

动软实战攻略

手把手教你用动软

文档编号: 20110421

版权所有 © 2004-2011 动软

在线帮助: <http://help.maticsoft.com>



目录

一.	产品介绍	3
二.	下载安装	3
三.	注册数据库连接	4
四.	查看数据库信息	8
五.	新建整个项目	9
六.	批量代码生成	15
七.	单表代码生成	16
八.	父子表代码生成（事务生成）	19
九.	代码生成规则设置	22
十.	自动生成存储过程	24
十一.	自动生成数据库 SQL 脚本	27
十二.	生成数据库文档	31
十三.	常用工具	32
1.	搜索表功能	32
2.	项目发布功能	33
3.	代码转换功能	35
十四.	代码生成组件接口开发	35
	常见问题（单击以下链接查看内容详细）	36

一. 产品介绍

动软.Net 代码生成器 是一款为.Net 程序员设计的自动代码生成器, 也是一个智能化软件开发平台, 它可以生成基于面向对象的思想 and 三层架构设计的代码, 结合了软件开发中经典的思想 and 设计模式, 融入了工厂模式, 反射机制等等一些思想。主要实现在对应数据库中表的基类代码的自动生成, 包括生成属性、添加、修改、删除、查询、存在性、Model 类构造等基础代码片断, 支持不同架构代码生成, 使程序员可以节省大量机械录入的时间和重复劳动, 而将精力集中于核心业务逻辑的开发。动软代码生成器 同时提供便捷的开发管理功能和多项开发工作中常用到的辅助工具功能, 您可以很方便轻松地进行项目开发。

动软让软件开发变得轻松而快乐! 让企业不断提升开发效率, 同样的时间创造出更大的价值。

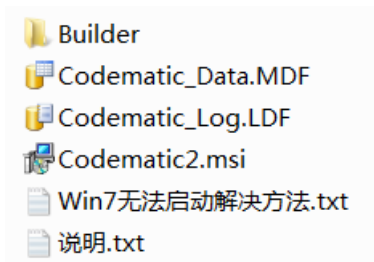
二. 下载安装

1. 系统要求:

Microsoft Windows2000/XP/2003/7 或者更高。机器必须安装.NET Framework v2.0。

2. 官方下载地址: <http://www.maticsoft.com/download.aspx>

3. 下载解压后安装包有如下文件:



Codematic2.msi 是动软.NET 代码生成器的安装文件。

Builder 文件夹是代码生成插件的源码, 动软.NET 代码生成器支持可扩展的代码生成插件, 用户可以定制自己的代码生成的插件, 根据接口开发自己的代码生成方式, 按自己的需求进行代码生成。

Codematic_Data.MDF 和 **Codematic_Log.LDF** 是通过动软新建项目中所带管理模块所需要的数据库文件。后台管理员默认登录用户名:admin 密码: 1

4. 双击 **Codematic2.msi** 进行直接安装即可。

安装动软时, 如果用户机器 360 弹出警告, 那仅仅是个签名认证提示, 并非木马, 选择“继续安装”, 然后点击“确定”即可。



动软郑重声明： 动软.NET 代码生成器，绝无插件木马，纯绿色软件。请放心安装。

5. 安装成功后，在开始-菜单和桌面上会有动软.NET 代码生成器的图标。

三. 注册数据库连接

安装完成后，首先要做的就是注册数据库的连接，因为所有的代码生成都将从数据库中获取数据结构信息进行生成。

1. 在软件界面的左侧数据库视图窗口，选择【服务器】，右键出现菜单：



2. 选择【添加服务器】，然后，出现“选择数据库类型”窗口



3. 根据自己的实际情况，选择一个机器上有的或自己项目中在用的数据库类型。并确保你选择的数据库是可以正常访问的。然后，【下一步】。
4. 如果选择的是 SQL Server，则会出现：



注意事项：

(1) 输入服务器 IP 地址，如果是本机也可以是：(local) 或是 . 或者 127.0.0.1。

如果服务器并非只有一个默认实例，请采用：服务器\实例名 的方式连接。

(2) 一定要选择和实际数据库服务器版本一致的选项，否则会导致连接数据库错误。

注意：请使用 SQLServer 的企业版或正式版本，不能是 SQL EXPRESS 版，否则无法连接。

(3) 身份验证可以选择是 SQL Server 认证，还是 Windows 认证。

(4) 输入数据库服务器用户名密码。如果不知道，请联系你的数据库管理员。

(5) 如果数据库的表比较多，连接速度会比较慢，启用【高效连接模式】实现快速连接。

(6) 可以通过【连接/测试】，来连接服务器并获取数据库列表，从而可以实现只选择连接一个库进行操作，减少不必要的连接时间，提高工作效率。

5. 如果选择的是 Oracle

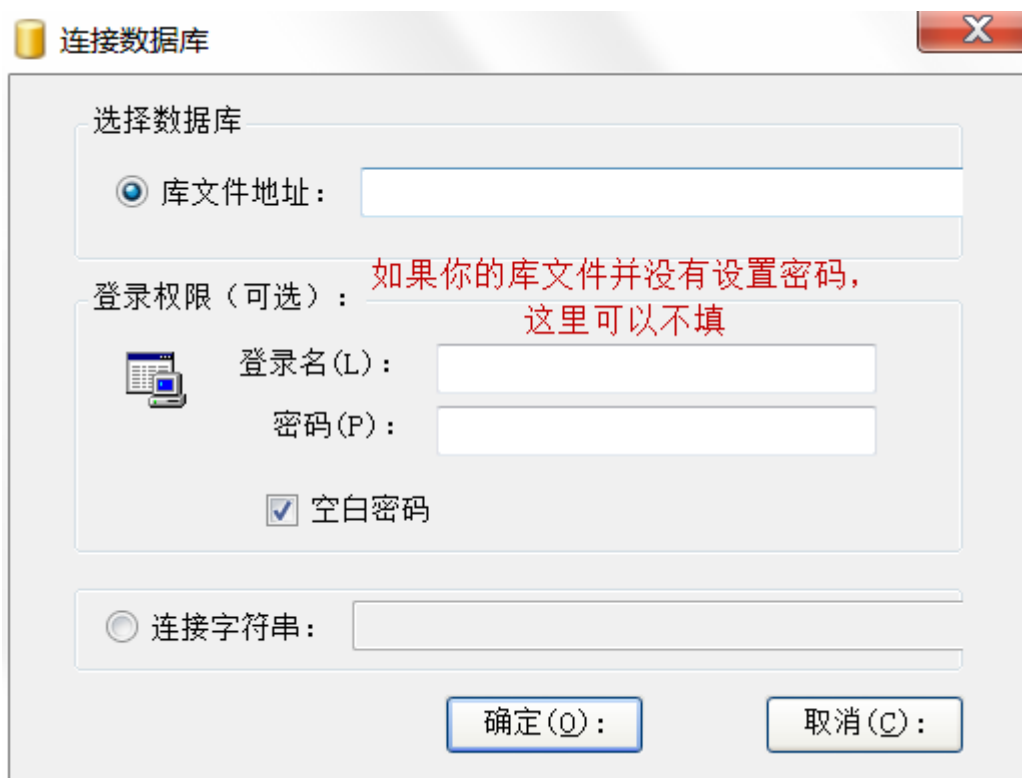


输入您自己安装过的 Oracle 的管理员用户名和密码，服务就是安装的 Oracle 服务名，一般默认是 Oracle 所在的机器名，如果不确定，请联系您的数据库管理员。

6. 如果选择的是 MySQL

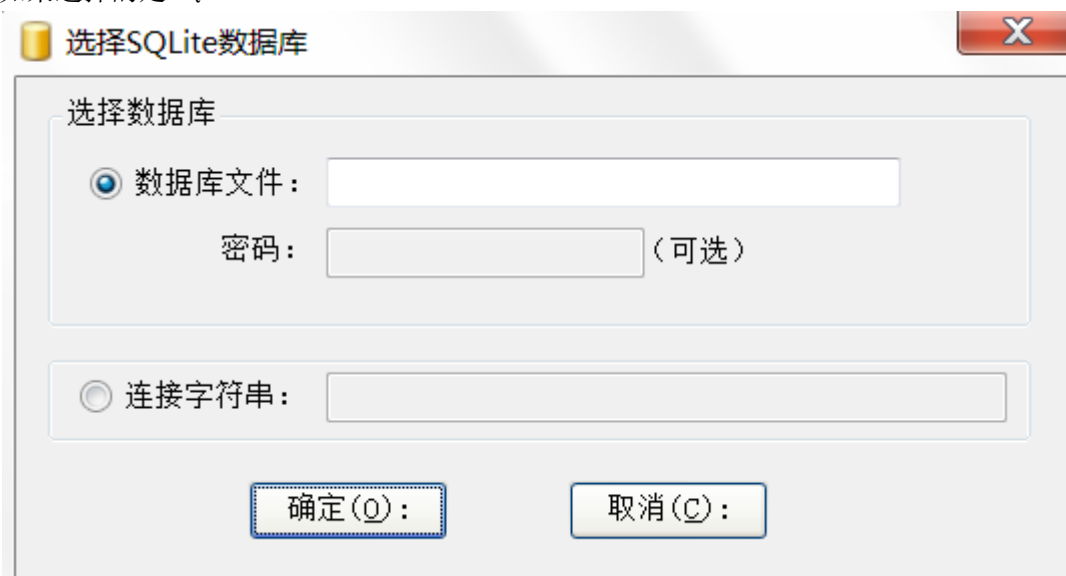


7. 如果选择的是 Oledb

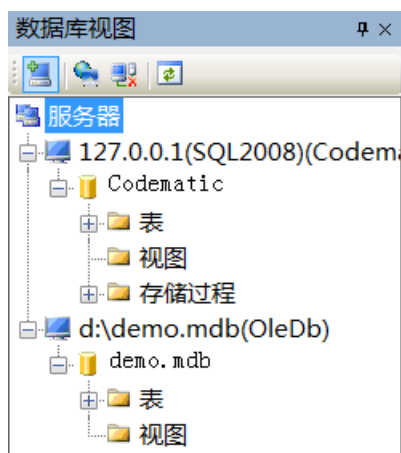


在地址文本框，可以直接输入 Access 库的文件地址，注意：库文件尽量不要放在桌面，并且确保你的文件地址正确。如果你的 Access 并没有设置密码，下面的密码可以不用管。

