

## C/S 框架代码生成器

### ClassGeneratorV2.01 使用指南

By Jonny Sun at 2012-02-18

(doc v1.1)

ClassGeneratorV2.01 代码生成器主要功能 .....	2
一. 生成 ORM.....	2
1.1 ORM 主界面 .....	2
1.2 使用说明及参数设置 .....	3
1.2.1 项目主目录 .....	3
1.2.2 连接 SQL 服务器 .....	4
1.2.3 第一步: 取用户表和视图 .....	4
1.2.4 第二步: 取表结构 .....	6
1.2.5 表格栏位说明 .....	6
1.2.6 ORM_FieldAttribute 特性定义 .....	7
1.2.7 第三步: 生成 ORM 源码 .....	7
1.2.8 ORM 输出目录 .....	8
二. 生成 BLL .....	9
2.1 生成 BLL 功能主界面 .....	9
2.2 使用说明及参数设置 .....	9
三. 生成 DAL.....	10
3.1 生成 DAL 功能主界面 .....	10
3.2 使用说明及参数设置 .....	11
四. 生成的源码抢先看! .....	12
4.1 ORM 源码 .....	12
4.2 BLL 源码 .....	12
4.3 DAL 源码 .....	12
五. 请提意见或建议 .....	13

# ClassGeneratorV2.01 代码生成器主要功能

ClassGenerator 用于生成 ORM/BLL/DAL/窗体的源码。

## 一. 生成 ORM

### 1.1 ORM 主界面

代码生成器 v2.01 - C/S框架网 www.csframework.com

项目主目录: E:\CSFramework-WebService-CHS-3.0

生成->ORM 生成-> BLL 生成-> DAL 生成的代码 生成->数据字典窗体 生成->业务数据窗体 生成->报表窗体

连接SQL: Server=.\SQLEXPRESS;DataBase=CSFramework3.Normal;Integrated Security=True;

**步骤1:** 取用户表和视图 (请选择一张表)

用户表和视图: tb\_SO ☒ 主表 ☒ 数据库字符集

类名: tb\_SO \* (类名字头为"tb\_", 在这里可自定义)

**步骤2:** 取表结构

注释: ORM模型, 数据表:tb\_SO,由ClassGenerator自动生成

主键: SONO \* (必须填写!) 命名空间: CSFramework3.Models

列名	类型	长度	是否查询列	是否新增修改列	是否主键列	是否外键	是否流水号
ISID	int	4	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
SONO	varchar	20	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>
VerNo	varchar	2	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
CustomerCode	varchar	20	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
ReceiveDay	datetime	8	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
PayType	varchar	10	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
CustomerOrderNo	varchar	20	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Salesman	varchar	20	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Currency	varchar	4	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Amount	numeric	9	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
FinishingStatus	varchar	10	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
OrderFinishDay	datetime	8	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>

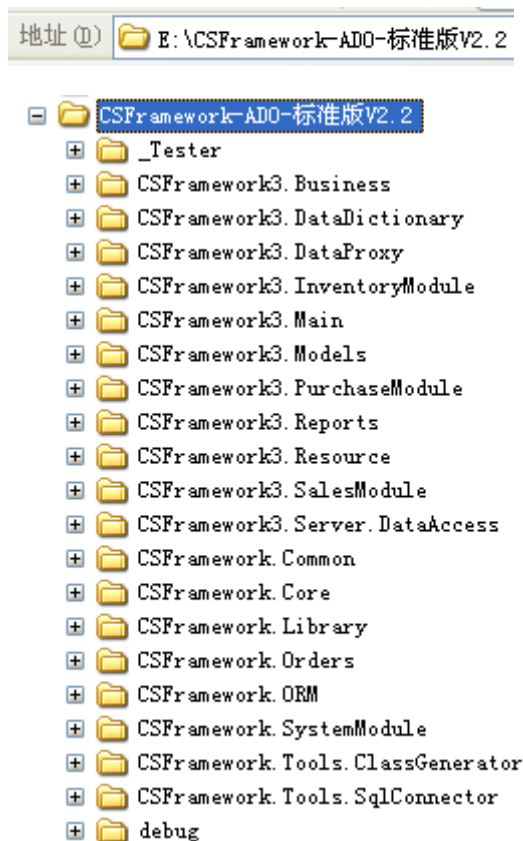
**ORM** 输出目录: E:\CSFramework-WebService-CHS-3.0\CSFramework3.Models\

