零基础 30 分钟开启你的快速开发之旅

1. 前言

接触 AgileEAS.NET SOA 中间件平台（以下简称 EAS.NET 平台）有 4 个多月时间，经过试用认为可以把它作为一个开发的基础平台，开发团队可以把开发的重点放在需求的把控和项目的交付上，从而节省大量的时间，提高项目的开发、交付效率，降低对项目团队的深层技术要求，更重要的一点是 EAS.NET 平台的开发团队持续不断地维护和改进平台以及对反馈问题的快速反应，使我对平台的持续发展充满信心。

由于 EAS.NET 平台的资料比较多，需要花费较多的时间才能够初步了解平台并能够使用平台开始开发，所以，在此，我把学习 EAS.NET 平台的过程总结了一下，形成本篇短文，希望能够让初次接触 EAS.NET 平台的朋友能够用 30 分钟时间，跟着本文案例实际操作一遍，对 EAS.NET 平台有一个真实的体验，节省大家的时间，也算对 EAS.NET 平台的一点回报，希望有更多的人了解它的优势并真正用好它，为 EAS.NET 平台的使用者带来价值，也为 EAS.NET 平台的开发者带来效益，最终实现合作共赢的美好结局。

说明：此平台由此平台来自魏琼东开发所有，可免费进行商业化开发，不开源，可有偿提供源码和提供个性化定制服务。详细内容见作者网站 http://www.smarteas.net/

注意：本文中涉及平台的功能操作，是通过运行 Publish 目下的平台客户端 EAS.WinClient.Start.exe，以 Administrator 账户、密码 sa 登录系统进行操作的，如输入字典、函数管理、报表管理、角色管理、账户管理、模块管理、导航分组等。

2. 基础环境

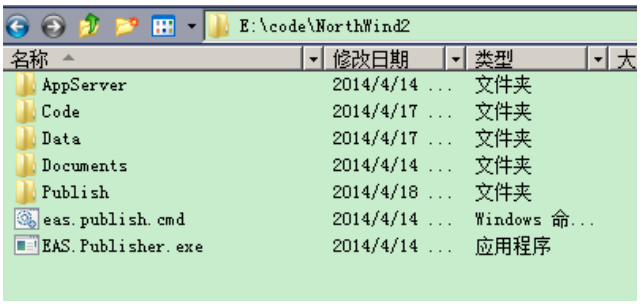
开发环境：windows server 2008 64 位、SQL SERVER 2008R2
开发工具：VS 2012
以上为本人的开发测试环境，如果与以上环境不同，遇到问题时请与平台作者联系。
在正式开始开发之前，首先下载案例代码及相关程序包，下载地址：
直接下载：http://42.121.30.77/downloads/eas/Northwind.rar。
SVN 更新：http://42.121.30.77:8080/svn/Northwind，登录用户:eas，密码 eas.
在开始工作之前，首先对 VS2012 添加 EAS 平台需要的控件。
在工具箱中添加选项卡【EAS】
通过【选择项】添加 EAS.Windows、EAS.Data.Controls、EAS.GridReport.Controls 三个组件中的控件。如下图所示：