8. 如果选择的是 SQLite

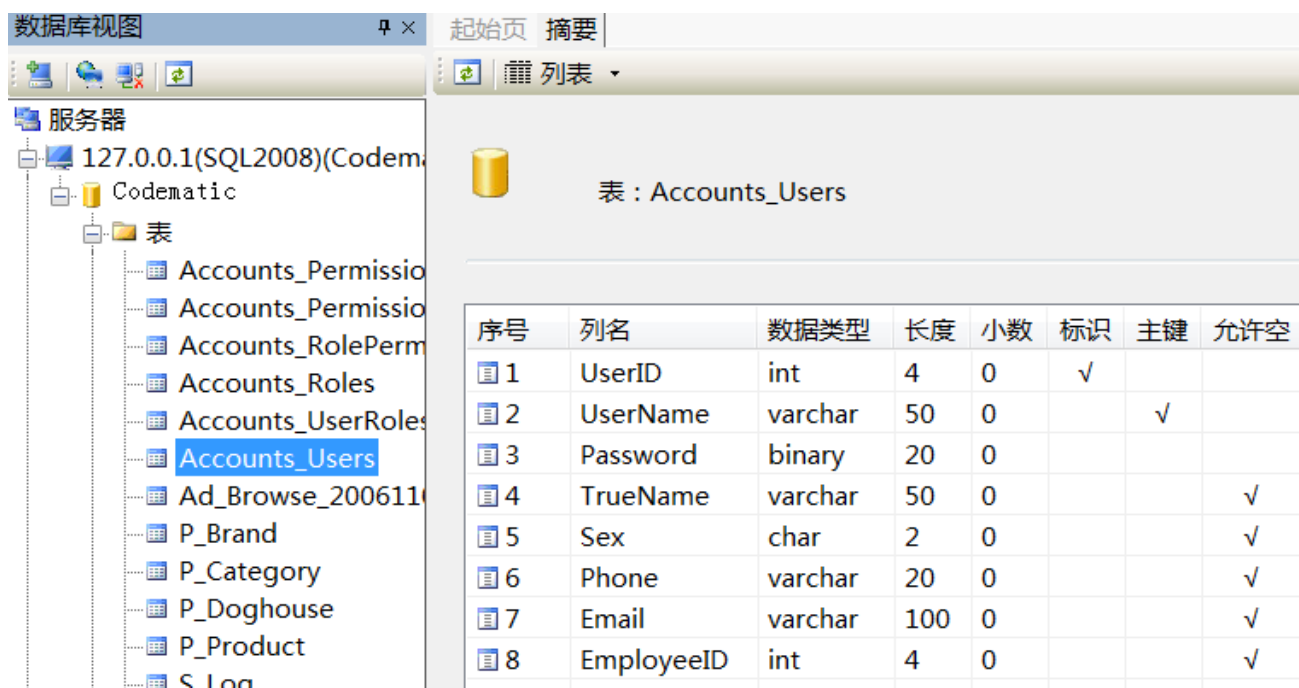
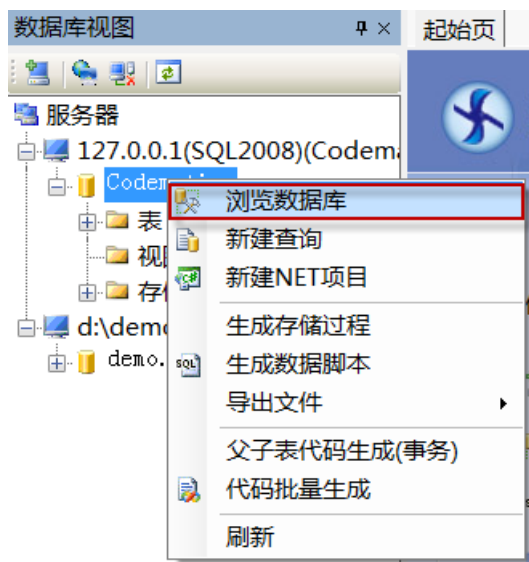


9. 在以上步骤确定后，动软.NET 代码生成器的数据库视图就出现了数据库服务器的信息。

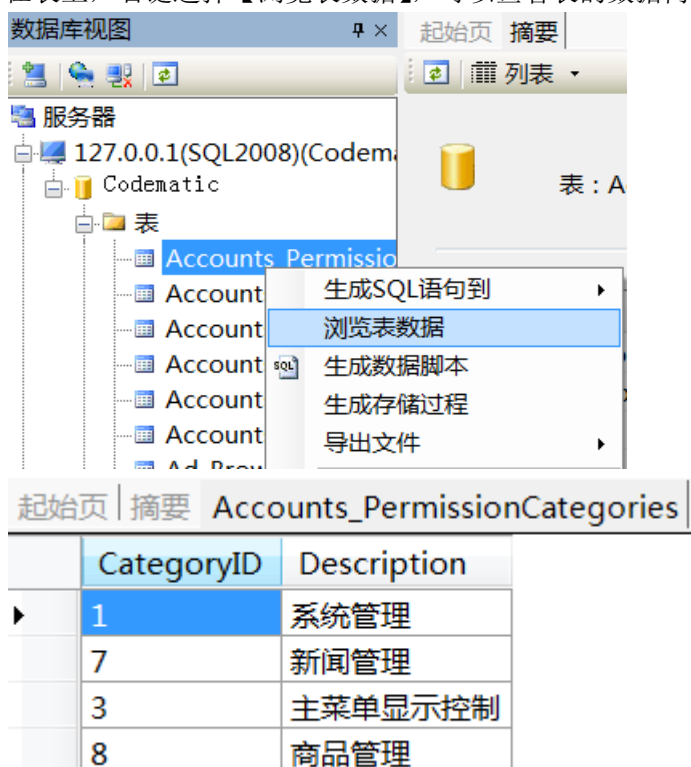


四. 查看数据库信息

1. 在数据库上右键，【浏览数据库】，通过选择库和表可以查看表和字段的信息。



2. 选择【新建查询】菜单，即出现 SQL 的查询分析器窗口，可以输入 SQL 语句进行查询。
3. 在表上，右键选择【浏览表数据】，可以查看表的数据内容。



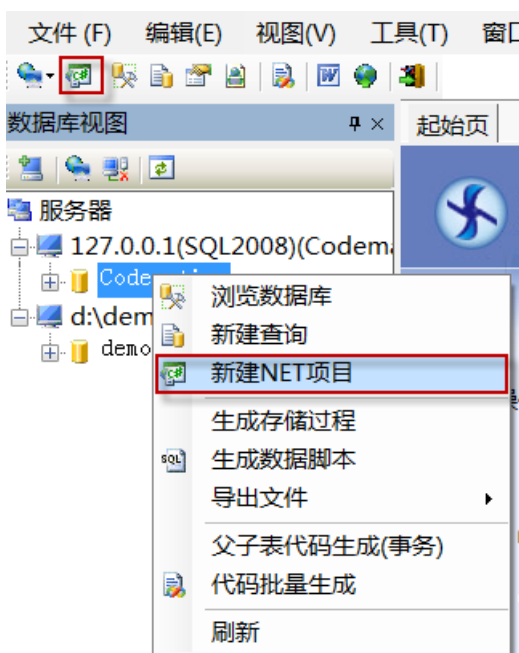
4.

五. 新建整个项目

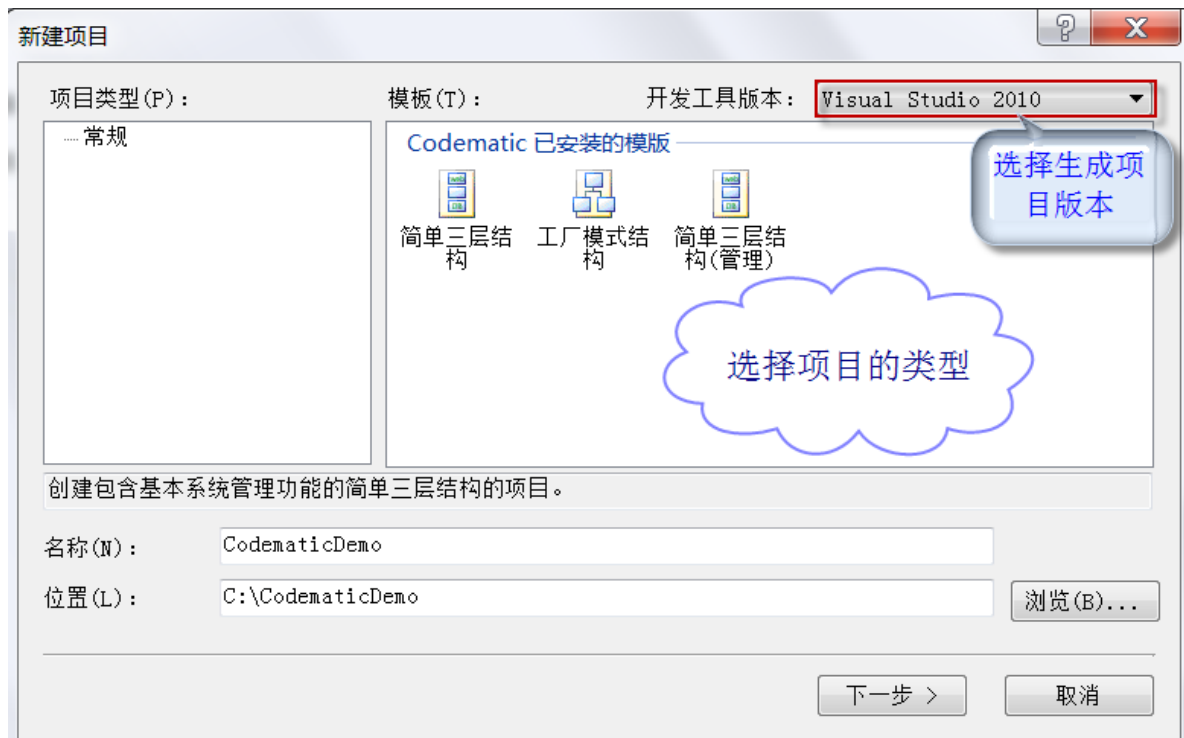
在看过了基本的数据库管理功能之外，下面我们就可以开始生成代码了。

首先，一般第一次生成，我们要生成的是整个项目框架。

1. 选中数据库，然后右键：【新建 NET 项目】，或者直接点工具栏上的快捷图标均可。



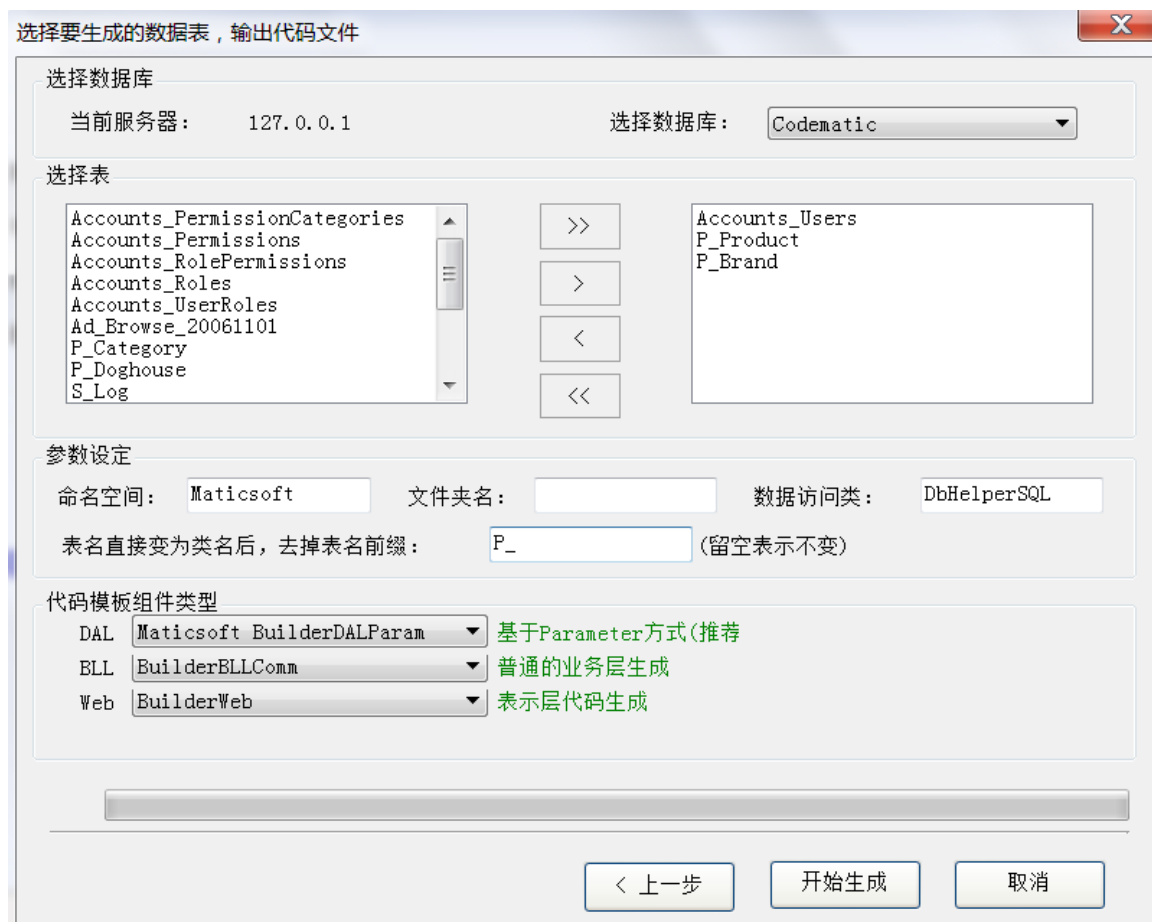
2. 然后，选择项目类型和版本：



- 简单三层结构：生成标准的三层架构项目。
- 工厂模式结构：生成基于工厂模式的项目架构，适合一个项目多数据库类型的情况。
- 简单三层结构（管理）：生成标准的三层架构项目，并且带有基本的系统管理功能和界面，这些通用的功能主要是节省开发人员的时间，可以在此基础上直接去开发自身业务模块。