**步骤3:** 生成表结构静态类 生成实体类(对象) ☒ 生成字段名常量 ☒ 生成代码后跳到预览页面 ☒ 输出到.cs文件

## 1.2 使用说明及参数设置

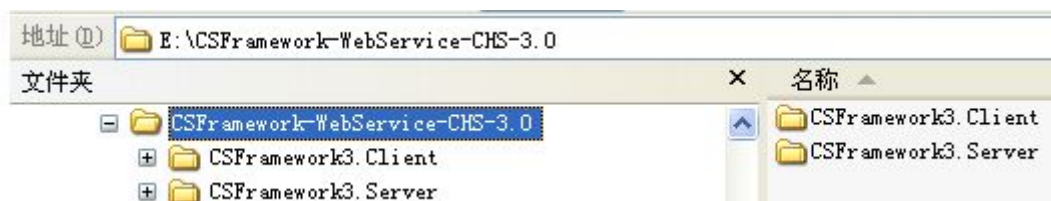
### 1.2.1 项目主目录

解决方案源代码的主目录，如 V2.2 的目录结构为：



项目主目录则定义：E:\CSFramework-ADO-标准版 V2.2

V3.0 的目录结构为：



项目主目录则定义：E:\CSFramework-WebService-CHS-3.0

**注：项目主目录仅用于参考目的，主要用于自动生成输出目录。**

如项目主目录为:

E:\CSFramework-WebService-CHS-3.0\CSFramework3.Client

则 ORM 的输出目录自动改为:

E:\CSFramework-WebService-CHS-3.0\CSFramework3.Client\CSFramework3.Models\

输出目录:

E:\CSFramework-WebService-CHS-3.0\CSFramework3.Client\CSFramework3.Models\

BLL 的输出目录为:

E:\CSFramework-WebService-CHS-3.0\CSFramework3.Client\BLL\_DataDict\

E:\CSFramework-WebService-CHS-3.0\CSFramework3.Client\BLL\_Business\

输出目录:

E:\CSFramework-WebService-CHS-3.0\CSFramework3.Client\CSFramework3.Business\BLL\_DataDict\

数据字典

生成数据字典BLL代码

输出目录:

E:\CSFramework-WebService-CHS-3.0\CSFramework3.Client\CSFramework3.Business\BLL\_Business\

业务单据

生成业务单据BLL代码

## 1.2.2 连接 SQL 服务器

连接SQL:

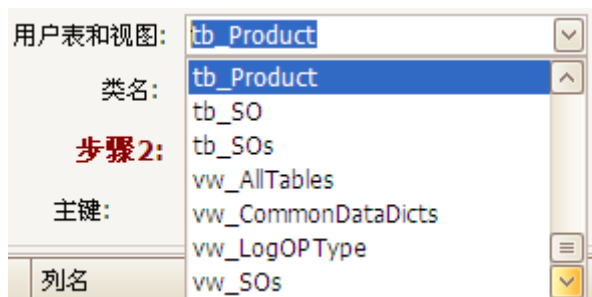
Server=.\SQLEXPRESS;DataBase=CSFramework3.Normal;Integrated Security=True;

建立连接后获取表结构, 然后通过表结构生成 ORM。

## 1.2.3 第一步: 取用户表和视图

- 取用户表和视图

当建立连接后即可加载用户表和视图方便用户选取, 如下图:



Copyright 2011 C/S 框架网 [www.csframework.com](http://www.csframework.com) 版权所有

- 主表

用于区分主表和明细表，假设业务仅一张表均视为主表，**明细表不能勾选！**

用户表和视图:  ☐ 主表 ☒ 数据库字符集

- 类名: 生成的 ORM 类名。

类名:  \* (类名字头为"tb\_", 在这里可自定义)

```
///<summary>
/// ORM模型, 数据表:tb_Product,由ClassGenerator自动生成
/// </summary>
[ORM_ObjectClassAttribute("tb_Product", "SONO", true)]
public sealed class tb_Product
{
}
```

