## 1、建立项目目录

]]) D:\EFStudy\自跟	見踪实体演	陈 🔻 😽	搜索自跟踪实	史体演练				
编辑(E) 查看(V)	工具(I)	帮助( <u>H</u> )						
包含到库中 ▼	共享 ▼	刻录	新建文件夹				= -	
Ę	<b>^</b>	名称	<u>^</u>		修改日期	类型		大小
;	=	🌗 out			2015/8/13 21:13	文件夹		
i		똃 EngineEF	.sln		2015/8/13 21:16	Microsoft	Visual	

可以先建立一个 out 子目录作为项目输出,并建立一个空白的 sln 文件。

2、在 EngineEF.sln 下添加一个 C#类库工程,起名 DAL,

	Visual C#
WPF Ribbon Application	Visual C#
WPF 应用程序	Visual C#
空 控制台应用程序	Visual C#
美库	Visual C#
CF 服务应用程序	Visual C#
Excel 2010 工作簿	Visual C#

3、在 DAL 中添加 EF5.0 支持。 启动"程序包管理控制台",注意选择默认项目"DAL",然 后执行:

```
PM> Install-Package EntityFramework -Version 5.0.0
正在安装"EntityFramework 5.0.0"。
您正在从 Microsoft 下载 EntityFramework,有关此程序包的许可协议在 http://go.m:
即构成您接受其许可协议。如果您不接受这些许可协议,请从您的设备中删除相关组件。
已成功安装"EntityFramework 5.0.0"。
正在将"EntityFramework 5.0.0"添加到 DAL。
已成功将"EntityFramework 5.0.0"添加到 DAL。
```

4、在 DAL 中添加 "ADO.NET 实体数据模型":

删除创建工程时 vs 添加的 Class1.cs。

已安装的模板	排序依据: 默认值	• • • • • • • • • • • • • • • • • • • •	搜索已安装的模板 👂
◢ Visual C# 项 Web	数据库单元测试	Visual C# 项	<b>类型:</b> Visual C# 项 用于创建 ADO.NET 实体数据模型的项目
Windows Forms WPF	ADO.NET 实体数据模型	Visual C# 项	项。
常规 代码	LINQ to SQL 类	Visual C# 项	5
数据 Reporting	マ の の の の の の の の の の の の の の の の の の の	<mark>Visual</mark> C# 项	
Workflow 联机模板		Visual C# 项	
	XSLT 文件	Visual C# 项	
	本地数据库	Visual C# 项	
名称(N): Model1.ec	Imx	12 1.50.7	
			添加(A) 取消

点击"添加",并按照提示选择 sqlce 数据库,如:

连接属性 ? 🛛
輸入信息以连接到选定的数据源,或单击"更改"选择另一个数据 源和/或提供程序。
数据源( <u>S</u> ):
Microsoft SQL Server Compact 3.5 (用于 N 更改( <u>C</u> )
数据源
④ 我的电脑(M)
◎ ActiveSync 连接的设备( <u>A</u> )
连接属性
数据库( <u>D</u> ):
D:\EFStudy\自跟踪实体演练\out\1.sdf
创建(E) 浏览(B)
密码(P):
保存密码(S)
高级⊻
测试连接(1) 确定 取消

实体数据模型向导	8 23
<b>选</b>	经的数据连接
您的应用程序应	使用哪个数据连接与数据库进行连接(W)?
1.sdf	▼ 新建连接(C)
此连接字符串似 险。是否要在连	乎包含连接数据库所需的敏感数据(例如密码)。在连接字符串中存储敏感数据可能有安全风 接字符串中加入这些敏感数据?
🔘 否 , 从道	接字符串中排除敏感数据。我将在应用程序代码中设置此数据(E)。
◎ 是,在	Microsoft Visual Studio
实体连接字符串	
metadata=res res://*/Model Source=D:\EF	所选连接使用的本地数据文件不在当前项目中。是否要将该文件复制到     项目中并修改连接?
	是(Y) 否(N)
☑ 将 App.Conf	ig 中的实体连接设置另存为(S):
Entities	
	< 上一步(P) 下一步(N) > 完成(F) 取消

完成后, 解决方案如图所示:

₩	决方案 "EngineEF" (1 个项目) DAL
Þ	Properties
Þ	画 引用
	间 1.sdf
	🚯 App.config
. a	🍕 Model1.edmx
	Model1.Designer.cs
	🕒 packages.config

删除 Model1.edmx 附属的 Model1.Designer.cs.