这里暂以“简单三层结构（管理）”为例进行说明。

3. 点击【下一步】，选择要生成的表和配置：



双击选择要生成的表，选到右侧列表框。然后点击【开始生成】即可。

如果需要修改一些配置可以修改你自己的命名空间名字，是否去掉表的前缀。

代码模板组件类型，一般初学者建议默认即可。

相关组件说明：

BuilderDALParam 数据访问层（DAL）基于 Parameter 方式的代码生成组件(推荐)

BuilderDALProc 数据访问层（DAL）基于存储过程方式的代码生成组件

BuilderDALSQL 数据访问层（DAL）基于 SQL 拼接方式的代码生成组件

BuilderDALTranParam 数据访问层（DAL）带有事务的代码生成组件

DAL 由于不同项目要求不同，根据项目需求，选择其中一种方式即可。

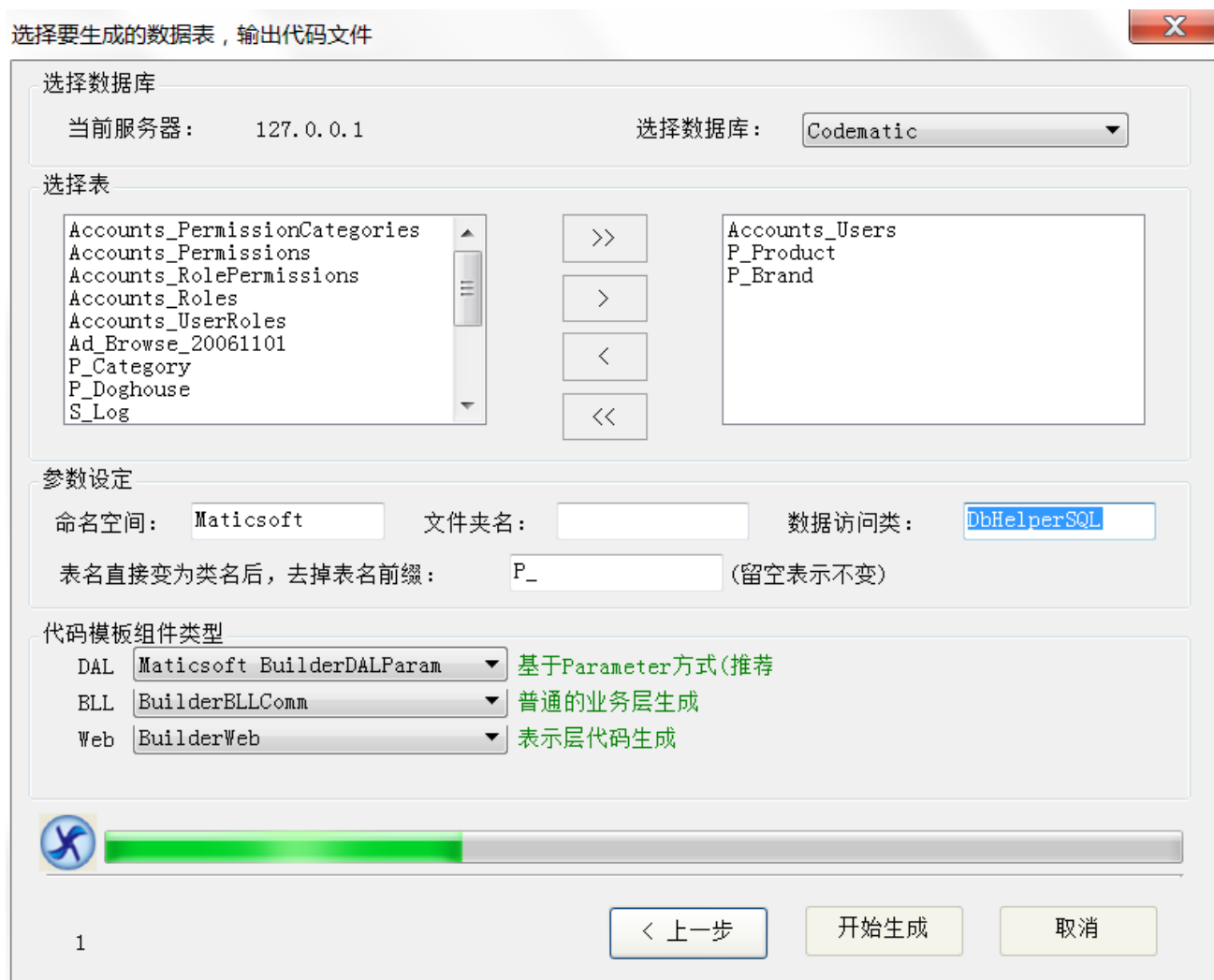
BuilderBLLComm 基于标准的业务逻辑层代码（BLL）

BuilderModel Model 层的代码生成组件

BuilderWeb 表示层的代码生成组件

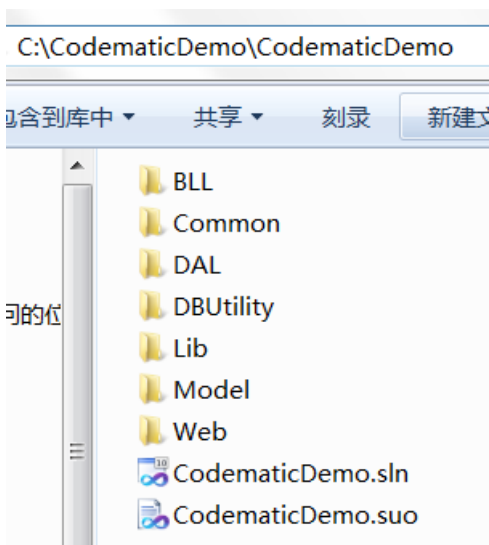
备注：代码还有一些生成规则，是在菜单【工具】-【选项】-【代码生成设置】中进行设置。

4. 点击【开始生成】，则开始进行代码的生成

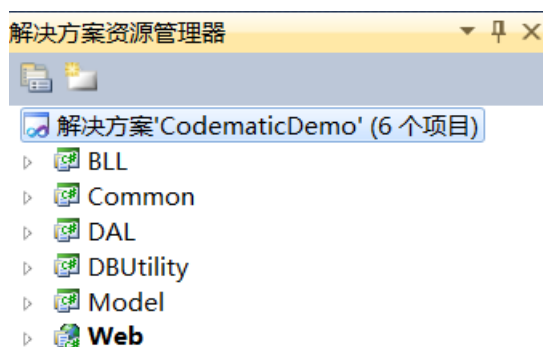


直到出现“项目工程生成成功”提示，项目生成完毕。

5. 打开生成的文件夹

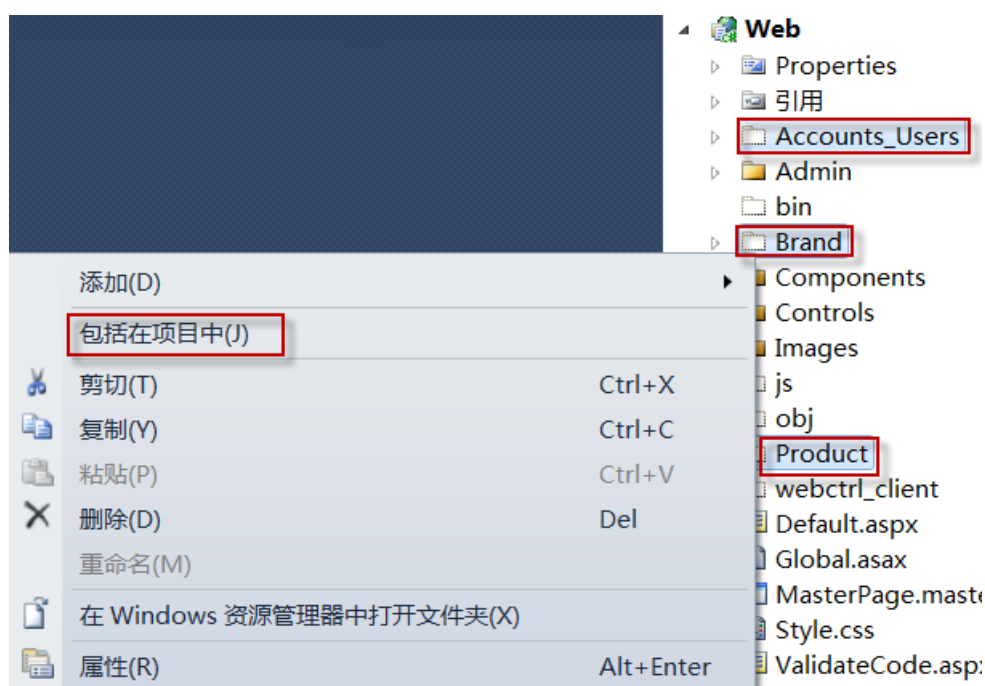


6. 双击解决方案文件，打开整个项目如图：



注：“简单三层结构”目前暂时是 VS2005 版本，是为了兼容当前还在用 2005 的朋友。如果你使用的是 VS2008，生成项目后，请先打开 VS2008，选择【菜单-文件-打开项目】的方式打开该项目，此时会提示升级项目版本，选择升级一下项目版本到 VS2008 或 VS2010 即可，对代码没有任何影响。

