

关于T4模板的使用

1、使用场景

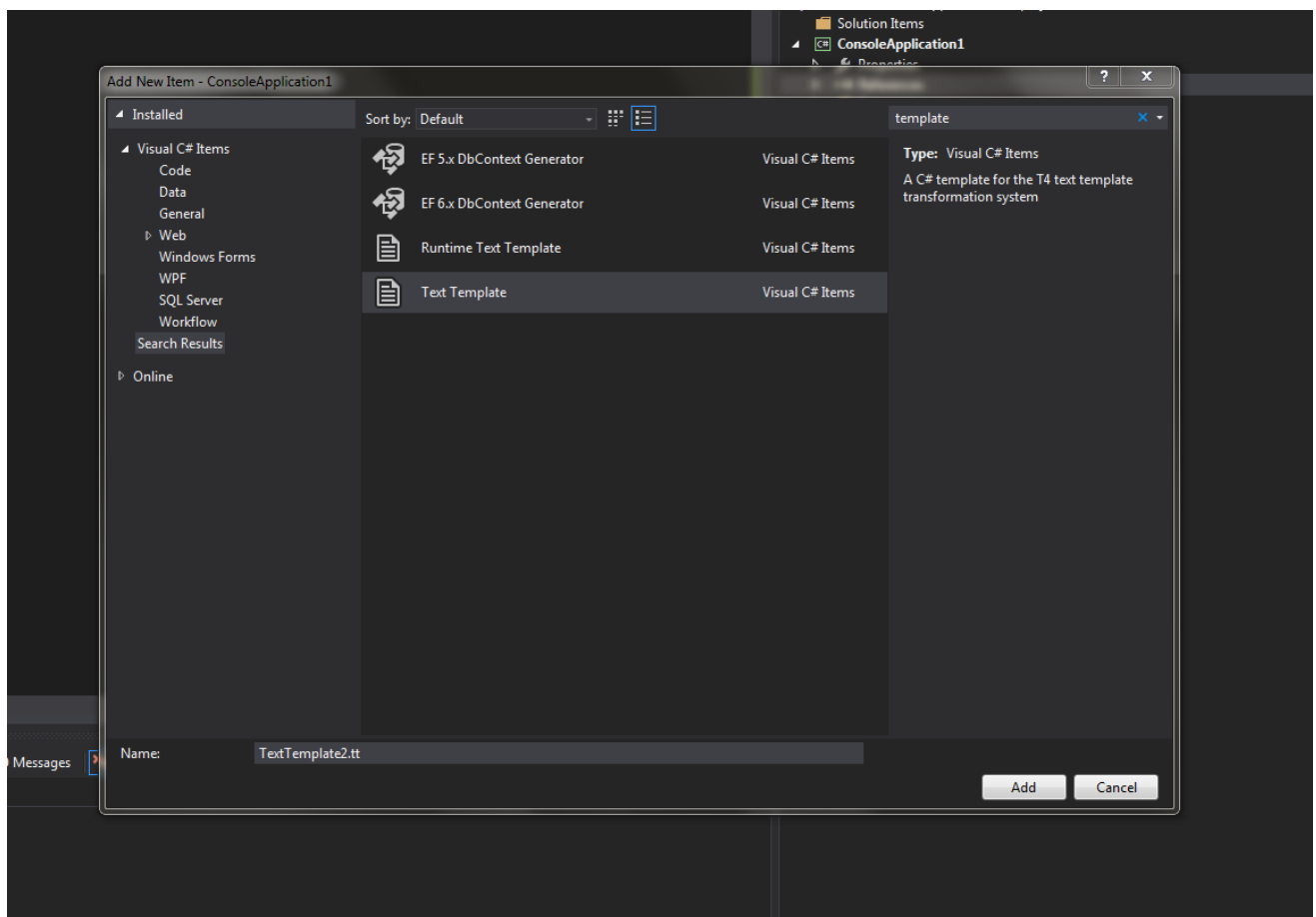
当编写epos类，每个类有许多成员变量，以及对应的属性，依次之间格式统一，手动编写工作量大并且易出错。

