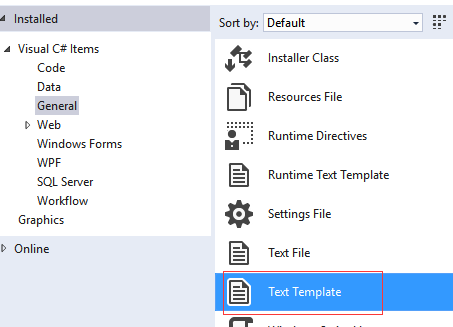
# T4模板

T4，即4个T开头的英文字母组合：Text Template Transformation Toolkit。

T4文本模板，即一种自定义规则的代码生成器。根据业务模型可生成任何形式的文本文件或供程序调用的字符串。（模型以适合于应用程序域的形式包含信息，并且可以在应用程序的生存期更改）

详情查看：<http://www.cnblogs.com/heyuquan/archive/2012/07/26/2610959.html>

## 创建模板文件



文件如下：



修改T4.tt。只需要修改最后一行，表示生成文件的后缀。

|  |
| --- |
| <#@ template debug="false" hostspecific="false" language="C#" #>  <#@ assembly name="System.Core" #>  <#@ import namespace="System.Linq" #>  <#@ import namespace="System.Text" #>  <#@ import namespace="System.Collections.Generic" #>  <#@ output extension=".cs" #> |

结果:



## 编写模板

|  |
| --- |
| <#@ template debug="false" hostspecific="false" language="C#" #>  <#@ assembly name="System.Core" #>  <#@ import namespace="System.Linq" #>  <#@ import namespace="System.Text" #>  <#@ import namespace="System.Collections.Generic" #>  <#@ output extension=".cs" #>  using System;  using System.Collections.Generic;  using System.Linq;  using System.Text;  using System.Threading.Tasks;  namespace T4模板  {  public partial class Person  {  <#  for(int i=0; i<5;i++)  {#>  int num<#=i#> { get; set; }  <#}#>  }  } |

<# #>中是T4的控制块

有三种类型的控制块，根据其左括号对它们进行区分：

1. <# 标准控制块 #> 可以包含语句。

2. <#= 表达式控制块 #> 将一个可以计算为字符串的表达式括起来,用于提供要写入“输出”文件的字符串的代码。

3. <#+ 类功能控制块 #> 可以使用类功能控制块向文本模板添加方法、属性、字段甚至是嵌套类。必须作为文件中的最后一个块显示，或者用<#@ include #>引入外部文件。

## T4与edmx结合