7. 打开 Web 项目，选中刚才选择生成的那几个表的页面文件夹，右键【包括在项目中】



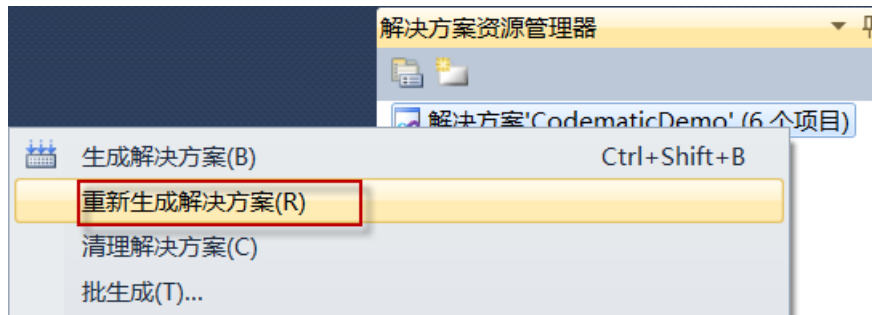
8. 打开 web 项目下 web.config 修改数据库连接字符串。

注意：新建项目后，请记得先将安装包里附带的数据库文件 Codematic_Data.MDF 附加到 SQLServer 中。

如果需要加密，可以使用安装包里的加解密工具（官方下载该工具）。

```
<?xml version="1.0"?>
<configuration>
  <appSettings>
    <!-- 连接字符串是否加密 -->
    <add key="ConStringEncrypt" value="false"/>
    <!-- 数据库连接字符串，（如果采用加密方式，上面一项要设置为true；加密工具，可在官方下载，
    如果使用明文这样server=127.0.0.1;database=.....,上面则设置为false。 -->
    <add key="ConnectionString" value="server=127.0.0.1;database=codematic;uid=sa;pwd=1"/>
    <!-- Maticsoft.Accounts权限角色管理模块连接字符串-->
    <add key="ConnectionStringAccounts" value="server=127.0.0.1;database=codematic;uid=sa;pwd=1"/>
    <!--其它模块连接字符串，可以不断增加以便同一个项目支持连接多个数据库。如果没有，可以删除该行-->
    <add key="ConnectionString2" value="server=127.0.0.1;database=codematic2;uid=sa;pwd=1"/>
  </appSettings>
</configuration>
```

9. 然后，选择解决方案进行重新生成整个解决方案。整个创建项目过程即全部完成。

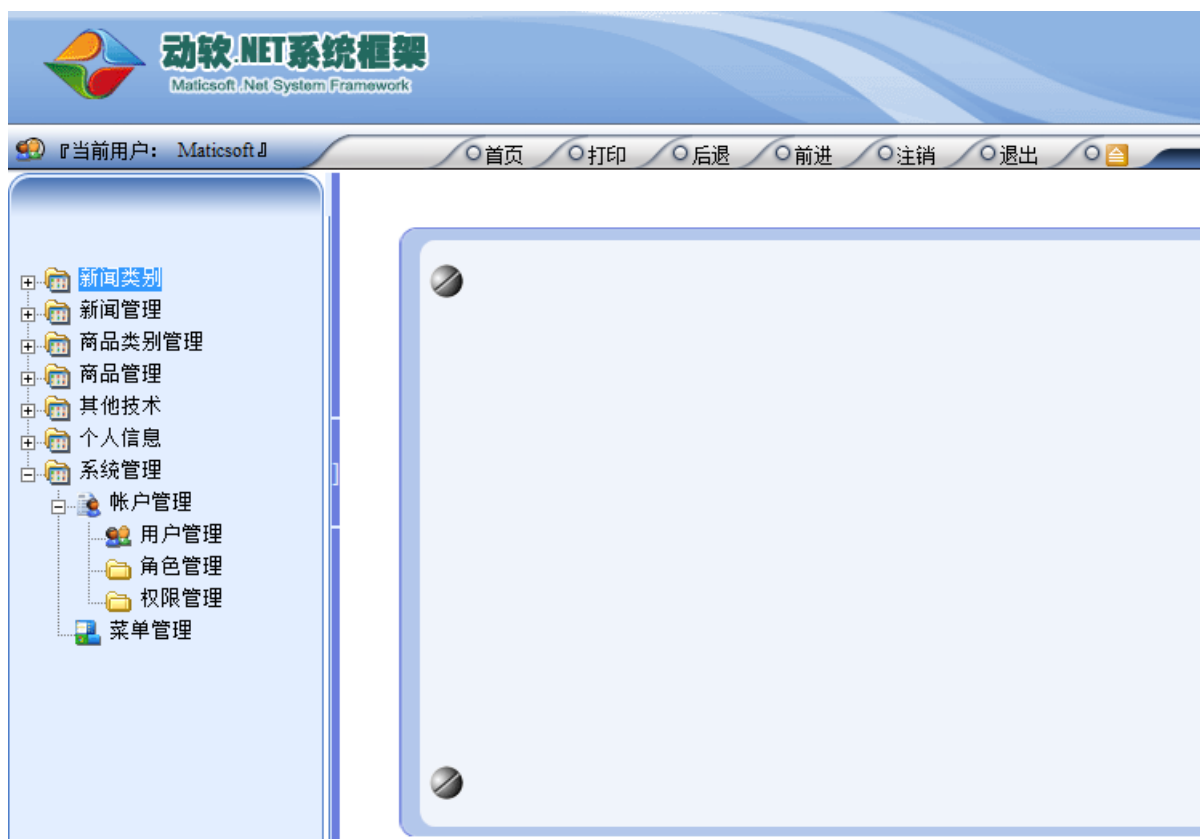


10. 如果编译没有错误，直接按 F5 键运行即可。整个创建项目过程即全部完成。

运行启动登录页面 login.aspx，输入用户名：admin，密码：1

即登录动软系统框架的后台，界面如下

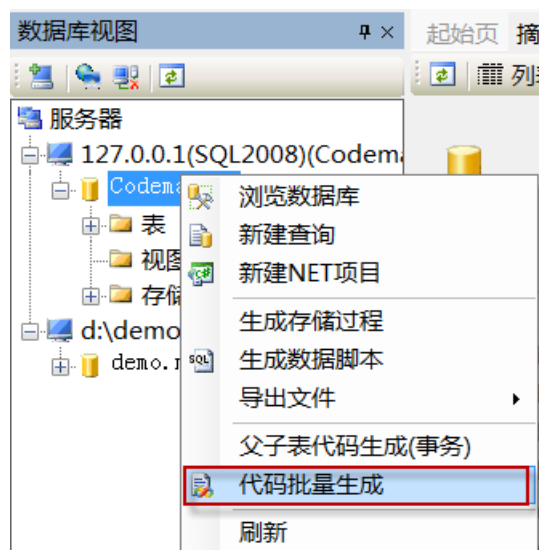




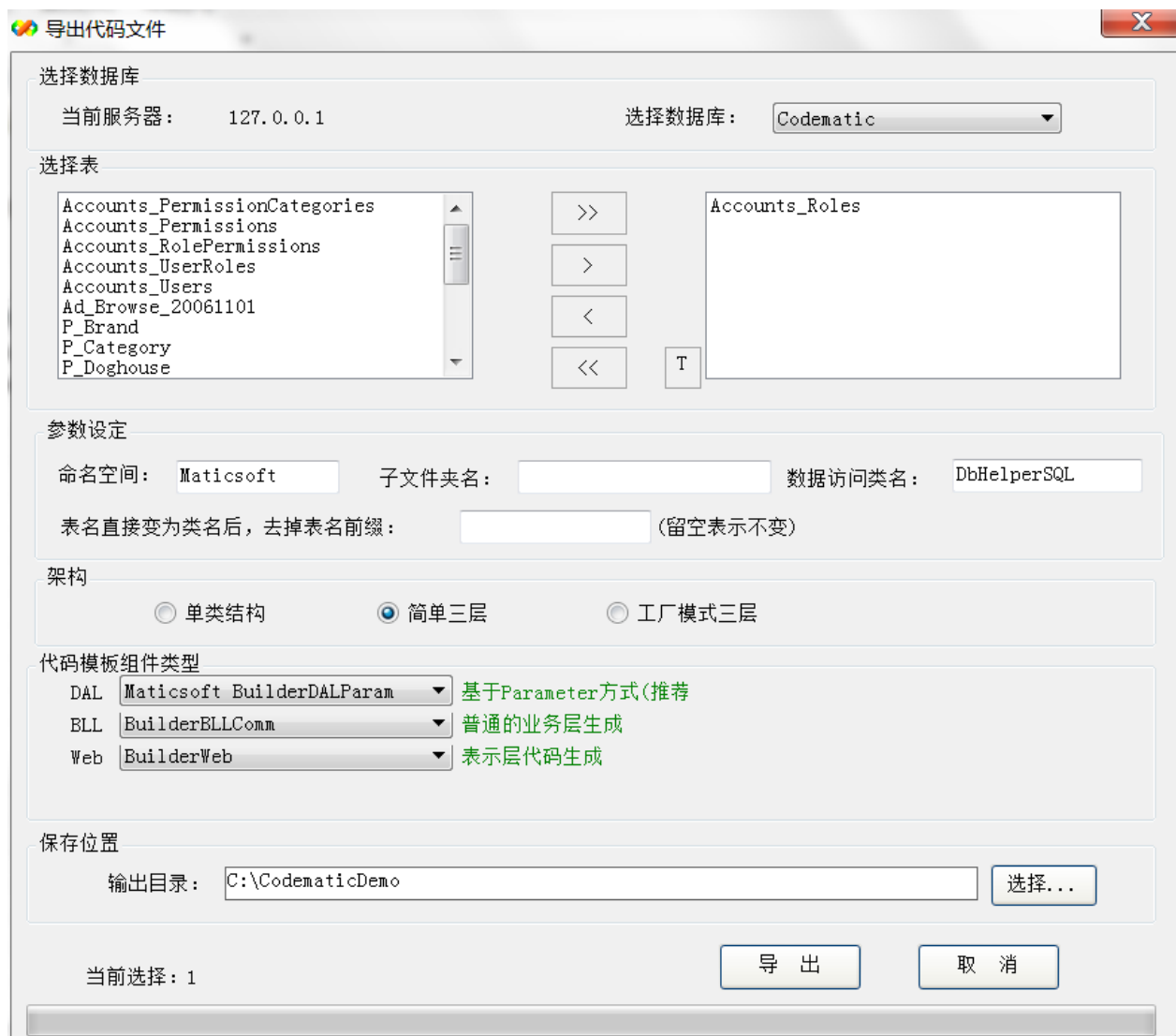
六. 批量代码生成

新建项目功能只适合于第一次，因为不可能每次都新建项目，特别是项目已经在开发中。所以，以后的项目开发中更多的应用的是【批量代码生成】功能。批量代码生成特别适合项目后期追加代码时使用。