案例测试环境配置：

1、 在 E 盘新建目录 code

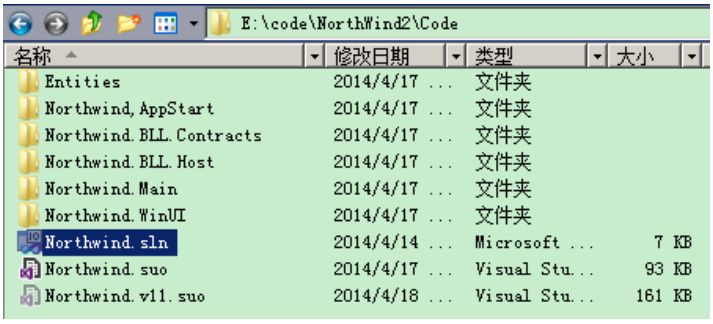
2、 在 E 盘 code 目录下新建目录 Northwind2

3、 把下载的压缩包解压到 Northwind2 下（如果是用 svn 更新获取，则把相应目录内容复制到此处），此时项目目录结果如下：

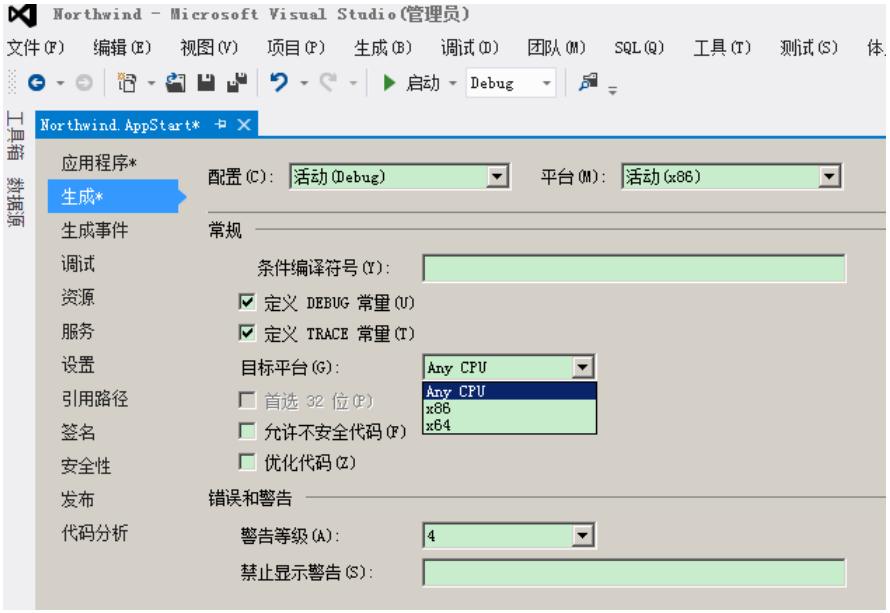


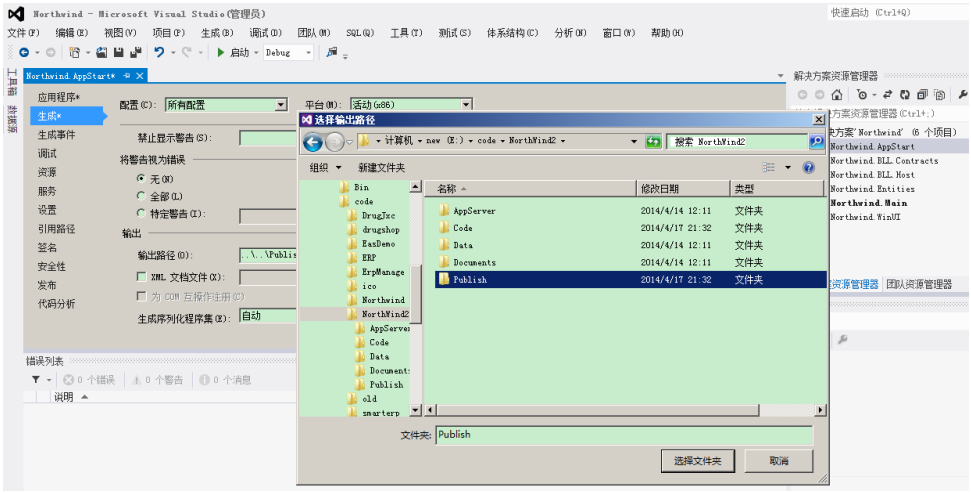
4、 在 VS 中打开案例

进入目录 code，双击 Northwind.sln

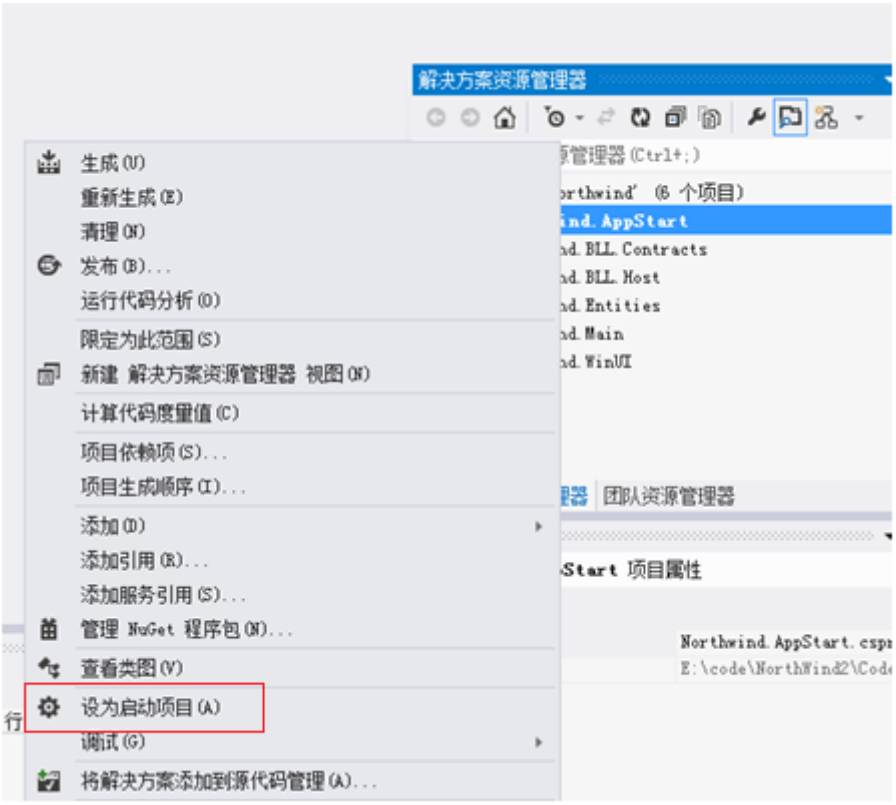


在每个项目上右键/属性，修改生成页签下的目标平台为 Any CPU，输出路径为 E:\code\Northwind2\Publish，如下图所示：

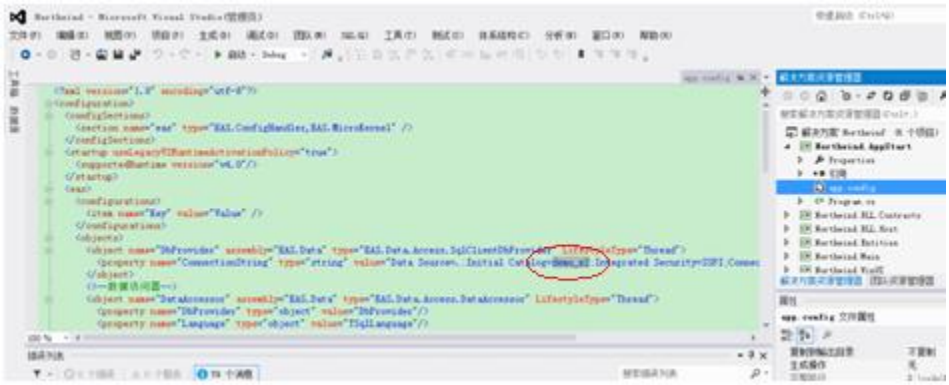