## 1.2.4 第二步：取表结构

根据当前选择的数据表从数据库服务器获表结构。

**步骤2:** 取表结构      注释: ORM模型, 数据表:tb\_Product,由ClassGenerator自动生成

主键: ProductCode \* (必须填写!)      命名空间: CSFramework3.Models

列名	类型	长度	是否查询列	是否新增修改列	是否主键列	是否外键	是否流水号
Isid	int	4	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
ProductCode	varchar	20	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
ProductName	varchar	50	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
SellPrice	decimal	9	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Supplier	varchar	50	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Remark	varchar	100	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>

注释: ORM模型, 数据表:tb\_Product,由ClassGenerator自动生成

命名空间: CSFramework3.Models

- 主键：当前数据表的主键定义。
- 注释：类的注释。
- 命名空间：当前 ORM 所在的命名空间。

## 1.2.5 表格栏位说明

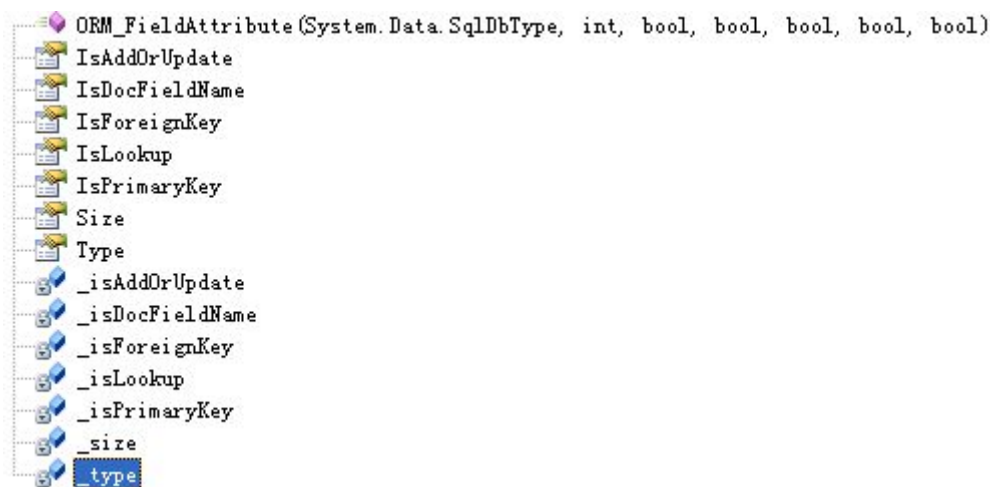
栏位标题	说明	ORM_FieldAttribute 属性
列名	对应字段名	-
类型	数据类型 <a href="#">SqlDbType</a>	Type
长度	数据长度定义	Size
是否查询列	如果是 <b>参考字段不支持数据更新</b> ，必须设为 True!	IsLookup
是否新增修改列	是否新增修改列， <b>只更新此值为 True 的栏位!</b>	IsAddOrUpdate
是否主键列	资料表主键， <b>每张表必须定义一个主键!</b>	IsPrimaryKey
是否外键	<b>主表一定是 False</b> ，明细表必须指定一个外键!	IsForeignKey
是否流水号	是否单据号码(如:SONO)，如 <b>此表自动生成单据号码，必须勾选!</b>	IsDocFieldName

## 1.2.6 ORM\_FieldAttribute 特性定义

注意: 后台数据更新程序是根据 ORM\_FieldAttribute 特性自动生成 SqlCommand 更新数据的, 所以正确配置表格中五个复选框关系到能否正常更新数据。

ORM\_FieldAttribute 类属性的值是跟据用户在表格中勾选设置自动生成的!