1. 选中数据库或者表，然后单击右键菜单【代码批量生成】

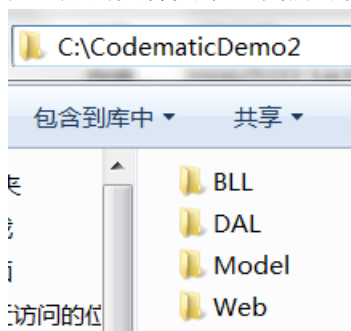


2. 出现的窗口和新建项目基本相似，只是多了一个选中架构的选项。



备注：代码还有一些生成规则，是在菜单【工具】-【选项】-【代码生成设置】中进行设置。

3. 选则要生成的表，然后点击【导出】
4. 在生成的文件夹中，我们可以看到：

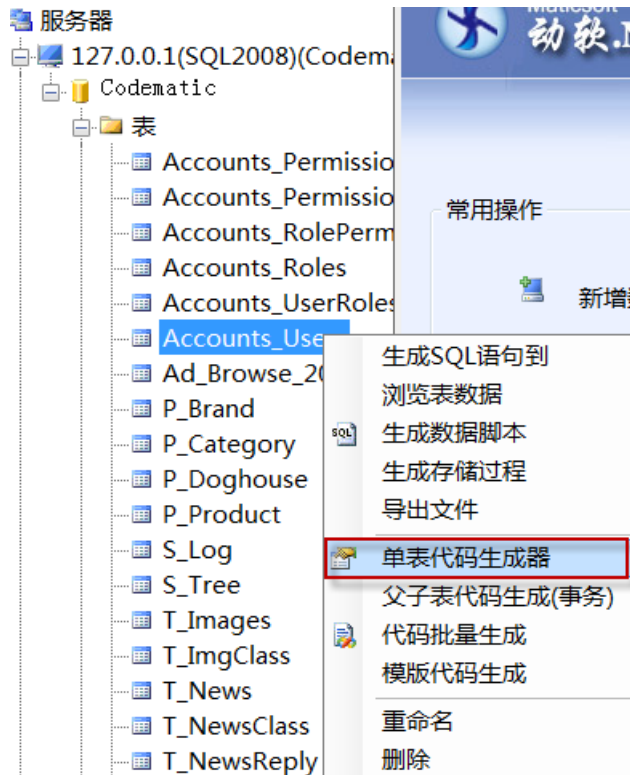


批量生成代码只生成业务表的代码，不再有解决方案文件和项目文件，以及其它类库等。我们可以将生成的这些文件直接拖到现有的解决方案中即可。

七. 单表代码生成

除了新建项目和批量代码生成，偶尔我们希望更个性化自定义一些代码生成的字段，而不是全部的自动生成，这是我们可以考虑针对单表的代码生成。

1. 在左侧【数据库视图】，选中表，右键菜单【单表代码生成器】



2. 然后，出现单表的代码生成器界面，我们设置自己需要更改的信息。

起始页 代码生成器

序号	列名	数据类型	长度	小数	标识	主键	允许空	默认值
<input checked="" type="checkbox"/> 1	UserID	int	4	0	√			
<input checked="" type="checkbox"/> 2	UserName	varchar	50	0		√		
<input checked="" type="checkbox"/> 3	Password	binary	20	0				
<input checked="" type="checkbox"/> 4	TrueName	varchar	50	0			√	
<input type="checkbox"/> 5	Sex	char	2	0			√	
<input checked="" type="checkbox"/> 6	Phone	varchar	20	0			√	
<input type="checkbox"/> 7	Email	varchar	100	0			√	

操作

全选(A)
反选(I)
清空(C)

添加主键(条件)字段
UserName
1个主键

生成代码

参数

项目名称:

二级命名空间:

顶级命名空间: Maticsoft

类名: User

类型

DB脚本
C#代码
Web页面

架构选择

单类结构
简单三层
工厂模式三层

代码类型

Model
DAL
BLL

代码模板组件类型

DAL
Maticsoft BuilderDALParam
基于Parameter方式(推荐)

方法选择

☐ GetMaxID
☒ Exists
☒ Add
☒ Update
☒ Delete
☒ GetModel
☒ GetModelByCache
☒ GetList

可以自定义选择要生成的字段

可以自定义条件字段或主键字段

对应三层中不同项目的代码

选项说明:

- **项目名称:** 主要用在生成 DB 脚本中。
- **二级命名空间:** 指的是这个类放在某一个二级文件夹下,从而命名空间中应该带这个文件夹的名字。
- **顶级命名空间:** 就是项目的命名空间名称。
- **类名:** 可以自己根据表名定义自己需要的名字。
- **类型:** 主要是生成什么代码, DB 脚本主要生成表的存储过程和表的创建脚本及数据脚本。
- **架构选择:** 目前仅支持这 3 种最常用的架构。
- **代码类型:** 指生成指定架构中具体某一个项目中的代码。
- **代码模板组件类型:** 指生成代码的方式,因为即使同一个代码有很多的写法,组件主要实现的是不同的写法,但每种写法实现的功能都是一样的。主要看项目需要和个人习惯进行选择。
- **方法选择:** 指生成最基本的增删改查的方法代码,后续版本将支持用户自定义这些方法。

备注: 代码还有一些生成规则,是在菜单【工具】-【选项】-【代码生成设置】中进行设置。

- 然后,点击【生成代码】按钮,即可生成该类的代码:

Maticsoft Technology Co., Ltd
<http://www.maticsoft.com> Mail: support@maticsoft.com

第 18 页 / 共 37 页

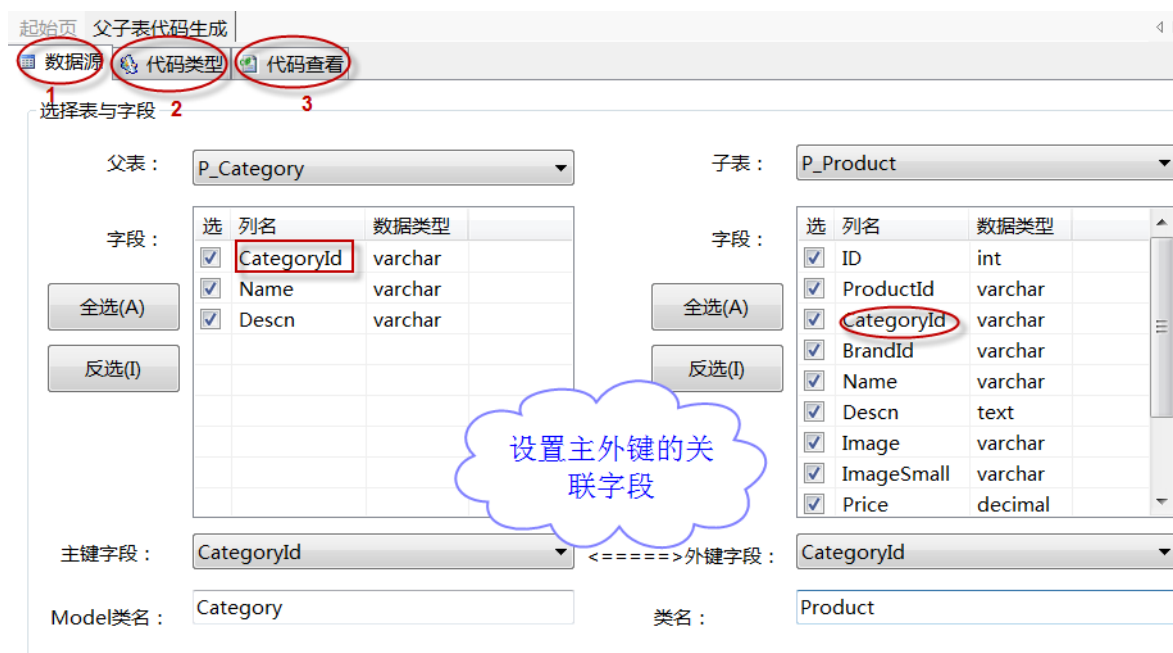
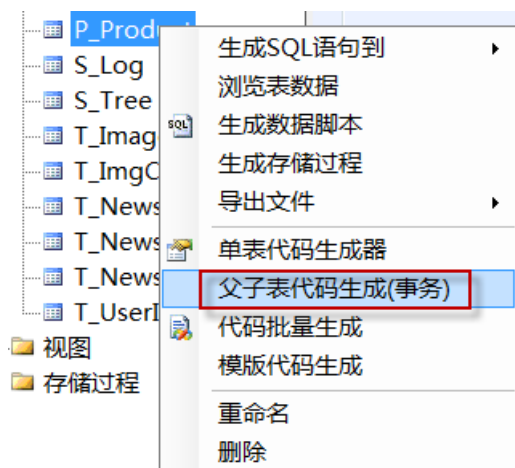


生成的代码，可以直接复制到项目文件中，也可以右键保存成 CS 文件。
通过窗体下面的 Tab 按钮可以来回切换设计视图和代码。

八. 父子表代码生成（事务生成）

日常开发中，经常会遇到父子表的情况，同时需要父子表的记录保存需要在事务中实现，以保证数据的一致性。动软也在功能上做了这方面的考虑。

1. 在表上，右键，选择【父子表代码生成（事务）】:



父子表的代码生成分 3 步窗体：1，2，3，当前第一个窗体主要设置父表和子表，以及二者主外键关联的字段。

2. 然后，选择【继续设置】

起始页 父子表代码生成

数据源 代码类型 代码查看

参数

项目名称 :

二级命名空间 :

顶级命名空间 : Maticsoft

类型

☒ C#代码
☐ Web页面

架构选择

☒ 简单三层
☐ 工厂模式三层

代码类型

☐ Model
☒ DAL
☐ BLL

方法选择

☒ Exists
☒ Add
☒ Update
☒ Delete
☒ GetModel
☒ GetModelByCache
☒ GetList

代码模板组件类型

DAL

BuilderDALTranParam

父子表生成

生成类代码 >>

设置代码生成的参数

选择父子表代码生成的组件。和其他的不一样的哦

3. 点击【生成类代码】:

起始页 父子表代码生成

数据源 代码类型 代码查看

```

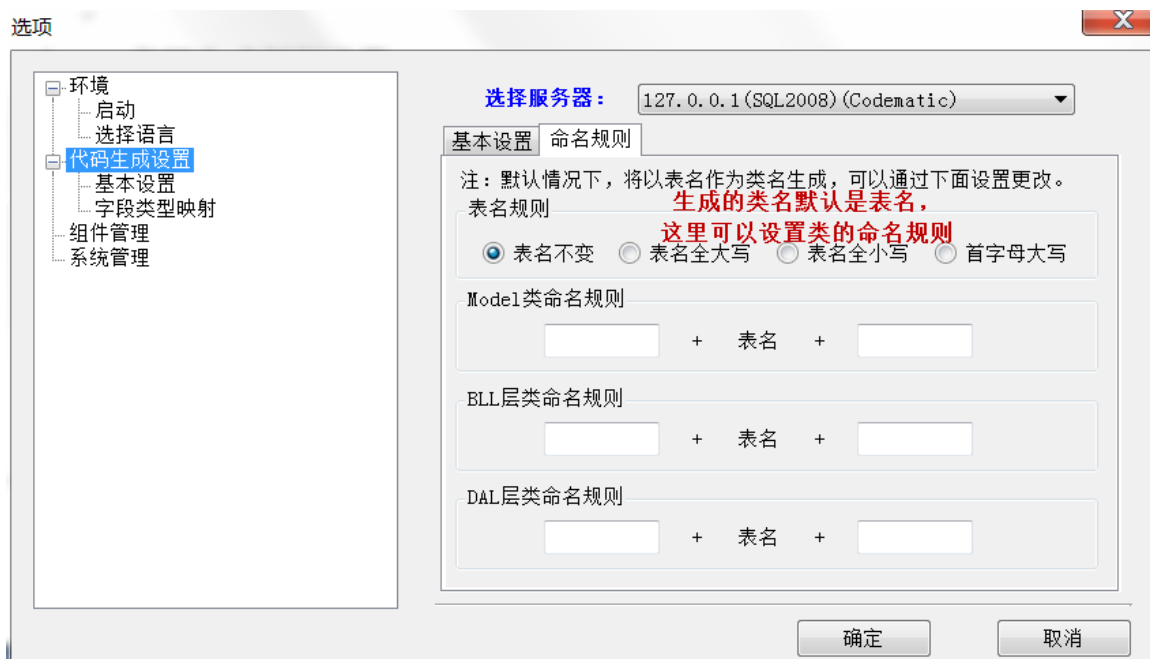
1  using System;
2  using System.Data;
3  using System.Text;
4  using System.Collections.Generic;
5  using System.Data.SqlClient;
6  using Maticsoft.DBUtility; //Please add references
7  namespace Maticsoft.DAL
8  {
9      /// <summary>
10     /// 数据访问类:Category
11     /// </summary>
12     public partial class Category
13     {
14         #region Method
15         /// <summary>
16         /// 增加一条数据,及其子表数据
17         /// </summary>
18         public void Add(Maticsoft.Model.Category model)
19         {
20             StringBuilder strSql=new StringBuilder();
21             strSql.Append("insert into P_Category(");
22             strSql.Append("CategoryId,Name,Descn");
23             strSql.Append(" values (");
24             strSql.Append("@CategoryId,@Name,@Descn");
25             SqlParameter[] parameters = {
26                 new SqlParameter("@CategoryId", SqlDbType.VarChar, 20),
27                 new SqlParameter("@Name", SqlDbType.VarChar, 80),
28                 new SqlParameter("@Descn", SqlDbType.VarChar, 255)};
29             parameters[0].Value = model.CategoryId;
30             parameters[1].Value = model.Name;
31             parameters[2].Value = model.Descn;
32
33             List<CommandInfo> sqlList = new List<CommandInfo>();
34             CommandInfo cmd = new CommandInfo(strSql.ToString(), parameters);
35             sqlList.Add(cmd);
36             StringBuilder strSql2;
37             foreach (Maticsoft.Model.Product models in model.Products)
38             {
39                 strSql2=new StringBuilder();
40                 strSql2.Append("insert into P_Product(");
41                 strSql2.Append("VipPrice,Cheapness,ProductId,CategoryId,BrandId,Name,I");
42                 strSql2.Append(" values (");
43                 strSql2.Append("@VipPrice,@Cheapness,@ProductId,@CategoryId,@BrandId,@");
44                 SqlParameter[] parameters2 = {

