**[学习ASP.NET MVC(八)——“Code First Migrations ”工具](http://www.cnblogs.com/chillsrc/p/4099407.html)**

        在本篇文章中，我们学习如何使用实体框架的“Code First Migrations ”（也称为代码先行功能）工具，使用其中的“迁移”功能对模型类进行一些修改，同时同步更新对应数据库的表结构。

        默认情况下，当您使用实体框架的“Code First Migrations ”工具，实体框架会自动创建一个数据库。“代码先行功能”首先会添加一个新表到数据库中，以便跟踪数据库的架构是否与模型类同步。如果它们不同步，实体框架会抛出一个错误。这样开发人员在开发时就能更容易地追查问题出在什么地方，而其他的开发方式就只能在运行时通过模糊的错误信息查找什么地方出错了。

**一、设置代码优先迁移的模型更改**

         如果您使用的是Visual Studio 2012，从解决方案资源管理器中双击Test.mdf文件，会打开数据库管理工具。其中Visual Studio Express For Web版本的VS将显示数据库资源管理器，Visual Studio 2012其他版本会显示服务器资源管理器。如果您使用的是Visual Studio 2010中，使用SQL Server对象资源管理器。   
         1）在服务器资源管理器中—》“数据连接”中（数据库管理工具、数据库浏览器、服务器资源管理器或SQL Server对象资源管理器），右键单击developer\sqlexpress.Test.dbo，然后选择删除。如下图1、图2。图1是删除连接，图2是删除数据库。

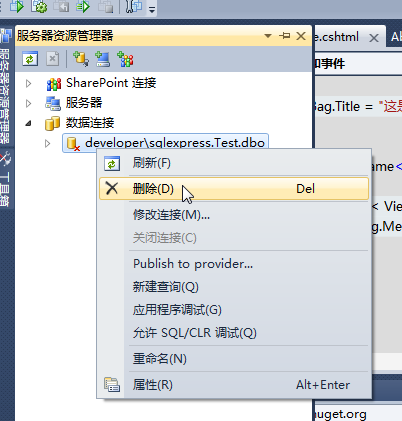
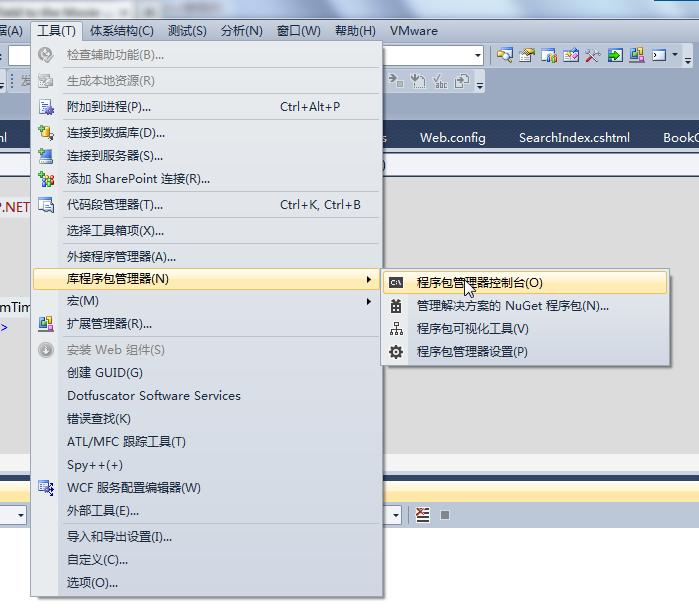


