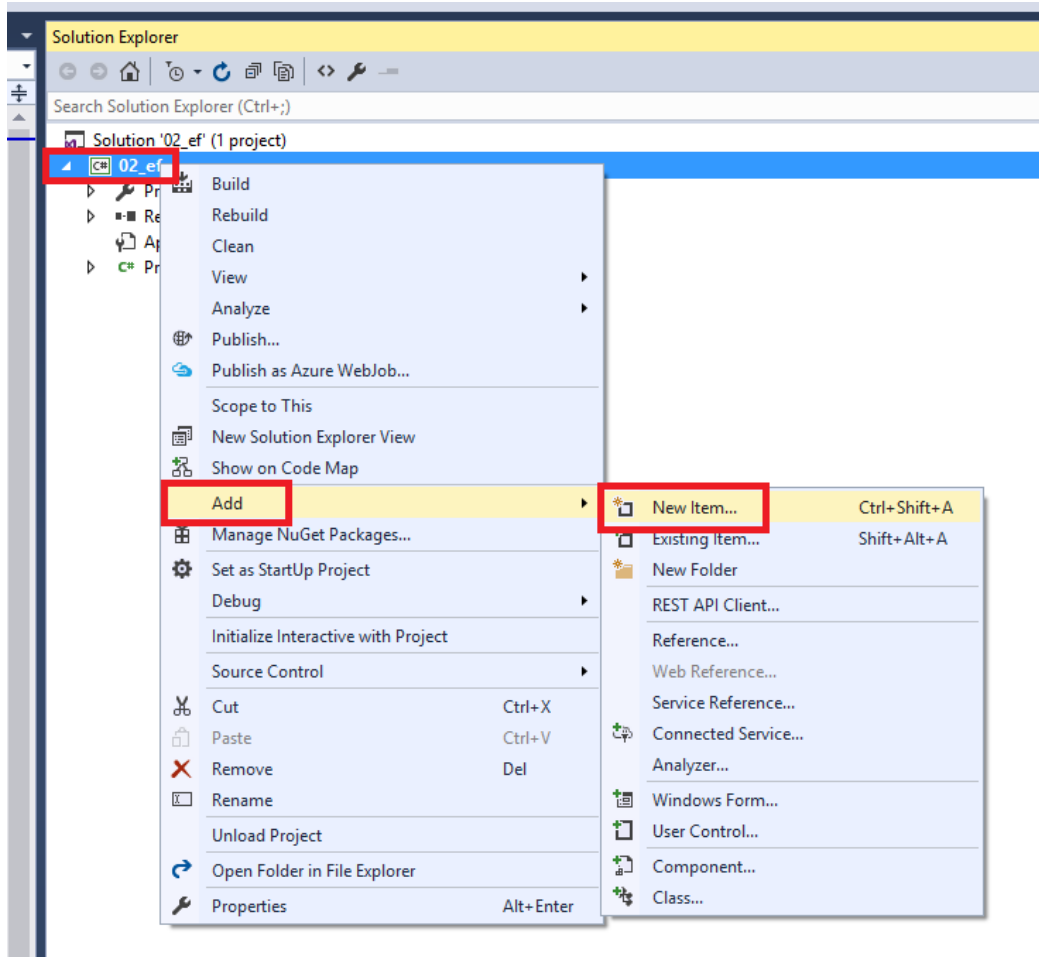