```

这里的所有插入语句都将放在一个事务里完成。

九. 代码生成规则设置

打开菜单【工具】-【选项】-【代码生成设置】



这些配置保存后，在生成代码的时候将按照这个规则进行生成。

代码生成的规则设置范围还在不断增加中。

另外，不同数据库类型的数据类型各有不同，这里提供了字段类型和 C# 中的类型建立映射关系，生成代码时将按映射关系来生成代码字段属性的类型。



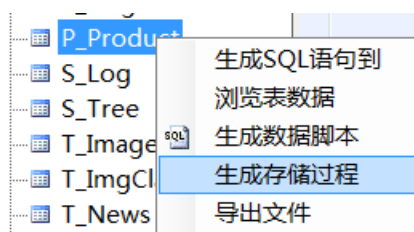
十. 自动生成存储过程

对于习惯使用存储过程的用户，手写表的增、删、改、查的存储过程也是非常头大的事情，动软提供了可以自动生成数据表增删改查的存储过程脚本生成。

主要通过以下几个地方进行生成：

1. 单表的存储过程生成：

在表上，右键选择【生成存储过程】



或者：在表上右键，选择【单表代码生成器】，可以更灵活的定义生成存储过程。

起始页 代码生成器

序号	列名	数据类型	长度	小数	标识	主键	允许空	默认值
<input checked="" type="checkbox"/> 1	ID	int	4	0	√			
<input checked="" type="checkbox"/> 2	ProductId	varchar	20	0		√		
<input checked="" type="checkbox"/> 3	CategoryId	varchar	20	0			√	
<input checked="" type="checkbox"/> 4	BrandId	varchar	20	0			√	
<input checked="" type="checkbox"/> 5	Name	varchar	80	0			√	
<input checked="" type="checkbox"/> 6	Descn	text		0			√	
<input checked="" type="checkbox"/> 7	Image	varchar	80	0			√	

操作

全选(A)

反选(I)

清空(C)

添加主键(条件)字段

ID

参数

项目名称 :

二级命名空间 :

顶级命名空间 : Maticsoft

类名 : P_f

类型

☒ DB脚本
☐ C#代码
☐ Web页面

数据库脚本

☒ 存储过程
☐ DDL数据脚本

存储过程前缀 :

+

P_Product

存储过程方法 :

☐ GetMaxID

☒ Exists

☒ Add

☒ Update

☒ Delete

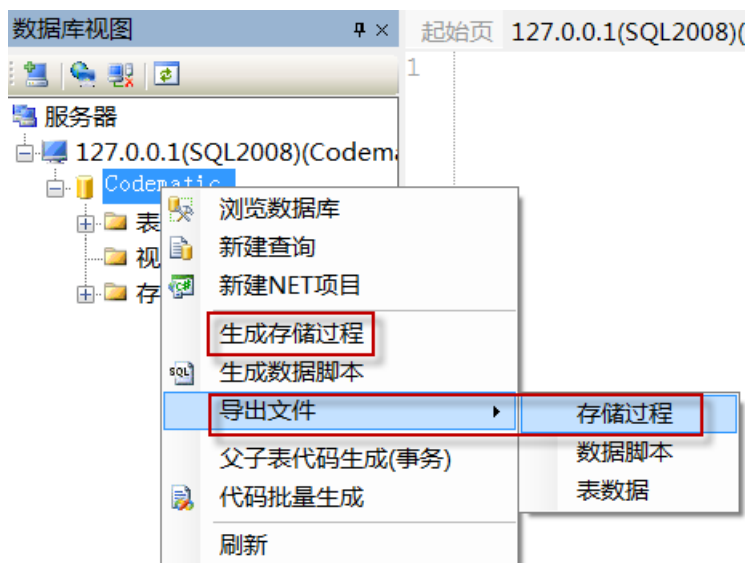
```

起始页 P_Product存储过程.sql
1  /*****
2  * 表名: P Product
3  * 时间: 2011/4/21 23:42:13
4  * Made by Codematic
5  *****/
6  if exists (select * from dbo.sysobjects where id
7  drop procedure [P Product ADD]
8  GO
9  -----
10 --用途: 增加一条记录
11 --项目名称:
12 --说明:
13 --时间: 2011/4/21 23:42:13
14 -----
15 CREATE PROCEDURE P Product ADD
16 @ID int output,
17 @ProductId varchar(20),
18 @CategoryId varchar(20),
19 @BrandId varchar(20),
20 @Name varchar(80),
21 @Descn text,
22 @Image varchar(80),
23 @ImageSmall varchar(80),
24 @Price decimal(19,2),
25 @VipPrice decimal(19,2),
26 @Cheapness int
27
28 AS
29     INSERT INTO [P Product] (
30         [ProductId], [CategoryId], [BrandId], [Name], [De
31     ) VALUES (
32         @ProductId, @CategoryId, @BrandId, @Name, @Descn,
33     )
34     SET @ID = @@IDENTITY
35
36 GO

```

2. 批量表的存储过程生成。

在数据库上，右键，选择【生成存储过程】，或者选择【导出文件】－【存储过程】。都将生成整个数据库中所有表的存储过程。



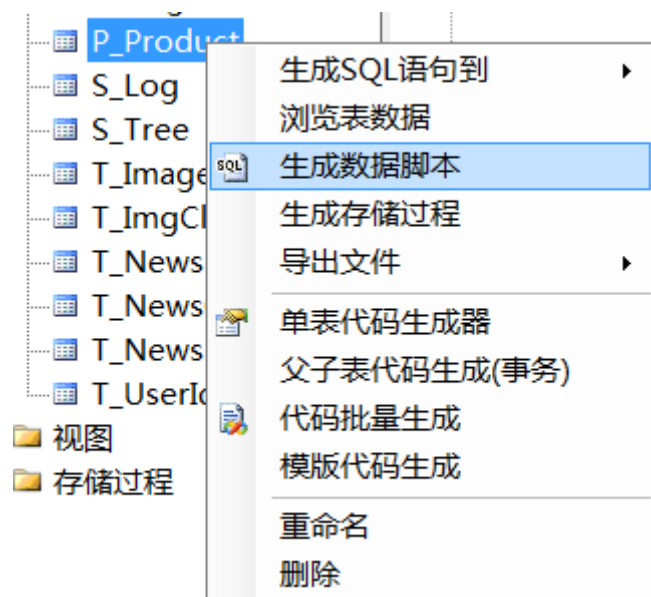
十一. 自动生成数据库 SQL 脚本

有时候需要迁移数据库，或者生成表结构脚本，导出数据脚本到别的数据库，日常开发中也比较常见，这个可以通过动软自动生成 SQL 脚本功能来帮您节省大量的时间。

主要通过以下几个地方进行生成：

1. 单表的数据脚本生成

在表上右键，选择【生成数据脚本】



或者，在表上右键，选择【单表代码生成器】，可以更灵活的定义生成存储过程

代码生成器

序号	列名	数据类型	长度	小数	标识	主键	允许空	默认值
<input checked="" type="checkbox"/> 1	ID	int	4	0	√			
<input checked="" type="checkbox"/> 2	ProductId	varchar	20	0		√		
<input checked="" type="checkbox"/> 3	CategoryId	varchar	20	0			√	
<input checked="" type="checkbox"/> 4	BrandId	varchar	20	0			√	
<input checked="" type="checkbox"/> 5	Name	varchar	80	0			√	

操作

全选(A)

反选(I)

清空(C)

添加主键(条件)字段

ID

参数

项目名称 :

二级命名空间 :

顶级命名空间 :

Maticsoft

类名 :

类型

☒ DB脚本

☐ C#代码

☐ Web页面

数据库脚本

☐ 存储过程

☒ DDL数据脚本

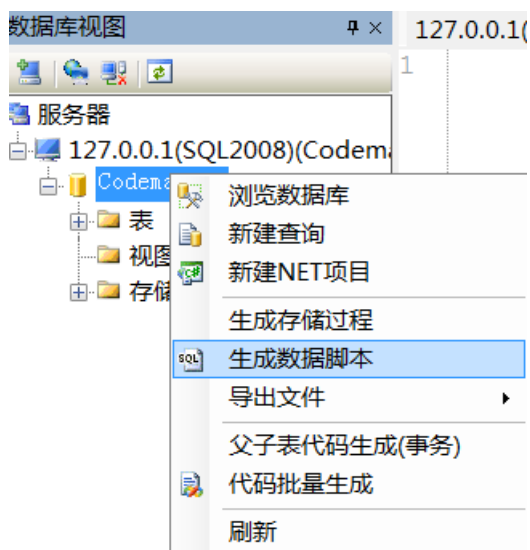
生成后:

代码生成器

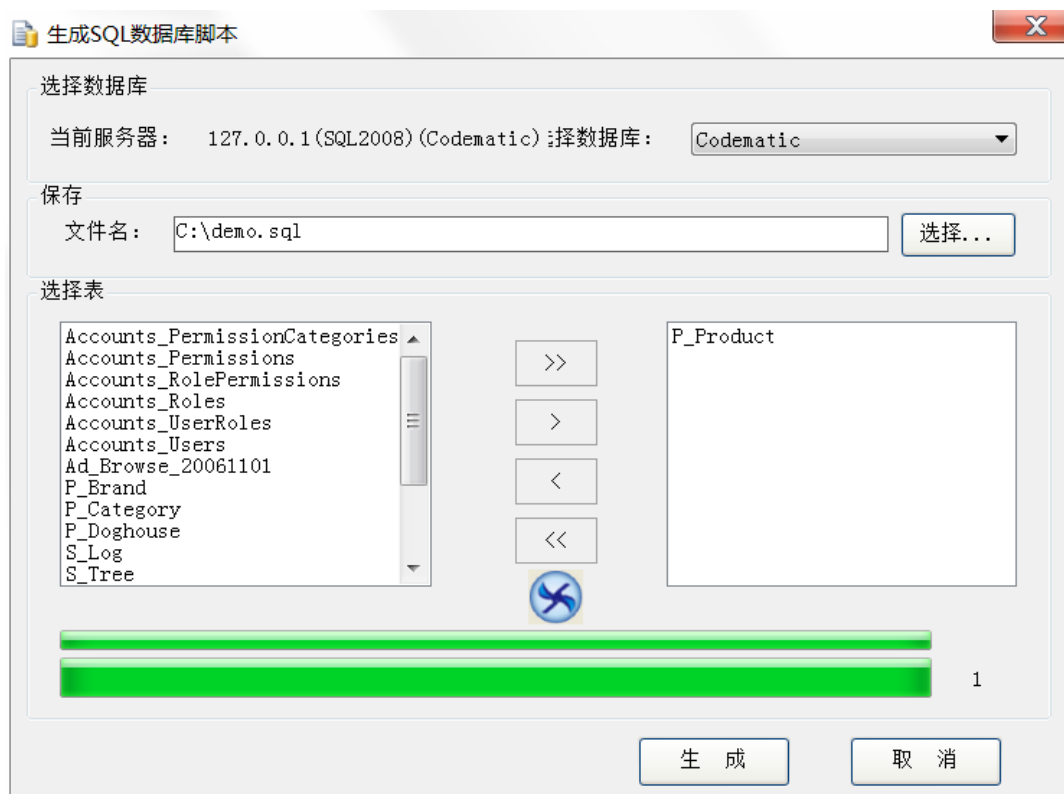
```

1  if exists (select * from sysobjects where id = OBJECT_ID(' [P_Prod
2  DROP TABLE [P_Product]
3
4  CREATE TABLE [P_Product] (
5  [ID] [int] IDENTITY (1, 1) NOT NULL,
6  [VipPrice] [decimal] (19,2) NULL,
7  [Cheapness] [int] NULL,
8  [ProductId] [varchar] (20) NOT NULL,
9  [CategoryId] [varchar] (20) NULL,
10 [BrandId] [varchar] (20) NULL,
11 [Name] [varchar] (80) NULL,
12 [Descn] [text] NULL,
13 [Image] [varchar] (80) NULL,
14 [ImageSmall] [varchar] (80) NULL,
15 [Price] [decimal] (19,2) NULL)
16
17 ALTER TABLE [P_Product] WITH NOCHECK ADD CONSTRAINT [PK_P_Produc
18 SET IDENTITY_INSERT [P_Product] ON
19
20 INSERT [P_Product] ([ID], [VipPrice], [Cheapness], [ProductId], [Cate
21 INSERT [P_Product] ([ID], [VipPrice], [Cheapness], [ProductId], [Cate
22 INSERT [P_Product] ([ID], [VipPrice], [Cheapness], [ProductId], [Cate
23 INSERT [P_Product] ([ID], [VipPrice], [Cheapness], [ProductId], [Cate
24 INSERT [P_Product] ([ID], [VipPrice], [Cheapness], [ProductId], [Cate
25 INSERT [P_Product] ([ID], [VipPrice], [Cheapness], [ProductId], [Cate
26 INSERT [P_Product] ([ID], [VipPrice], [Cheapness], [ProductId], [Cate
27
28 SET IDENTITY_INSERT [P_Product] OFF
    
```

2. 批量表的数据脚本生成。



出现生成 SQL 数据库脚本窗口，选择保存的文件名，点击【生成】即可。

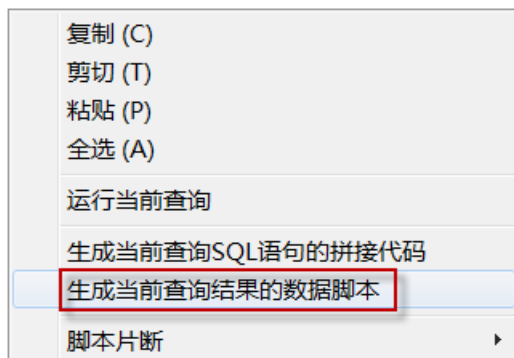


3. 根据查询结果生成数据脚本。

有时候我们并不像生成整个表的数据脚本，也许表很大，或者其他数据并不需要，只是需要一部分我们需要的数据，生成脚本后，执行导入到其他库，这时候我们可以使用动软代码生成器的这个功能。

在数据库上，右键【新建查询】，或者选择菜单【工具】-【查询分析器】，打开查询语句窗口，输入我们的查询语句，然后单击右键，选择【生成当前查询结果的数据脚本】：

```
P_Product查询.sql
1  select * from [P Product] where price < 100
```



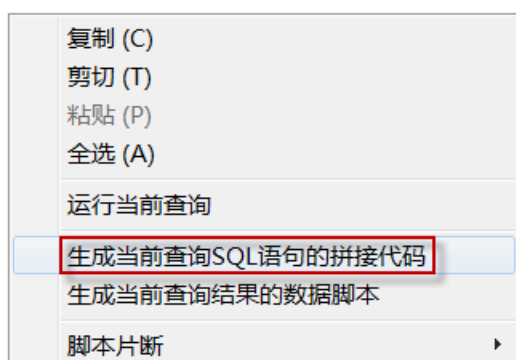
则生成的脚本仅仅包括语句查询结果的数据脚本：

```
P_Product查询.sql SQL1.sql
1  INSERT [P_Product] ([ID], [ProductId], [CategoryId], [Name], [Descn], [
2  INSERT [P_Product] ([ID], [ProductId], [CategoryId], [Name], [Descn], [
3  INSERT [P_Product] ([ID], [ProductId], [CategoryId], [Name], [Descn], [
4  INSERT [P_Product] ([ID], [ProductId], [CategoryId], [Name], [Descn], [
5
```

如此方便的功能，不得不说很贴心哦。

除此之外，我们可以看到，还有一个非常贴心的功能，就是如果数据访问层使用的是基于 SQL 拼接的方式的 DAL，有时候为了拼写一段长长的 SQL 语句，眼都看花了，看看动软这个功能吧：在查询分析器中，测试我们的查询语句，然后，右键选择【生成当前查询 SQL 语句的拼接代码】

```
P_Product查询.sql
1  select * from [P Product] where price < 100
```



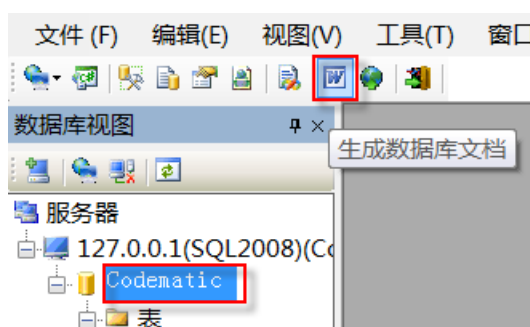
生成后，这些代码我们直接就可以复制到数据访问层，不用再用手一个个敲了，省出来的时间我们可以去喝咖啡喽。

```
P_Product查询.sql Class1.cs
1  StringBuilder strSql=new StringBuilder();
2  strSql.Append("select * from [P_Product] where price < 100 ");
3
```

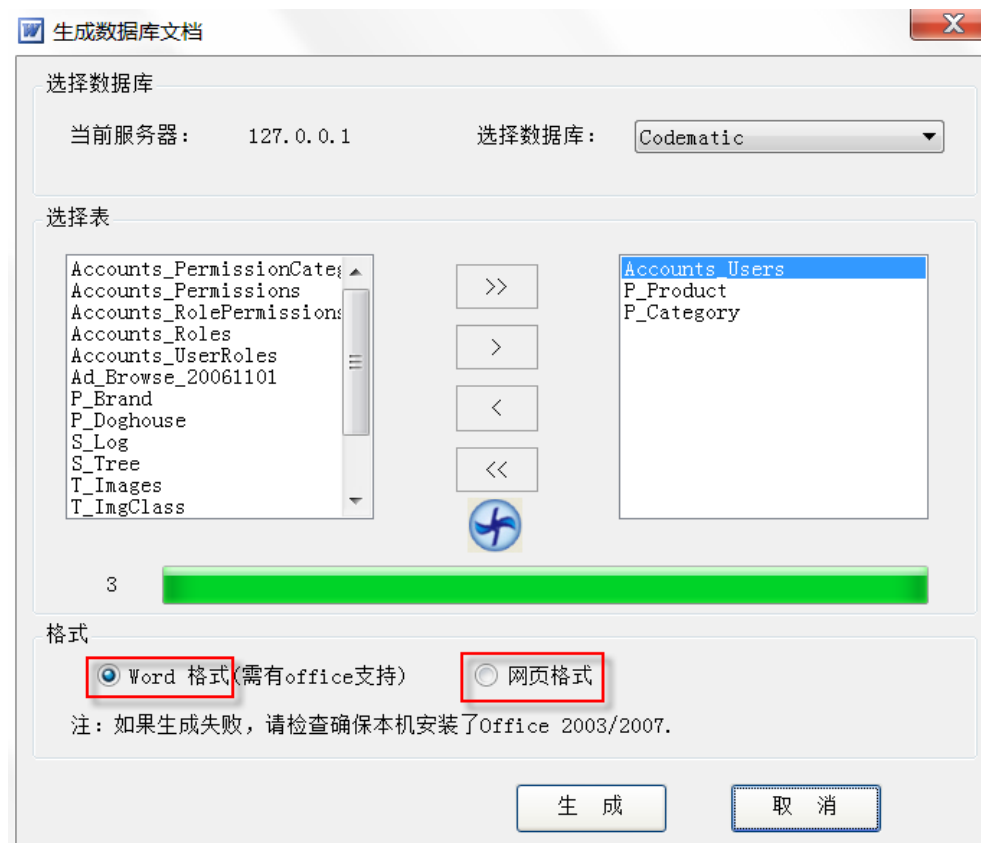
十二. 生成数据库文档

日常开发，表一多，是不是就头疼每个字段什么意思，以及数据库总是变，文档维护起来很麻烦。特别是新人加入项目团队时，如果没有数据库文档，那么更是云里雾里的搞不清各种数据的意思。动软.NET代码生成器，可以帮你解决这个问题，它可以根据选定的数据库和表批量生成所有表的字段详细信息，包括默认值，描述等表结构信息。免去手工写文档和维护的麻烦，真的不用加班熬夜，可以早回家了。