 图1

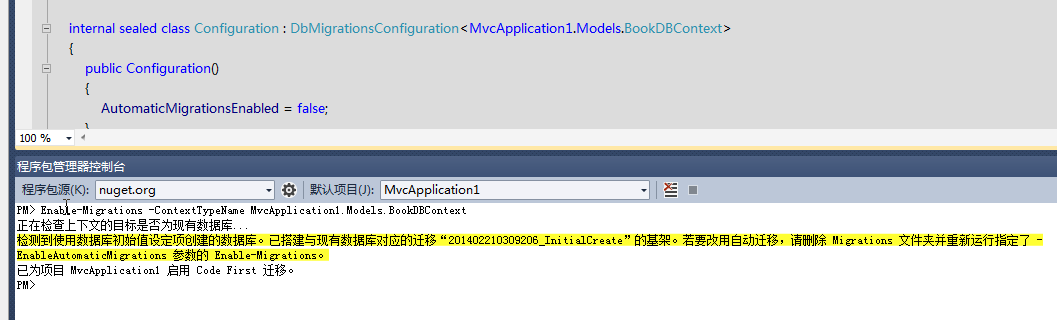


图2

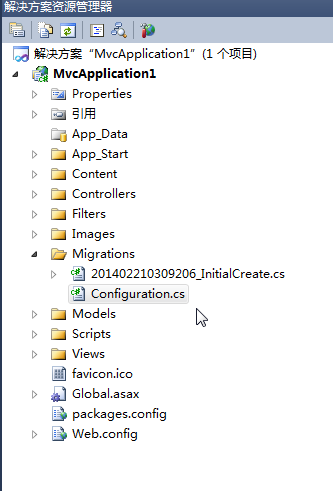
      2）执行菜单--》工具--》生成--》生成解决方案，以确保没有错误。   
       3）从菜单--》工具菜单中，单击库程序包管理器--》程序包管理器控制台。如下图。



            4）在Visual Studio的下方窗体中的“程序包管理器控制台”窗口的“>”提示符下输入“Enable-Migrations -ContextTypeName MvcApplication1.Models.BookDBContext”。如下图。



          5）在执行完Enable-Migrations命令之后（如上图所示），会在项目中创建一个新的文件夹“Migrations”， 在这个文件夹中创建一个Configuration.cs文件。如下图。



           6）在Visual Studio中打开Configuration.cs文件。用下面的代码替换Configuration.cs文件的Send方法：

[复制代码](javascript:void(0);)

namespace MvcApplication1.Migrations

{

using System;

using System.Data.Entity;

using System.Data.Entity.Migrations;

using System.Linq;

using MvcApplication1.Models;

internal sealed class Configuration : DbMigrationsConfiguration<MvcApplication1.Models.BookDBContext>

{

public Configuration()

{

AutomaticMigrationsEnabled = false;

}

protected override void Seed(MvcApplication1.Models.BookDBContext context)

{

// This method will be called after migrating to the latest version.

// You can use the DbSet<T>.AddOrUpdate() helper extension method

// to avoid creating duplicate seed data. E.g.

//

// context.People.AddOrUpdate(

// p => p.FullName,

// new Person { FullName = "Andrew Peters" },

// new Person { FullName = "Brice Lambson" },

// new Person { FullName = "Rowan Miller" }

// );

//

context.Books.AddOrUpdate(i => i.BookID,

new Book

{

Name = "When Harry Met Sally",

PublishDate = DateTime.Parse("1989-1-11"),

Category = "MS",

AuthorID=1,

Numberofcopies=12,

Price = 7.99M

},

new Book

{

Name = "Ghostbusters ",

PublishDate = DateTime.Parse("1984-3-13"),

Category = "MS",

AuthorID=1,

Numberofcopies=112,

Price = 8.99M

},

new Book

{

Name = "DB 2",

PublishDate = DateTime.Parse("1986-2-23"),

Category = "IBM",

AuthorID=1,

Numberofcopies=22,

Price = 9.99M,

}

);

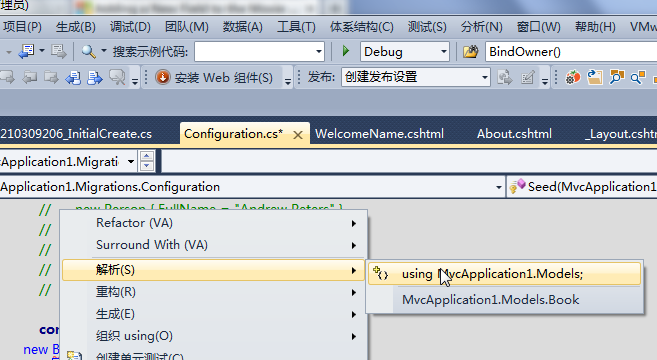
}

}

}

[复制代码](javascript:void(0);)