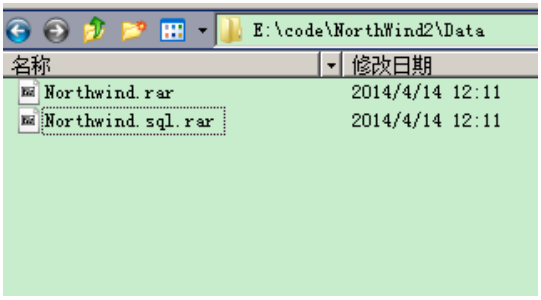
右键 Northwind.AppStart，选择【设为启动项目】，便于调试。


解决方案资源管理器
生成(U)
重新生成(E)
清理(N)
发布(B)...
运行代码分析(O)
限定为此范围(S)
新建 解决方案资源管理器 视图(N)
计算代码度量值(C)
项目依赖项(S)...
项目生成顺序(I)...
添加(D)
添加引用(R)...
添加服务引用(S)...
管理 NuGet 程序包(N)...
查看类图(V)
设为启动项目(A)
调试(G)
将解决方案添加到源代码管理(A)...
团队资源管理器
Start 项目属性

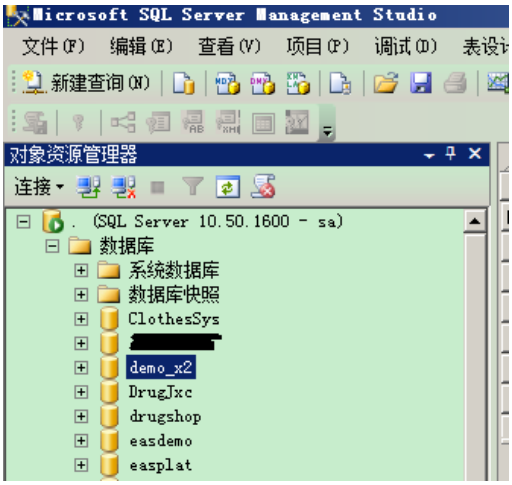
打开 Northwind.AppStart 下的 app.config 配置文件，修改数据库名称为实际的数据库名称，此处为 demo_x2。