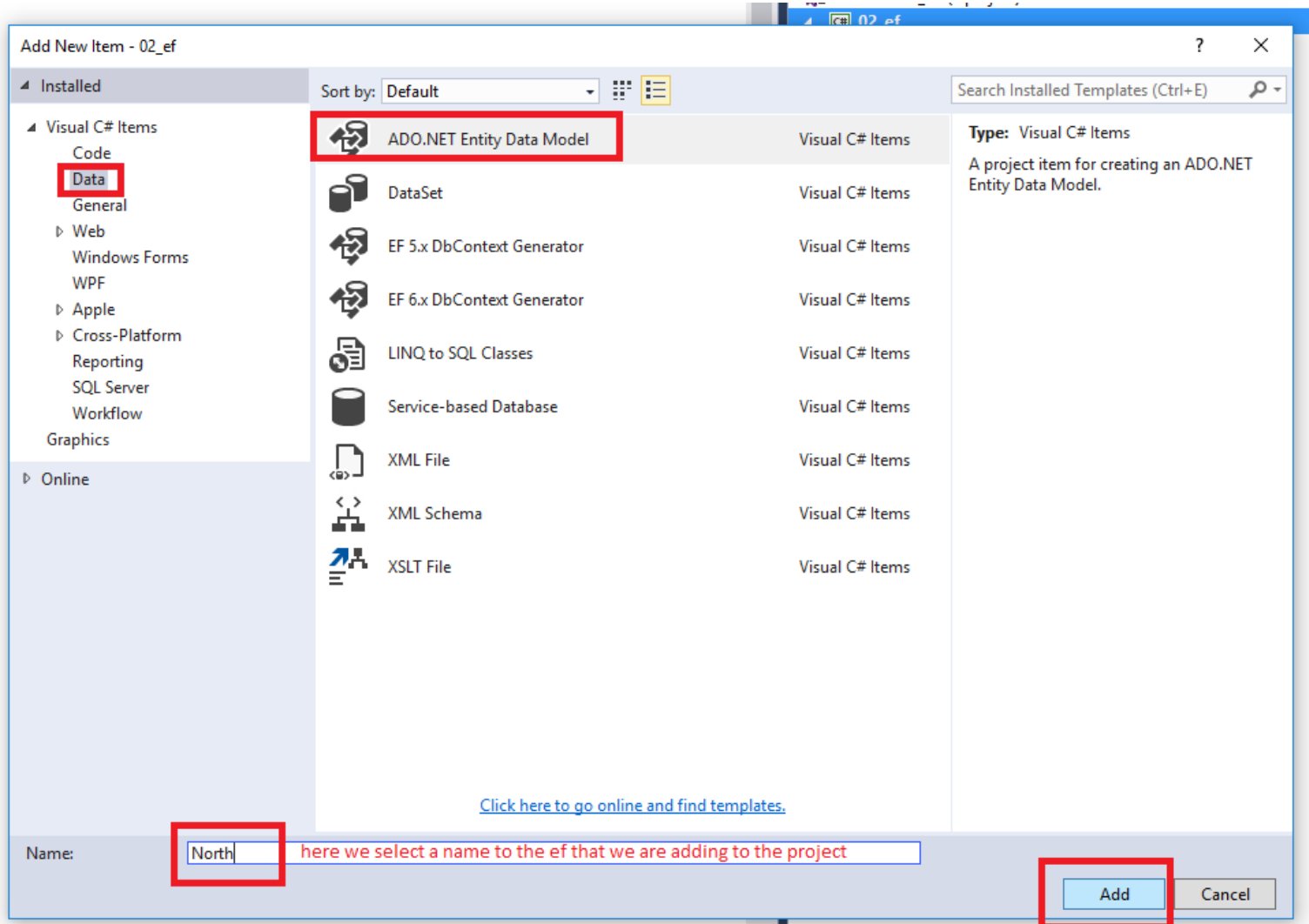

