

Contents

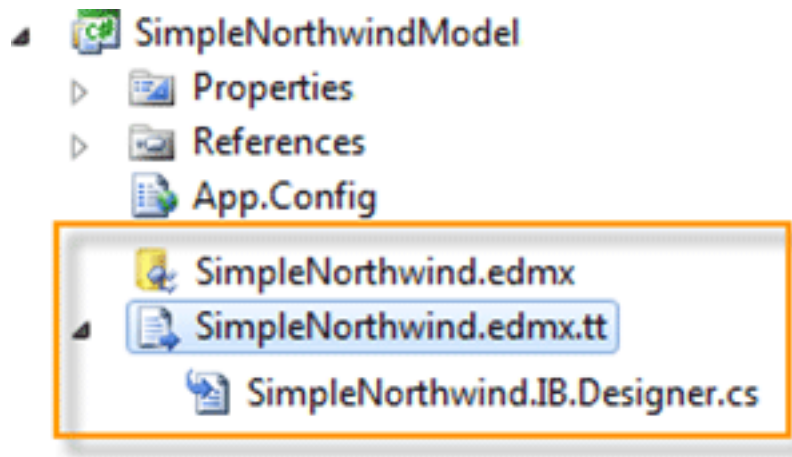
- [Code generation](#)
- [Generated source code file](#)
- [Customize the code generation template](#)

This topic describes how DevForce uses the **T4 code generator** and a **DevForce T4 template** to generate entity model source code file(s) based on an Entity Framework [Entity Data Model](#).

Code generation

DevForce generates entity model classes using the [T4 code generator](#) built into Visual Studio whenever you make a change to the Entity Framework [EDMX](#) file. The T4 generator is directed by the [DevForce T4 code generation template](#) which reads the EDMX and emits entity model source code file(s).

Look at your project in the Visual Studio **Solution Explorer** after adding an [Entity Framework](#) model.



Notice the file with a .tt extension that echoes the name of the model.

Expand the .tt file tree-view node to see the nested source code file. There is only one such file in this small example, the file ending ".Designer.cs". There could be several such files if the model were unusually large.

This source code file is regenerated whenever you modify and save the EDMX file. You can also regenerate it directly by right-clicking the .tt template file and selecting **Run Custom Tool**.

Generated source code file

A full exploration of the generated source code is covered in the [development section](#) but it won't hurt to peek and point out a few highlights. In the following snapshot you'll find three kinds of classes.

```

namespace SimpleNorthwindModel {
    /// <summary>
    /// The domain-specific EntityManager for your domain model.
    /// </summary>
    [IbEm.DataSourceKeyName(@"NorthwindEntities")]
    public partial class NorthwindManager : IbEm.EntityManager {
        Constructors
        DefaultManager
        EntityQueries
    }
    Category class
    Customer class
    Supplier class
    EntityRelations
}

```

At the top is a [model-specific EntityManager](#) component. The *EntityManager* is the most important component in all of DevForce and you will get to know it well.

Below that are the [generated entity classes](#), one after another, each one generated from the conceptual entity defined in your [edm](#). The *Customer* class is typical:

```

/// <summary>The auto-generated Customer class. </summary>
[DataContract(IsReference=true)]
[KnownType(typeof(Order))]
[IbEm.DataSourceKeyName(@"NorthwindEntities")]
[IbEm.DefaultEntitySetName(@"NorthwindEntities.Customers")]
public partial class Customer : IbEm.Entity {

    /// <summary>Returns the property path for the given expression. </summary> ...
    public static string PathFor(Expression<System.Func<Customer, object>> expr) ...

    Properties
    Navigation properties
    EntityProperty definitions
    EntityPropertyNames
}

```

Finally, at the bottom, is the *EntityRelations* class that defines every relationship between every entity in the model.

Customize the code generation template

In general you [customize the generated classes](#) by adding more code to the model. You can trim back some of the emitted code by adjusting the generator control properties within the [EDM Designer](#).

If these options aren't sufficient for your needs, you can [customize the code generation template](#) yourself.