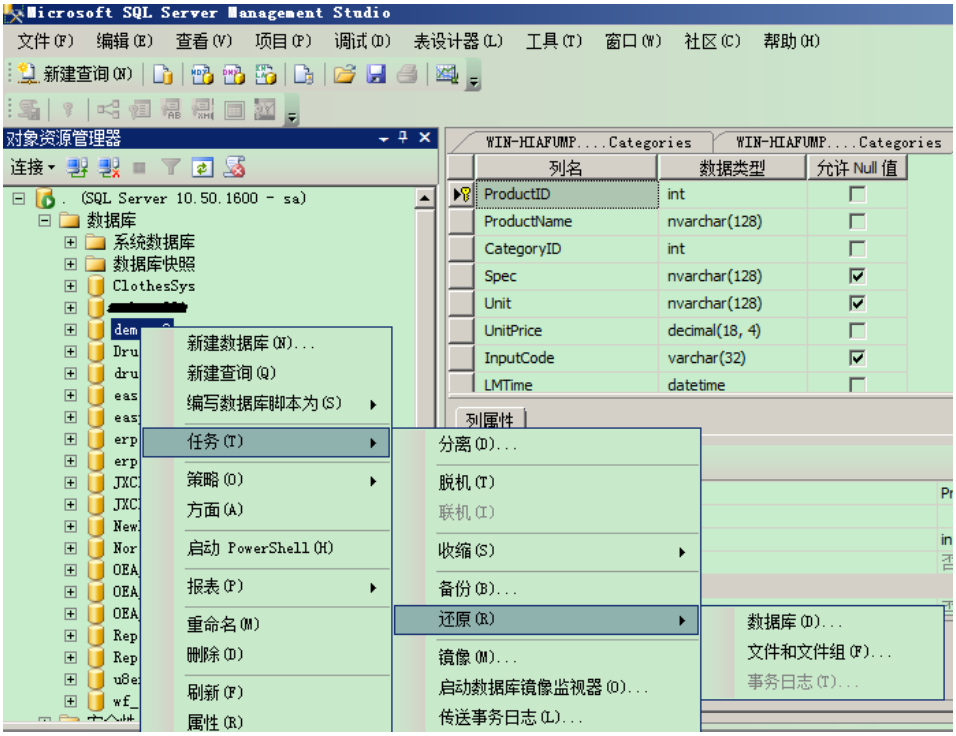
恢复数据库：解压 Data 目录下的 Northwind.rar 文件到当前目录，获得数据库备份文件 Northwind.bak。