1. 选中要生成的数据库，然后选中工具栏【生成数据库文档】按钮。



2. 然后，选中需要生成的表

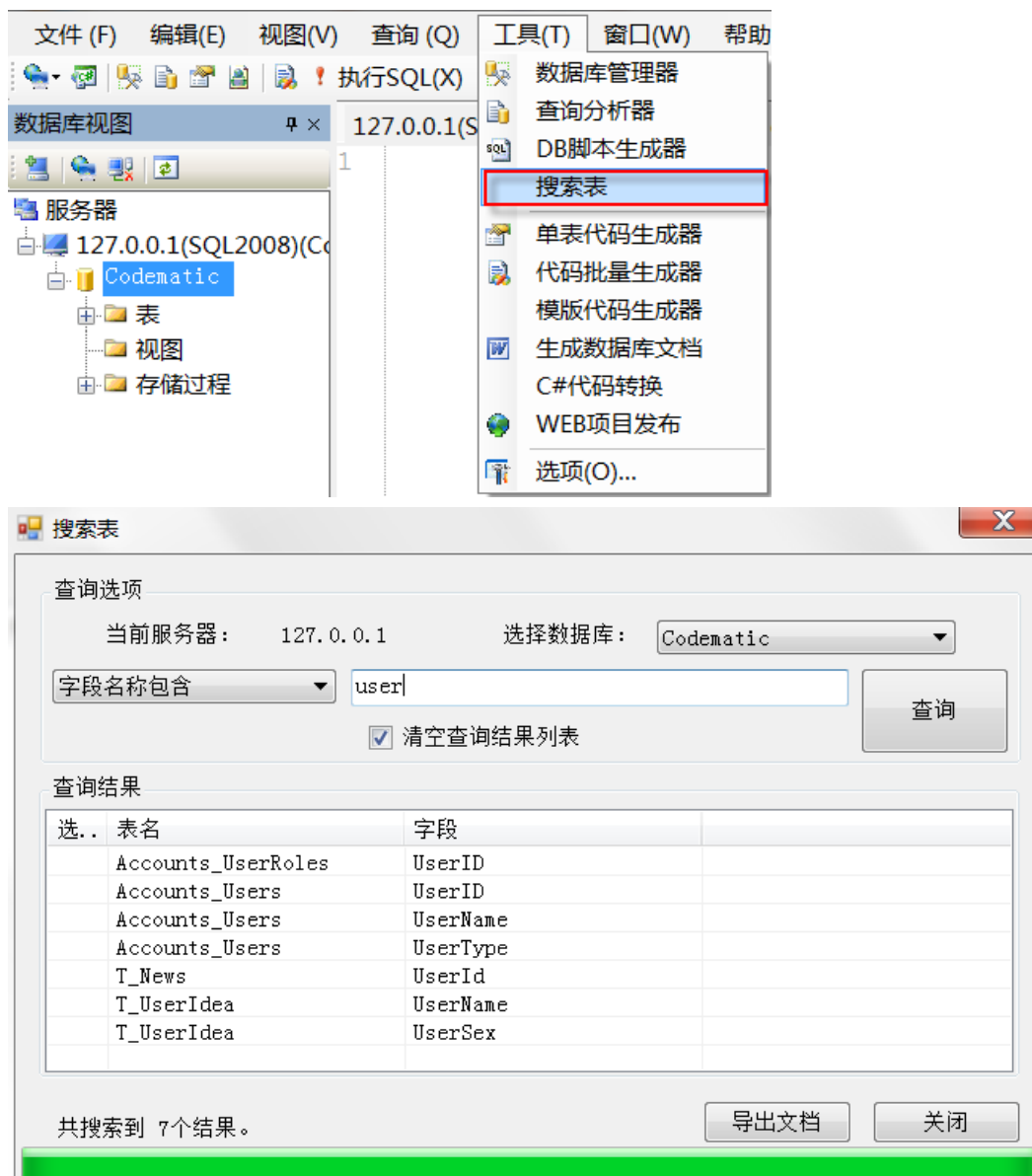


这里提供了两种格式的文档，一种是 Word 格式的，但需要本机安装 Office2003 或 2007。另一种是网页格式，可以生成两种界面风格的网页格式数据库文档。



十三. 常用工具

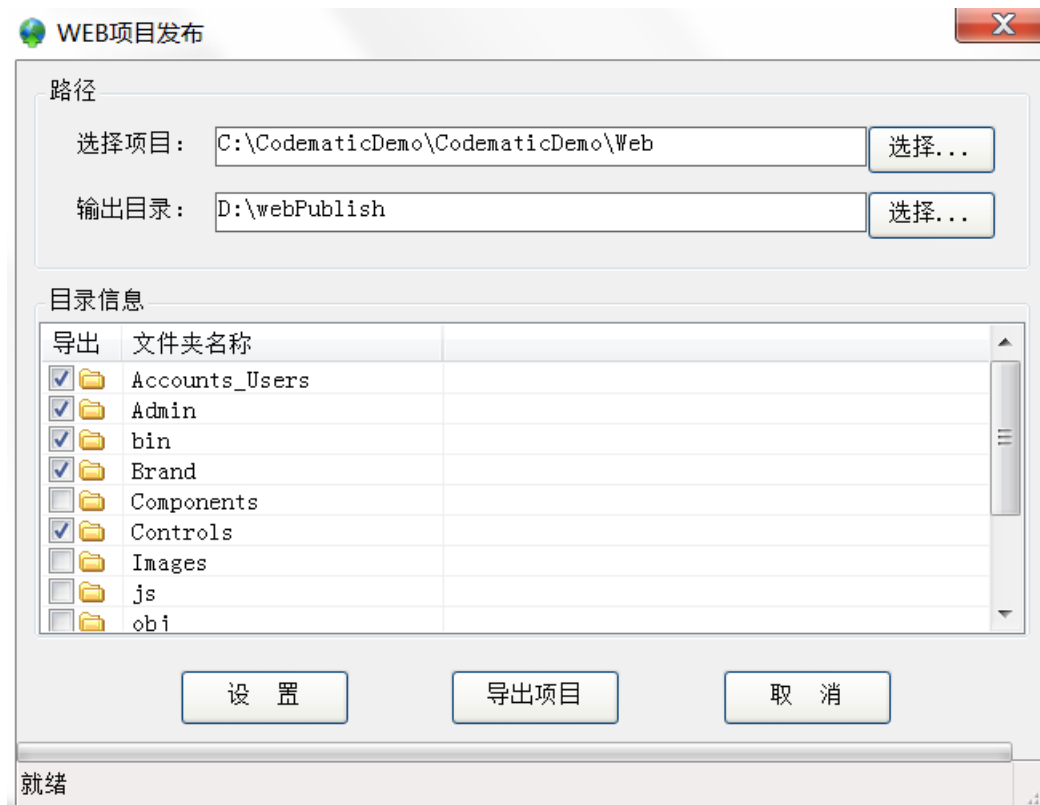
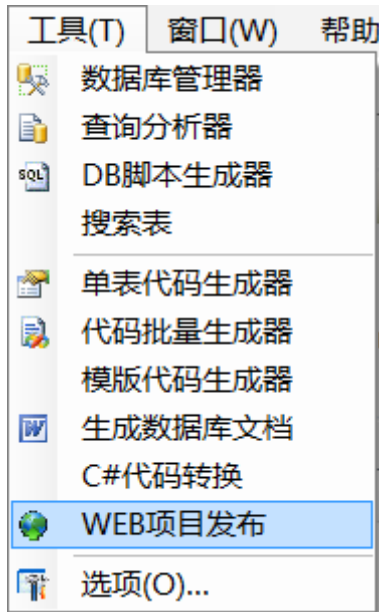
1. 搜索表功能



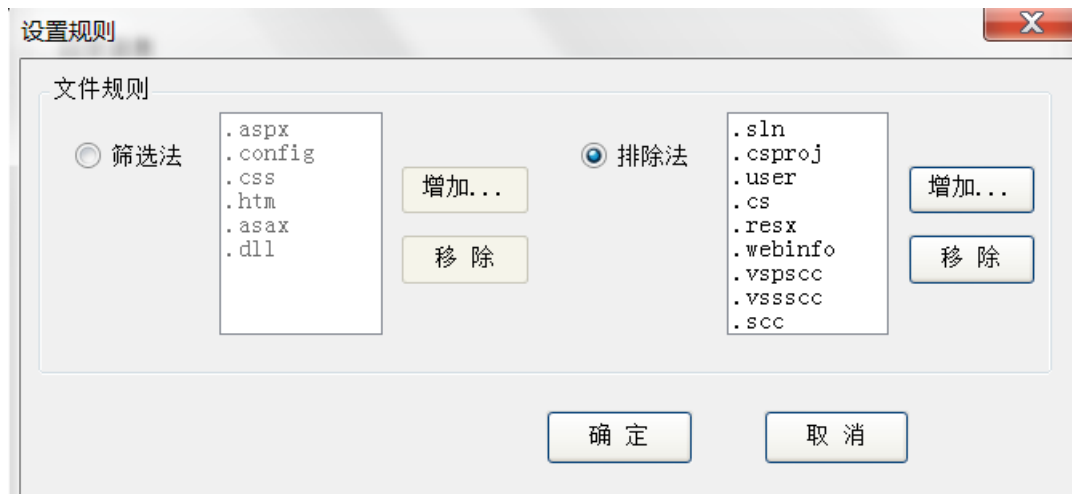
2. 项目发布功能

当 Web 项目开发完毕, 我们一般需把项目发布到一个测试服务器或者生产环境去, 但发布的时候, 只想发布我们需要发布的文件, 或者不想发布一些我们不需要的文件。虽然 VS. NET 自带的有发布功能, 但有时候觉得使用起来还不是很那么灵活方便, 动软提供了此功能, 让你更灵活, 简捷的发布网站文件。

选择菜单【工具】-【web 项目发布】



点击【设置】按钮，我们可以设置发布的规则。



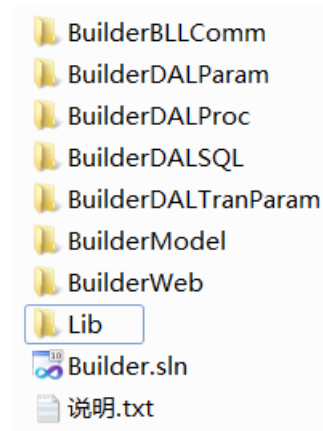
3. 代码转换功能

这是早期版本留下了的一个代码转换工具，可以支持 C#代码到 VB 代码的转换。

十四. 代码生成组件接口开发

动软.NET 代码生成器支持可扩展的代码生成插件，用户可以根据接口定制自己的代码生成插件，开发自己的代码生成方式，按需求进行代码生成。

该组件源码可以在下载的安装包里面看到或者去官方网站下载。



开发步骤：

1. 引用

Lib 文件夹下提供了接口类库，添加 4 个 dll 的引用：

LTP.CodeHelper.dll

LTP.IBuilder.dll

LTP.IDBO.dll

LTP.Utility.dll

2. 并在类中 using

```
using LTP.Utility;
using LTP.IDBO;
using LTP.CodeHelper;
```

3. 继承

DAL 数据访问层模板组件: `public class BuilderDAL : LTP.IBuilder.IBuilderDAL`

BLL 业务逻辑层模板组件: `public class BuilderBLL : LTP.IBuilder.IBuilderBLL`

4. 接口方法

具体参考插件源码项目。安装包中提供了插件的源码。

5. 使用配置

【工具】-【选项】-【组件管理】



在【代码参数】或代码生成器窗口，即可以看到如下插件列表：



常见问题（单击以下链接查看内容详细）

- 通过 [partial](#) 关键字扩展代码生成器生成的代码
- 动软左侧树菜单导航，无法正常显示？

- [动软新建项目（示例项目）如何登录？](#)
- [Access 用 OleDbParameter 无法更新数据解决方案](#)
- [动软 DBUtility 类库 DbHelperSQL 实现多数据库连接](#)
- [动软.net 代码生成器 启动报错退出解决方法](#)
- [Asp.net1.0 升级 ASP.NET 2.0 的几个问题总结](#)
- [示例项目源码无法创建对象实例的常见错误说明](#)