Entity framework

Adding to console application

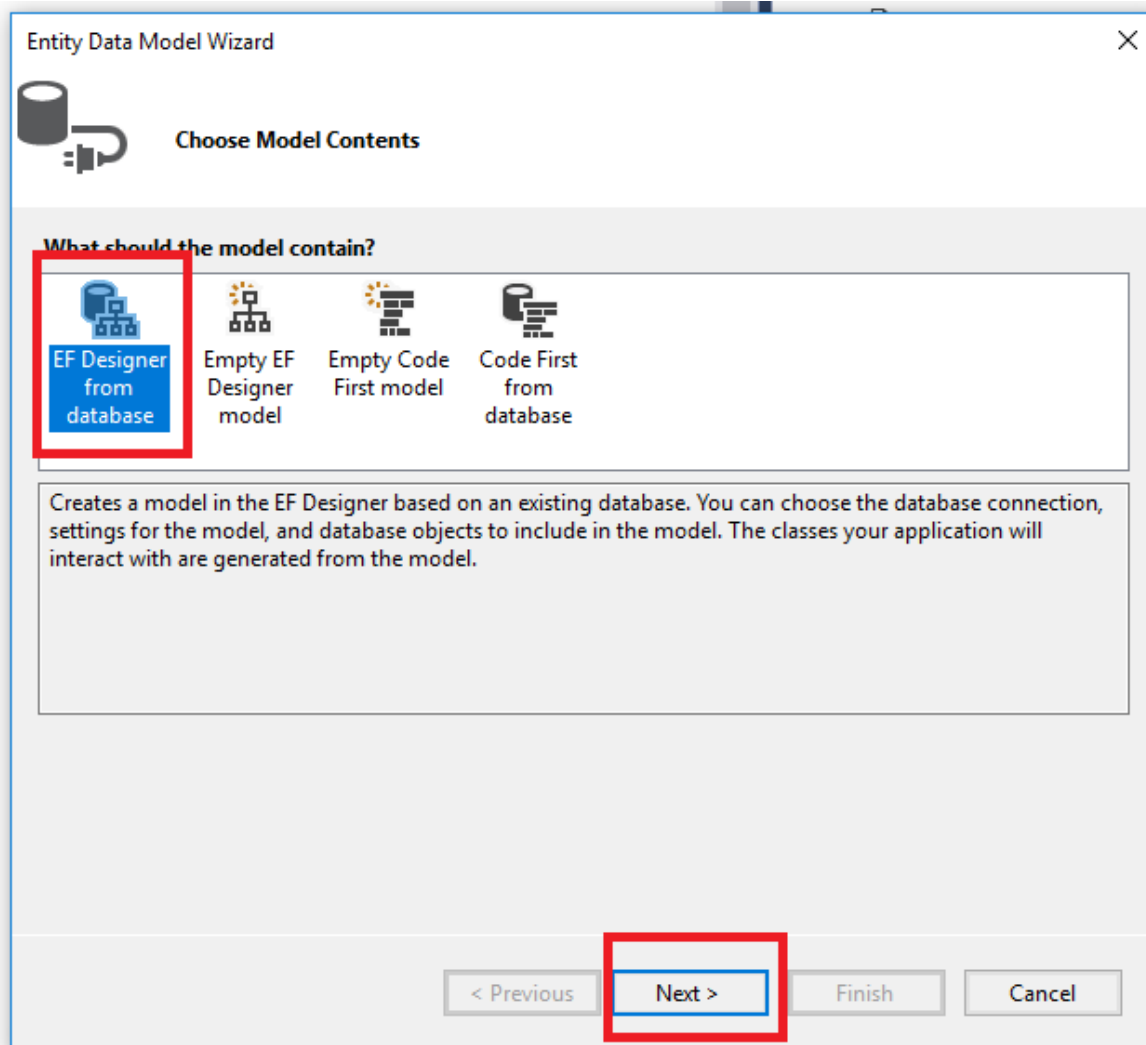
Step 1: Create a new console application and do the following steps:



Step 2

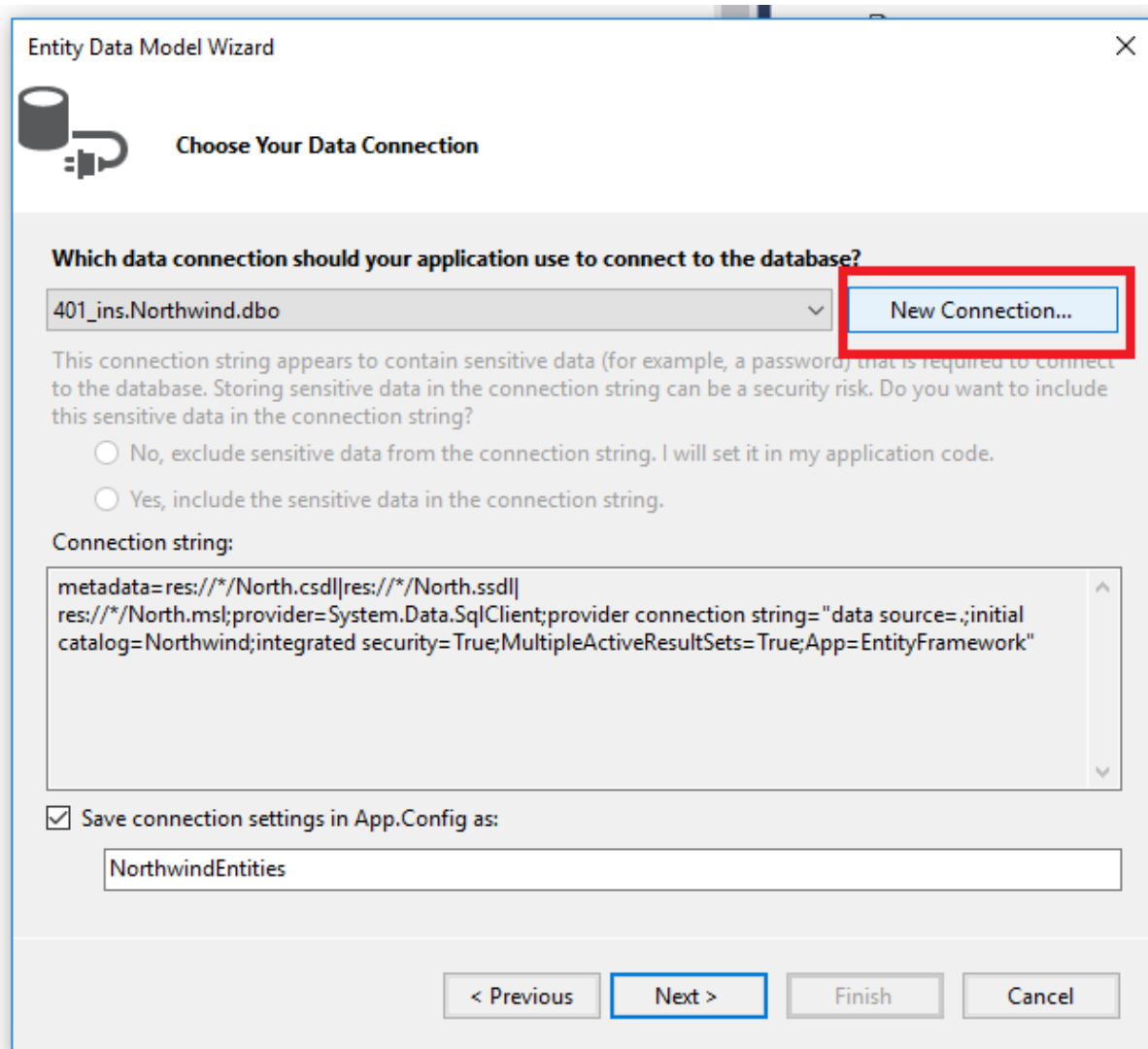


Step 3



Step 4

Create the connection string to the DB



The image shows the 'Entity Data Model Wizard' dialog box, specifically the 'Choose Your Data Connection' step. The title bar reads 'Entity Data Model Wizard' with a close button. Below the title bar is a database icon and the text 'Choose Your Data Connection'. The main question is 'Which data connection should your application use to connect to the database?'. A dropdown menu shows '401_ins.Northwind.dbo'. To the right of the dropdown is a button labeled 'New Connection...', which is highlighted with a red rectangle. Below this, a warning message states: 'This connection string appears to contain sensitive data (for example, a password) that is required to connect to the database. Storing sensitive data in the connection string can be a security risk. Do you want to include this sensitive data in the connection string?'. There are two radio buttons: 'No, exclude sensitive data from the connection string. I will set it in my application code.' (selected) and 'Yes, include the sensitive data in the connection string.'. Below the radio buttons is a section labeled 'Connection string:' with a text area containing the following text: `metadata=res://*/North.csdl|res://*/North.ssdl|res://*/North.msl;provider=System.Data.SqlClient;provider connection string="data source=.;initial catalog=Northwind;integrated security=True;MultipleActiveResultSets=True;App=EntityFramework"`. At the bottom left, there is a checked checkbox labeled 'Save connection settings in App.Config as:' followed by a text box containing 'NorthwindEntities'. At the bottom right, there are four buttons: '< Previous', 'Next >' (highlighted with a blue border), 'Finish', and 'Cancel'.

Entity Data Model Wizard

Choose Your Data Connection

Which data connection should your application use to connect to the database?

401_ins.Northwind.dbo

New Connection...

This connection string appears to contain sensitive data (for example, a password) that is required to connect to the database. Storing sensitive data in the connection string can be a security risk. Do you want to include this sensitive data in the connection string?

☐ No, exclude sensitive data from the connection string. I will set it in my application code.

☐ Yes, include the sensitive data in the connection string.

Connection string:

metadata=res://*/North.csdl|res://*/North.ssdl|
res://*/North.msl;provider=System.Data.SqlClient;provider connection string="data source=.;initial
catalog=Northwind;integrated security=True;MultipleActiveResultSets=True;App=EntityFramework"

☒ Save connection settings in App.Config as:

NorthwindEntities

< Previous Next > Finish Cancel

Connection Properties

Enter information to connect to the selected data source or click "Change" to choose a different data source and/or provider.

Data source:
Microsoft SQL Server (SqlClient) Change...

Server name:
. server name Refresh

Log on to the server

Authentication: Windows Authentication

User name:

Password:

☐ Save my password

Connect to a database

☒ Select or enter a database name:
Northwind database name

☐ Attach a database file:
 Browse...

