**Customizing Code First to an Existing Database**

**Customizing Code First to an Existing Database**

This topic covers how to customize the code that is generated when you use the Entity Framework Tools for Visual Studio to reverse engineer a model from an existing database. If you aren't already familiar with this functionality, see [Code First to an Existing Database](http://msdn.microsoft.com/en-in/data/jj200620).

|  |  |
| --- | --- |
| http://i.msdn.microsoft.com/ee402630.Numeral1_lg(en-us,MSDN.10).png | **Add the default templates to your project** |
|  | If you include a set of the code generation templates in your project EF will use these in favor of the default built-in templates. There are NuGet packages that allow you to add the default templates to your project as a starting point.   * **EntityFramework.CodeTemplates.CSharp** adds the default C# code generation templates. * **EntityFramework.CodeTemplates.VisualBasic** adds the default VB.NET code generation templates.   To install the templates to your project:   1. Right-click on your project and select **Manage NuGet Packages...** 2. Under the **Online** tab select **EntityFramework.CodeTemplates.CSharp** or **EntityFramework.CodeTemplates.VisualBasic** and click **Install**   Alternatively, you can install the templates from Package Manager Console:  PM> Install-Package EntityFramework.CodeTemplates.CSharp |
| http://i.msdn.microsoft.com/ee402630.Numeral2_lg(en-us,MSDN.10).png | **Customize the templates** |
|  | The NuGet package will have installed a **Context** and **EntityType** template under the **CodeTemplates\EFModelFromDatabase\** folder in your project. These are [T4 templates](http://msdn.microsoft.com/en-us/library/bb126445.aspx) that can be modified to affect the generated code.  As an example, entity types initialize their collection navigation properties with a HashSet<T>, but you may want to use List<T> instead. To do this, you would locate the following line of code in the EntityType template.  **<#= code.Property(collectionProperty) #> = new HashSet<<#= code.Type(collectionProperty.ToEndMember.GetEntityType()) #>>();**  You could then replace it with the following code to initialize the property with a List<T> instead.  **<#= code.Property(collectionProperty) #> = new List<<#= code.Type(collectionProperty.ToEndMember.GetEntityType()) #>>();** |
| http://i.msdn.microsoft.com/ee402630.Numeral3_lg(en-us,MSDN.10).png | **Run the reverse engineer process** |
|  | Once you have modified the templates, you can run the reverse engineer process to generate the code. For more details on this process, see see [Code First to an Existing Database](http://msdn.microsoft.com/en-in/data/jj200620).  If you want to make further modifications to the templates, you can do so and then re-run the reverse engineer process. The wizard will simply overwrite the previously generated files. |