           7）使用鼠标右键单击Book下的红色波浪线，在弹出菜单中选择“解析”菜单项，然后在弹出菜单中选择“MvcApplication1.Models”;如下图。

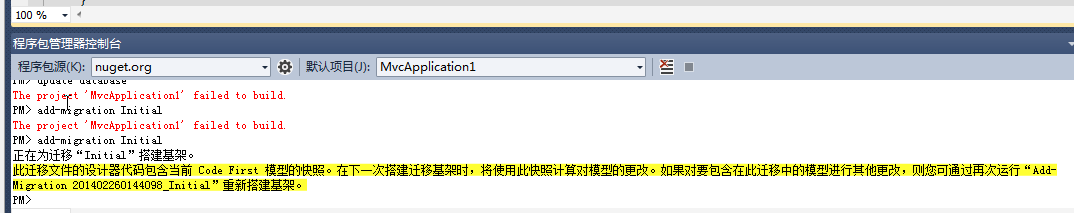


在经过这样做了之后，Visual Studio会自动在代码中增加using语句如下：

using MvcApplication1.Models;

        在“程序包管理器控制台”窗口中执行了“Code First Migrations ”功能中的“迁移”命名之后，代码会自动调用Send 方法（即程序包管理器控制台会调用此方法更新数据库中的数据），并且执行该方法时如果代码中的对象数据不存在则插入，如果存在则进行更新。

**二、迁移数据库**  
       1) 如果此时在Visual Studio中按Ctrl-Shift-B（或者菜单-->生成-->生成解决方案）来编译项目，将会出现编译不了的错误。如下图。   
       2) 接下来要建立一个DbMigration类来进行初始迁移。使用这个迁移命名创建一个新的数据库，这就是为什么我在上一步中删除了Test.mdf文件。   
       3) 在程序包管理器控制台窗口中，输入命令“dd-migration Initial”创建初始迁移。其中“Initial”的命名是任意的，是用来命名新创建的迁移文件。如下图。



           4)“Code First Migrations ”功能中的“迁移”命名会在迁移文件夹中的创建一个新的类文件（名为{DATESTAMP}\_Initial.cs），而且这个类中已经包含了创建数据库架构的代码。迁移文件名的命名规则中有一个固定的时间戳，用以帮助排序。检查{DATESTAMP}\_Initial.cs文件，它包含了创建Books表的数据库的说明。这个{DATESTAMP}\_Initial.cs文件将运行并创建数据库架构。然后运行Send方法为数据库填充测试数据。   
           5) 在程序包管理器控制台，输入命令“update-database”来创建数据库，并执行Send方法。如下图1，图2。

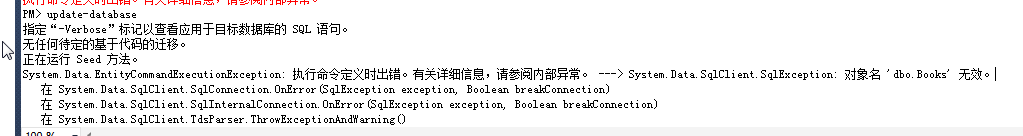


 图1

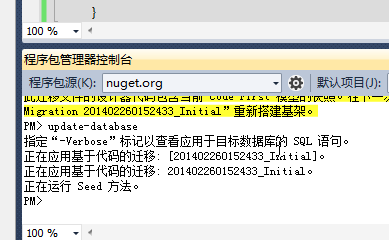


图2

       如果你得到一个错误，指示表已经存在，不能被创建，它可能是因为你正在运行应用程序，在你删除了数据库之后，执行更新数据库之前。（如上图1）。

      在这种情况下，再次删除Test.mdf文件，然后重试“update-database”命令。如上图2。如果仍然出现错误，请删除该文件夹和内容，然后开始在这个页面的顶部（也就是删除Test.mdf文件然后进行启用，迁移）的说明。   
          6) 按F5运行应用程序，在浏览器中浏览Books/Index网址。会显示Send方法中填充到数据库中的数据。如下图。