构造器传递的参数是表格中勾选的设置, True/False, 请参考表格栏位说明。



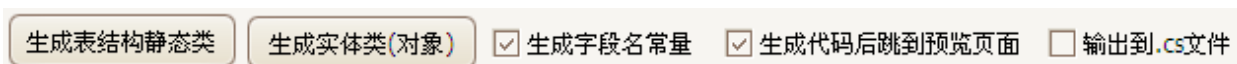
## 1.2.7 第三步: 生成 ORM 源码

ORM 及后台数据自动更新程序是开发框架核心功能之一, 为了以后扩展及增强灵活性, 开发框架设计了 IGenerateSqlCommand 接口并实现了两个具体的类, 跟据不同的 ORM 模型自动生动 SqlCommand, 以下是开发框架已实现的两个具体的类:

1. GenerateSqlCommandByTableFields 由表结构静态类自动生成 SqlCommand.
2. GenerateSqlCommandByObjectClass 由实体类(对象)自动生成 SqlCommand.

有了 IGenerateSqlCommand 接口实现了超强兼容能力, 更重要的是: 支持您自己开发 ORM 模型! 如 XML Mapping (类似 Nhibernate 架构下的 XML 定义), 强大到吐血! 伟大到壮烈!

ClassGenerator 支持生成两种类型的 ORM 类, 如下图:





- **生成表结构静态类:**

只是一个静态类, 仅生成了字段的常量名, 无任何属性和方法。

此类型的 ORM, 在数据层的 CreateSqlGenerator 方法必须创建 `GenerateSqlCmdByTableFields` 对象。

- **生成实体类(对象)**

实体类, 生成了字段的常量名和所有字段属性。可实例化为具体对象。  
如: User, Customer, Order 等。

此类型的 ORM, 在数据层的 CreateSqlGenerator 方法必须创建 `GenerateSqlCmdByObjectClass` 对象。

- 生成字段常量名  
勾选此复选框仅对生成实体类(对象)有效, 如勾选则生成字段常量名, 反之仅生成字段属性。
- 生成代码后跳到预览页面  
当生成代码自动跳到预览页面。
- 输出到.cs 文件  
当生成代码自动保存到.cs 文件。

## 1.2.8 ORM 输出目录

将生成的源码保存到输出目录, 文件名为类名。

输出目录:

E:\CSFramework-WebService-CHS-3.0\CSFramework3.Client\CSFramework3.Models\



## 二. 生成 BLL

### 2.1 生成 BLL 功能主界面

The screenshot shows the 'ClassGenerator v2.01 - C/S Framework' application window. The title bar includes the website 'www.csframework.com'. The main interface has a tabbed menu at the top with options: '生成->ORM', '生成->BLL' (selected), '生成->DAL', '生成的代码', '生成->数据字典窗体', '生成->业务数据窗体', and '生成->报表窗体'. Below the tabs, the '生成业务逻辑层代码' (Generate Business Logic Layer Code) section is active. It contains several input fields and a list box:

- 具体名称:** SO (Example: Customer)
- BLL类名:** blISO (Example: blCustomer)
- DAL类名:** dalISO (Example: dalCustomer)
- ORM模型:** tb\_SO (Example: tb\_Customer)
- 业务层的命名空间:** CSFramework3.Business
- 引用的名字空间:** A list box containing: CSFramework3.Models, CSFramework.Common, CSFramework.Core.Interfaces, CSFramework.Core.Log, CSFramework3.Server.DataAccess, CSFramework.Core, and CSFramework.Core.BLL\_Base. A note below says '(System.\* 除外) (断行符分开)'.

Below these fields is the '业务单据BLL选项' (Business Document BLL Options) section:

- 窗体名称:** frmSO (Example: frmCustomer)
- ☐ 版本历史记录(ISupportBusinessLog)

On the right side, there are two checked checkboxes: '输出到.cs文件' (Output to .cs file) and '生成代码后跳到预览页面' (Jump to preview page after generating code). At the bottom, there are two output directory fields and buttons:

- 输出目录:** E:\CSFramework-ADO-标准版V2.2\Source\CSFramework3.Business\BLL\_DataDict\ (Button: 数据字典 / Generate Data Dictionary BLL code)
- 输出目录:** E:\CSFramework-ADO-标准版V2.2\Source\CSFramework3.Business\BLL\_Business\ (Button: 业务单据 / Generate Business Document BLL code)