于是想到可以采用模板的方式编写一部分代码，合理T4模板可能可以减少一部分工作量。以后可以考虑一下。

2、概念

T4模板是vs2008中开始使用的代码生成引擎。在visual studio中，T4文本模板是由一些文本块和控制逻辑组成的混合模板，它可以生成文本文件。

3、创建T4模板



4、demo

控制逻辑代码块说明

1. <# #>用于存放普通代码块
2. <#=#>存放变量
3. <#+#>存放方法

- 从test.txt中获取到数据，存放在字典中
- 将字典的数据，按照指定格式输出在文本中

```

1  <#@ template debug="false" hostspecific="false" language="C#" #>
2  <#@ assembly name="System.Core" #>
3  <#@ import namespace="System.Linq" #>
4  <#@ import namespace="System.Text" #>
5  <#@ import namespace="System.IO" #>
6  <#@ import namespace="System.Collections.Generic" #>
7  <#@ output extension=".cs" #>
8  <#
9      string path =
10         @"C:\Users\ST_123\ConsoleApplication1\ConsoleApplication1\test.txt";
11         FileStream fs = File.OpenRead(path);
12         StreamReader sr = new StreamReader(fs);
13         string line = string.Empty;
14         Dictionary<string, string> dic = new Dictionary<string, string>();
15         while ((line = sr.ReadLine()) != null)
16         {
17             string[] strings = line.Split(':');
18             string key = strings[0].Trim();
19             string value = strings[1].Trim();
20             dic.Add(key, value);
21         }
22         sr.Close();
23         fs.Close();
24     #>
25     using Siemens.SMART200.TestHelperPlus;
26     using System;
27     using System.Collections.Generic;
28     using System.Linq;
29     using System.Text;
30     using System.Threading;
31     using System.Threading.Tasks;
32     using System.Xml;
33     class Test
34     {
35         <# foreach(KeyValuePair<string,string> item in dic){#>
36         private int <#=item.Key#>;
37         protected int <#=upperFirstLetter(item.Key)#>
38         {
39             get
40             {
41                 return <#=item.Key#>;
42             }
43             set
44             {
45                 <#=upperFirstLetter(item.Key)#> = value;
46             }
47         }
48     }
49     }
50     <#+
51     private string upperFirstLetter(string input)
52     {

```

```
53         return input.Substring(0, 1).ToUpper() + input.Substring(1);
54     }
55     #>
```

生成文件如下

```
1  using Siemens.SMART200.TestHelperPlus;
2  using System;
3  using System.Collections.Generic;
4  using System.Linq;
5  using System.Text;
6  using System.Threading;
7  using System.Threading.Tasks;
8  using System.Xml;
9  class Test
10 {
11     private int speedSp;
12     protected int SpeedSp
13     {
14         get
15         {
16             return speedSp;
17         }
18         set
19         {
20             SpeedSp = value;
21         }
22     }
23     private int configWord;
24     protected int ConfigWord
25     {
26         get
27         {
28             return configWord;
29         }
30         set
31         {
32             ConfigWord = value;
33         }
34     }
35     private int ackError;
36     protected int AckError
37     {
38         get
39         {
40             return ackError;
41         }
42         set
43         {
44             AckError = value;
45         }
46     }
47 }
```

```
48     }
49 }
50 private int rerterd;
51 protected int Rerterd
52 {
53     get
54     {
55         return rerterd;
56     }
57
58     set
59     {
60         Rerterd = value;
61     }
62 }
63 }
```

string path = @"C:\Users\ST_123\ConsoleApplication1\ConsoleApplication1\test.txt";

```
1 speedSp:12
2 configword:45
3 ackError:45
4 rerterd:4545
```

5、网站

<https://www.cnblogs.com/zeje/p/5248340.html>