Logical name:

Advanced...

Test Connection OK Cancel



Choose Your Data Connection

Which data connection should your application use to connect to the database?

401_ins.Northwind.dbo

New Connection...

This connection string appears to contain sensitive data (for example, a password) that is required to connect to the database. Storing sensitive data in the connection string can be a security risk. Do you want to include this sensitive data in the connection string?

- ☐ No, exclude sensitive data from the connection string. I will set it in my application code.
- ☐ Yes, include the sensitive data in the connection string.

Connection string:

```
metadata=res://*/North.csdl|res://*/North.ssdl|
res://*/North.msl;provider=System.Data.SqlClient;provider connection string="data source=.;initial
catalog=Northwind;integrated security=True;MultipleActiveResultSets=True;App=EntityFramework"
```

☒ Save connection settings in App.Config as:

NorthwindEntities

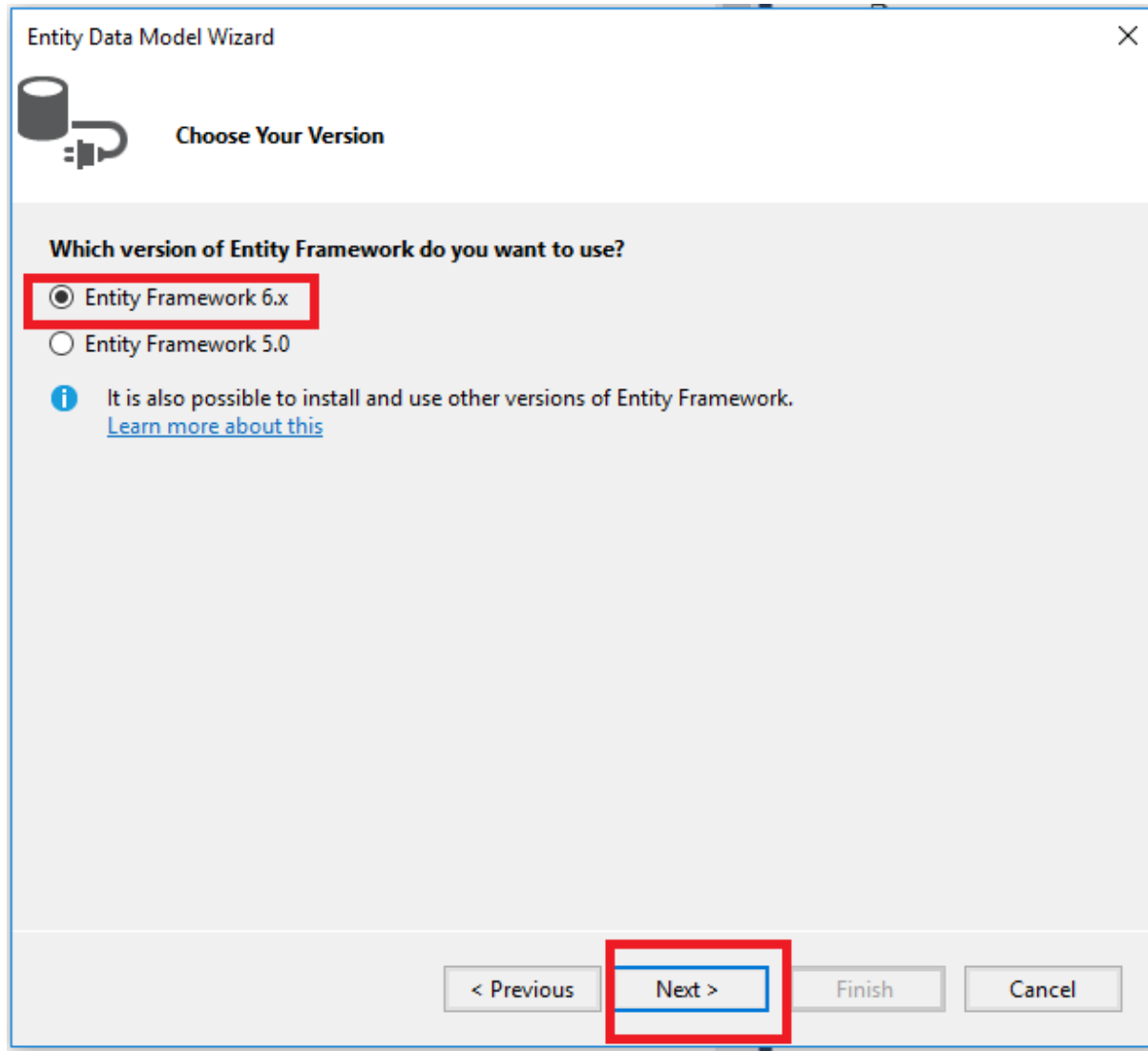
< Previous

Next >

Finish

Cancel

Step 5: select ef version



The image shows a Windows-style dialog box titled "Entity Data Model Wizard" with a close button (X) in the top right corner. Below the title bar is a header area containing a database icon and the text "Choose Your Version". The main content area has a question "Which version of Entity Framework do you want to use?" followed by two radio button options: "Entity Framework 6.x" (which is selected and highlighted with a red rectangle) and "Entity Framework 5.0". Below these options is an information icon (i) and a text block stating "It is also possible to install and use other versions of Entity Framework." with a blue hyperlink "Learn more about this". At the bottom of the dialog is a navigation bar with four buttons: "< Previous", "Next >" (highlighted with a red rectangle), "Finish", and "Cancel".

Entity Data Model Wizard

Choose Your Version

Which version of Entity Framework do you want to use?

☒ Entity Framework 6.x

☐ Entity Framework 5.0

i It is also possible to install and use other versions of Entity Framework.
[Learn more about this](#)

< Previous Next > Finish Cancel

Step 6:

Entity Data Model Wizard

Choose Your Database Objects and Settings

Which database objects do you want to include in your model?

☒ Tables

☐ VIEWS

☐ Stored Procedures and Functions

☒ Pluralize or singularize generated object names

☒ Include foreign key columns in the model

☒ Import selected stored procedures and functions into the entity model

Model Namespace:

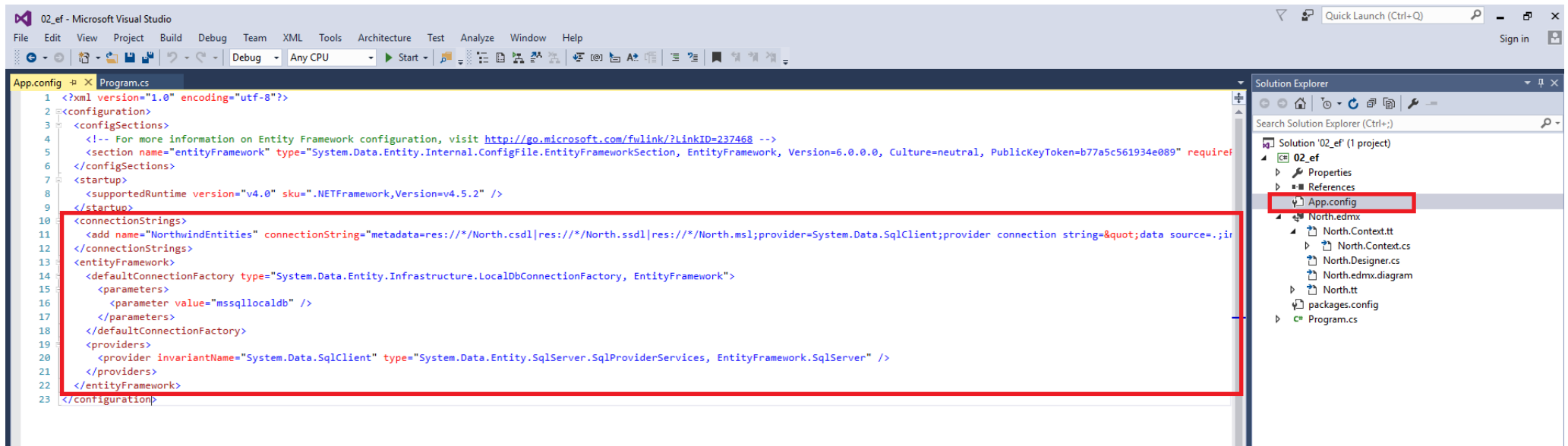
NorthwindModel

< Previous Next > **Finish** Cancel

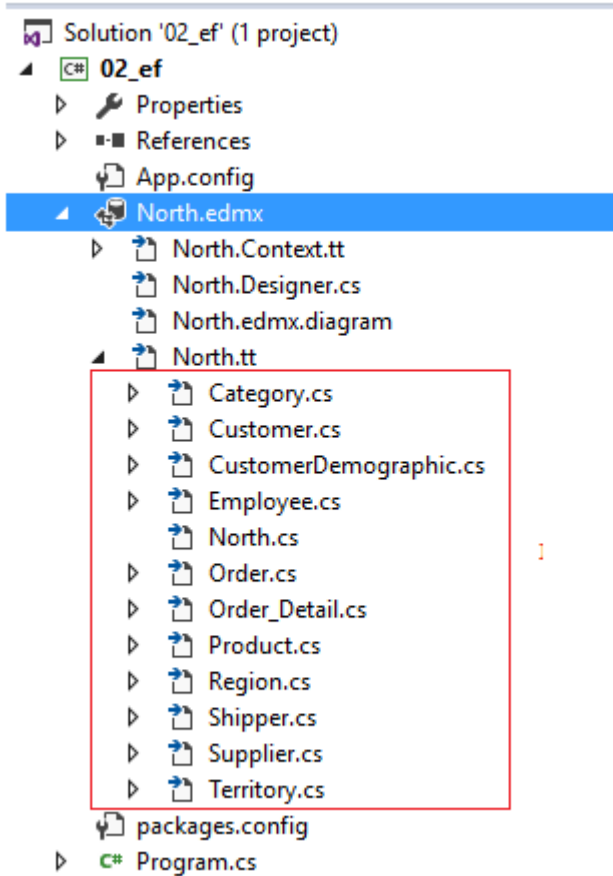
Entity framework

Added files by EF

After adding the ef to the console application, as we explained above, we get the connection string and ef config in the “app.config” file:

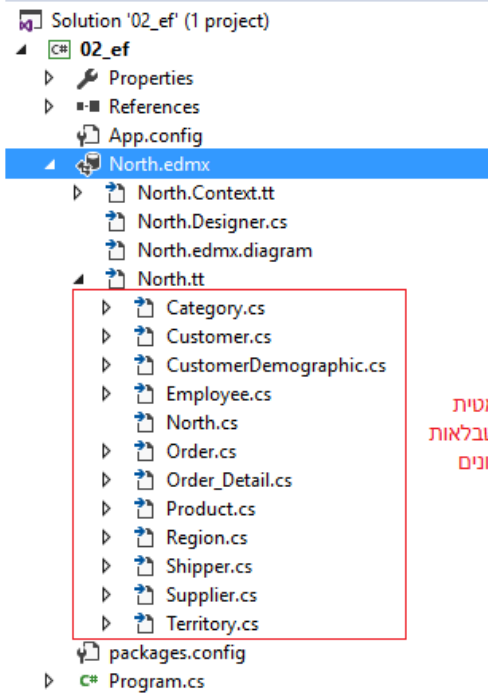


In addition, we get all the following content:



Bellow, we will explain each part of this files:

Class Model for each table in the DB

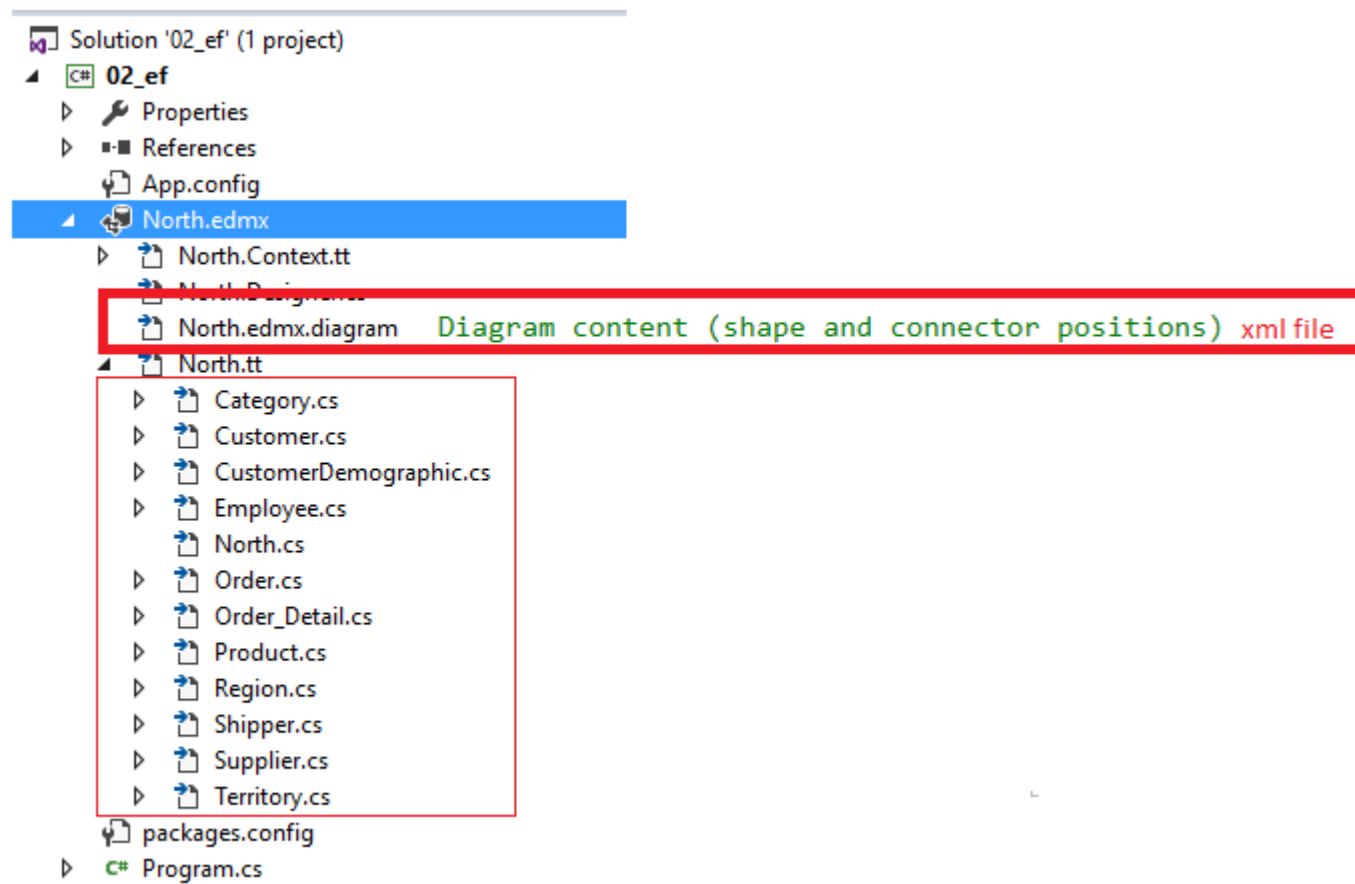


בחלק זה נוצרו אוטומטית
מחלקות עבור כל הטבלאות
שבחרנו מבסיס הנתונים



```
//-----  
// <auto-generated>  
//     This code was generated from a template.  
//  
//     Manual changes to this file may cause unexpected behavior in your application.  
//     Manual changes to this file will be overwritten if the code is regenerated.  
// </auto-generated>  
//-----
```

Diagram content



Class that represents the DB with all the tables (DbSet for each table):

The image shows a Visual Studio solution named '02_ef' (1 project). The project structure is as follows:

- 02_ef
 - Properties
 - References
 - App.config
 - North.edmx
 - North.Context.tt (highlighted with a red box)
 - North.Designer.cs
 - North.edmx.diagram
 - North.tt
 - Category.cs
 - Customer.cs
 - CustomerDemographic.cs
 - Employee.cs
 - North.cs
 - Order.cs
 - Order_Detail.cs
 - Product.cs
 - Region.cs
 - Shipper.cs
 - Supplier.cs
 - Territory.cs
 - packages.config
 - Program.cs

Red arrows indicate the flow from the 'North.Context.tt' file to the 'North.Context.cs' file, and then to the code block below.

```
namespace _02_ef
{
    using System;
    using System.Data.Entity;
    using System.Data.Entity.Infrastructure;

    public partial class NorthwindEntities : DbContext
    {
        public NorthwindEntities()
            : base("name=NorthwindEntities")
        {
        }

        protected override void OnModelCreating(DbModelBuilder modelBuilder)
        {
            throw new UnintentionalCodeFirstException();
        }

        public virtual DbSet<Category> Categories { get; set; }
        public virtual DbSet<CustomerDemographic> CustomerDemographics { get; set; }
        public virtual DbSet<Customer> Customers { get; set; }
        public virtual DbSet<Employee> Employees { get; set; }
        public virtual DbSet<Order_Detail> Order_Details { get; set; }
        public virtual DbSet<Order> Orders { get; set; }
        public virtual DbSet<Product> Products { get; set; }
        public virtual DbSet<Region> Regions { get; set; }
        public virtual DbSet<Shipper> Shippers { get; set; }
        public virtual DbSet<Supplier> Suppliers { get; set; }
        public virtual DbSet<Territory> Territories { get; set; }
    }
}
```

DB diagram