5、添加另一个类库项目 Model,删除多余的 class1.cs

```
6、在 DAL 项目中添加 tt 文件:
```

在 Model1.edmx 的设计视图空白处右键点击, 在菜单选择[添加代码生成项...], 点击[代码],

## 选择[EF 5.x DbContext Generator]:

添加新現 - DAL				1
已安装的模板		林序纹描: <b>默认道 •</b>		推索 已安美的楼街 🔰
<ul> <li>Visual C# II Web Window WPF&lt;</li> <li>常規 代码 設備 Reportin Workfle</li> </ul>	t ns Forms ng we	ADO.NET EntityObject 生成器 ADO.NET 自動時來は生成器 。 EF 5.x DbContext Generator	Visual C# 頃 Visual C# 頃 Visual C# 項	業發: Visual C# 項 A project item to generate a strongly- typed DioContext class and persistence ignorant entity classes. Use this template when working with an EF 5.x release.
名称(N);	Model1.tt			
				將加(A) 取論。
注意可以	忽略随后的警	警告提示:		
安全警告	-			? 🔀

运行此文本模板可能损害您的计算机。	如果此文本模板的获取来源不受信任,请不要运行它。
单击"确定"运行该模板。 单击"取消"停止该过程。	
□ 不再显示此消息(D)	
	确定 取消(C)

这将添加 Model1.Context.tt 和 Model1.tt, 删除二者生成的.cs 文件。

**7**、从 DAL 中"排除"(而不是删除)Model1.edmx 和 Model1.tt; 在 Model 项目中,添加对这 两个文件的文件链接:

0	解	决方案'EngineEF' (2 个项目)
4	C#	DAL
	Þ	Properties
	Þ	回 引用
		📋 1.sdf
		🚯 App.config
		📑 Model1.Context.tt
		packages.config
4	C#	Model
	Þ	Properties
	Þ	🔄 引用
	Þ	Model1.edmx
	⊳	Model1.tt

8、 点击 Model 项目中的 Model1.edmx,属性显示如下:

Model1.edmx 文件属性	-
<mark>€≣</mark> 2↓ 🖾	
▲ 高级	
复制到輸出目录	不复制
生成操作	EntityDeploy
自定义工具	EntityModelCodeGenerator
自定义工具命名空间	
▲ 杂项	
完整路径	D:\EFStudy\自跟踪实体演练\DAL\Mod
文件名	Model1.edmx

将[生成操作]保持为 "EntityDeploy",而将[自定义工具]修改为空,[自定义工具命名空间]修改为 "Model" (或其他名字,只要与类库 Model 一致即可)。

Mo	Model1.edmx 文件属性 •				
•	<b>2</b> ↓				
⊿	高级				
	复制到输出目录	不复制			
	生成操作	EntityDeploy			
	自定义工具				
	自定义工具命名空间	Model			

9、将 DAL 项目下的 Model1.Context.tt 和 Model 项目下的 Model1.tt 的[自定义工具命名空间] 都修改为 "Model"。

Mo	Model1.Context.tt 文件属性				
⊿	高级				
	复制到输出目录	不复制			
	生成操作	无			
	自定义工具	TextTemplatingFileGenerator			
	自定义工具命名空间	Model			

Model1.tt 文件属性		
⊿	级	
	复制到输出目录	不复制
	生成操作	无
	自定义工具	TextTemplatingFileGenerator
	自定义工具命名空间	Model
	复制到输出目录 生成操作 自定义工具 自定义工具命名空间	不复制 无 TextTemplatingFileGenerator Model

- 10、设置 DAL 项目引用 Model 项目。随后编译两个项目。
- 11、建立一个 WPF 项目测试刚才生成的两个 DLL:
- 11.1 创建 WPF 工程

11.2 将三个工程的输出目录都设置为..\out,

11.3 WPF 工程添加对 EF5.0 的支持,参见第3节,注意选择"默认项目"为 WPF 工程。

11.4 WPF 工程添加对 DAL 和 Model 的引用

11.5 将 DAL 项目中 App.config 中的连接字符串

<connectionStrings>

<add name="Entities"

connectionString="metadata=res://\*/Model1.csdl|res://\*/Model1.ssdl|res://\*/Model1.msl;provider=System.D ata.SqlServerCe.3.5;provider connection string="Data Source=|DataDirectory|\1.sdf""

providerName="System.Data.EntityClient" />

</connectionStrings>

拷贝到 WPF 工程的 App.config 中。这个很重要! 连接字符串放在 DLL 中是无效的,必须放 到 EXE 文件的 App.config!!!

**11.6** 添加一个 DataGrid 到界面中:

<Grid>

<DataGrid x:Name="dg1" />

</Grid>

11.7 在 cs 中填充数据:

11.8 运行下 😊