### 2.2 使用说明及参数设置

- 具体名称: 即业务名称, 比如: User, Customer, SalesOrder 等。
- BLL 类名: bl+具体名称, 自动生成, 可修改。
- DAL 类名: 当前 BLL 层引用的 DAL 层。 dal+具体名称, 自动生成, 可修改。
- ORM 模型: tb\_+具体名称, 自动生成, 可修改。
- 业务层的命名空间: 生成当前业务逻辑类所在命名空间, 可修改。

- 引用的名字空间：生成当前业务逻辑类所依赖的名字空间。
- 业务单据 BLL 选项，当勾选版本历史记录时需要输入窗体名称。

业务单据BLL选项	
窗体名称:	<input type="text" value="frmSO"/> 如:frmCustomer
<input type="checkbox"/> 版本历史记录(ISupportBusinessLog)	

- 生成数据字典 BLL 代码。数据字典 BLL 继承 `bllBaseDataDict`
- 生成业务单据 BLL 代码。数据字典 BLL 继承 `bllBaseBusiness`

## 三. 生成 DAL

### 3.1 生成 DAL 功能主界面

The screenshot shows the 'ClassGenerator V2.01 - C/S Framework' application window. The title bar includes the version and website 'www.csframework.com'. The main interface has a tabbed menu at the top with options: '生成->ORM', '生成->BLL', '生成->DAL' (selected), '生成的代码', '生成->数据字典窗体', '生成->业务数据窗体', and '生成->报表窗体'. Below the tabs, the '生成数据访问层代码' (Generate Data Access Layer Code) section is active. It contains several input fields: '具体名称' (Specific Name) set to 'SO' (example: 'Customer'), 'DAL类名' (DAL Class Name) set to 'dalSO' (example: 'dalCustomer'), and 'ORM模型' (ORM Model) set to 'tb\_SO' (example: 'tb\_Customer'). The '数据层的命名空间' (Data Layer Namespace) is set to 'CSFramework3.Server.DataAccess'. A list of '引用的名字空间' (Referenced Namespaces) includes 'CSFramework3.Models', 'CSFramework.Common', 'CSFramework.ORM', and 'CSFramework.Core.DAL\_Base'. The '输出目录' (Output Directory) is set to 'E:\CSFramework-ADO-标准版V2.2\Source\CSFramework3.Server.DataAccess\DAL\_DataDict\'. On the right, there are checkboxes for '输出到.cs文件' (checked) and '生成代码后跳到预览页面' (unchecked). Below this, there are two buttons: '生成数据字典DAL代码' (Generate Data Dictionary DAL Code) and '生成业务单据DAL代码' (Generate Business Document DAL Code). The interface also features a logo for 'C/S 框架网' and the text 'C#.Net快速开发平台'.

## 3.2 使用说明及参数设置

- 具体名称：即业务名称，比如：User, Customer, SalesOrder 等。
- DAL 类名：dal+具体名称，自动生成，可修改。
- ORM 模型：tb\_+具体名称，自动生成，可修改。
- 数据层的命名空间：生成当前数据层类所在的命名空间，可修改。
- 引用的名字空间：当前数据层所依赖的名字空间。
- 生成数据字典 DAL 代码
- 生成业务单据 DAL 代码

## 四. 生成的源码抢先看！

### 4.1 ORM 源码

<http://www.csframework.com/archive/5/arc-5-20110514-1481.htm>

### 4.2 BLL 源码

<http://www.csframework.com/archive/5/arc-5-20110514-1482.htm>

### 4.3 DAL 源码

<http://www.csframework.com/archive/5/arc-5-20110514-1483.htm>

## 五. 请提意见或建议

QQ: 23404761, 1980854898

Email: [jonnysun@csframework.com](mailto:jonnysun@csframework.com)

C/S 框架网

[www.csframework.com](http://www.csframework.com)

Feedback 意见反馈 – 您的意见是我们进步的根源!

<http://www.csframework.com/feedback.htm>

Feedback

意见反馈 – 您的建议是我们进步根源!