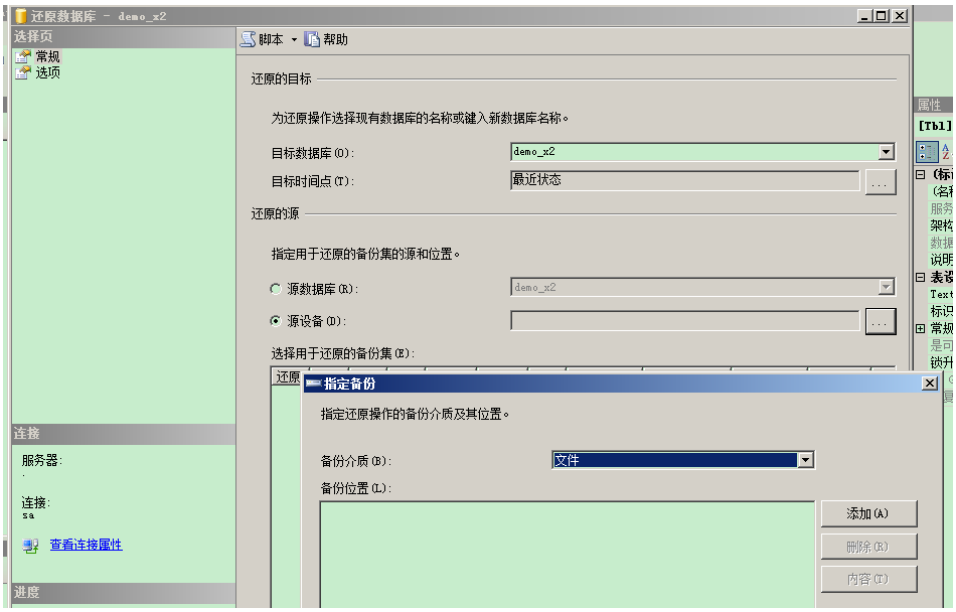
打开 sql server 管理器，新建数据库 demo_x2。

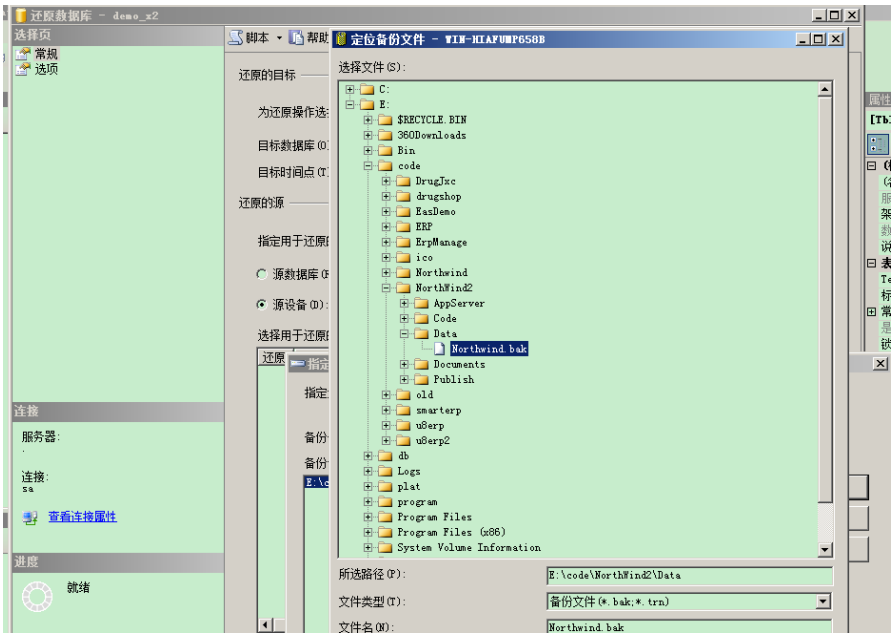


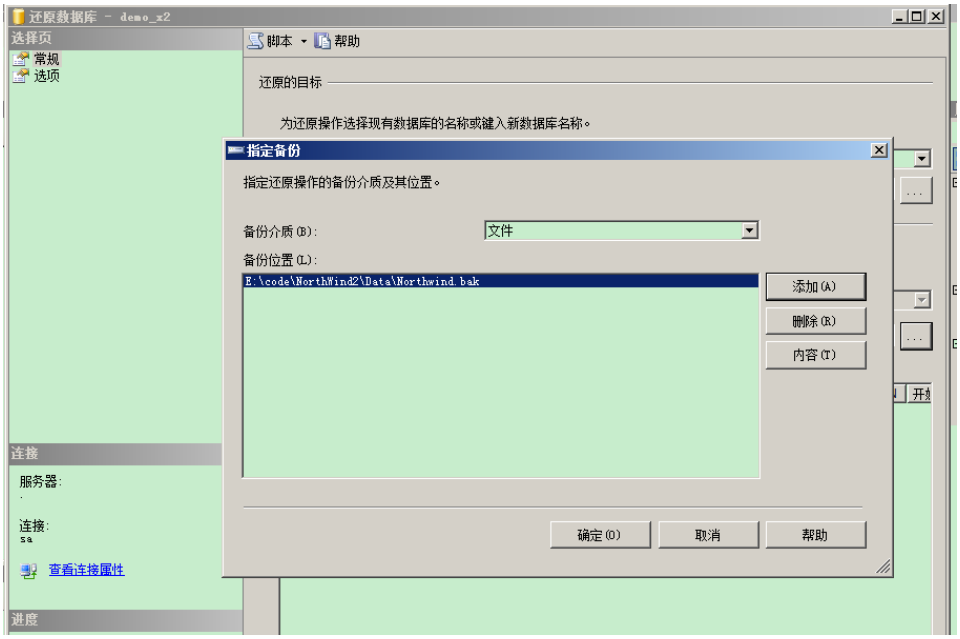
右键 demo_x2，选择任务/还原/数据库

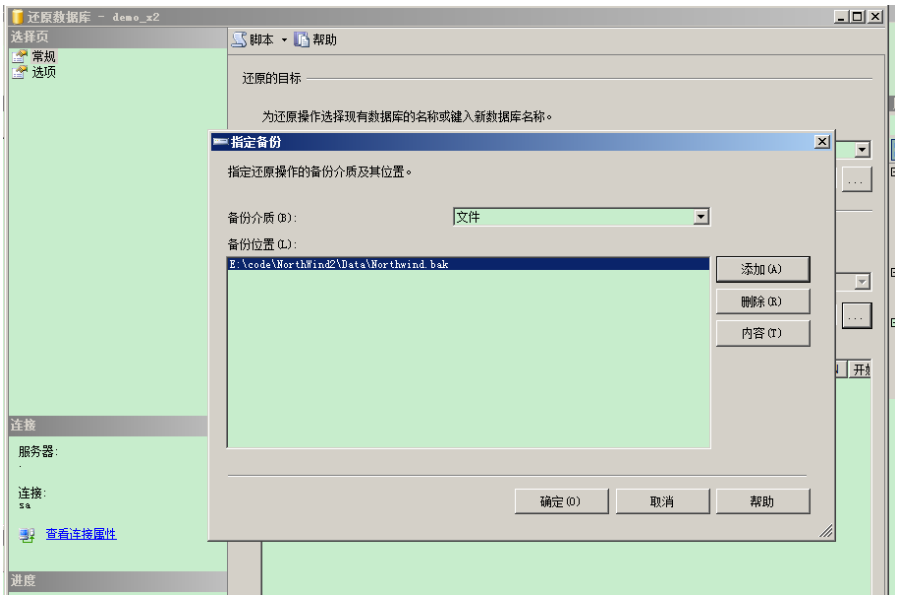


选择源设备，添加 E:\code\Northwind2\Data 下的 Northwind.bak

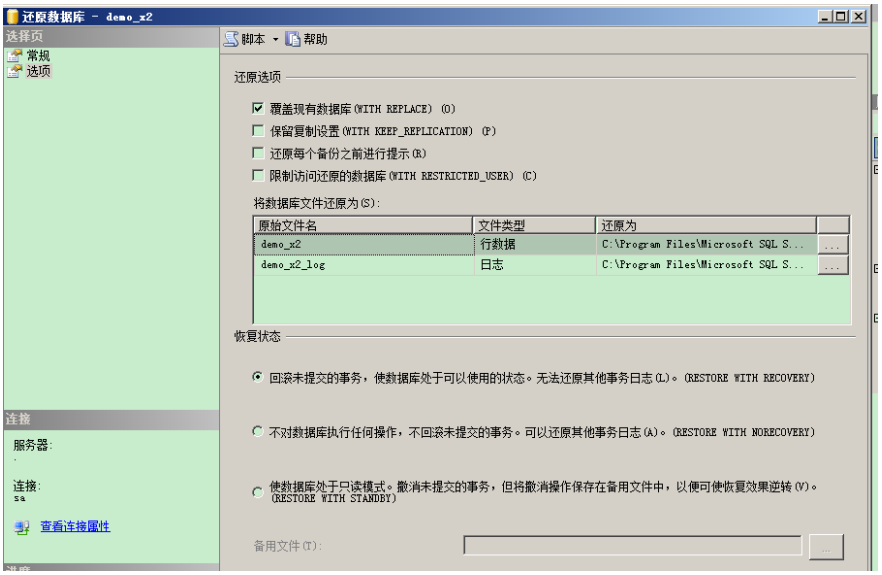



还原数据库 - demo_x2
选择页
常规
选项
脚本
帮助
还原的目标
为还原操作选
目标数据库(D)
目标时间点(T)
还原的源
指定用于还原
源数据库(F)
源设备(D)
选择用于还原
还原
指定
备份
备份
连接
服务器:
连接:
sa
查看连接属性
进度
就绪
定位备份文件 - WIN-HIAFUMP658B
选择文件(S):
C:
E:
$RECYCLE.BIN
360Downloads
Bin
code
DrugJxc
drugshop
EasDemo
ERP
ErpManage
ico
Northwind
NorthWind2
AppServer
Code
Data
Northwind.bak
Documents
Publish
old
smarterp
u8erp
u8erp2
db
Logs
plat
program
Program Files
Program Files (x86)
System Volume Information
所选路径(P):
E:\code\NorthWind2\Data
文件类型(T):
备份文件(*.bak;*.trn)
文件名(N):
Northwind.bak

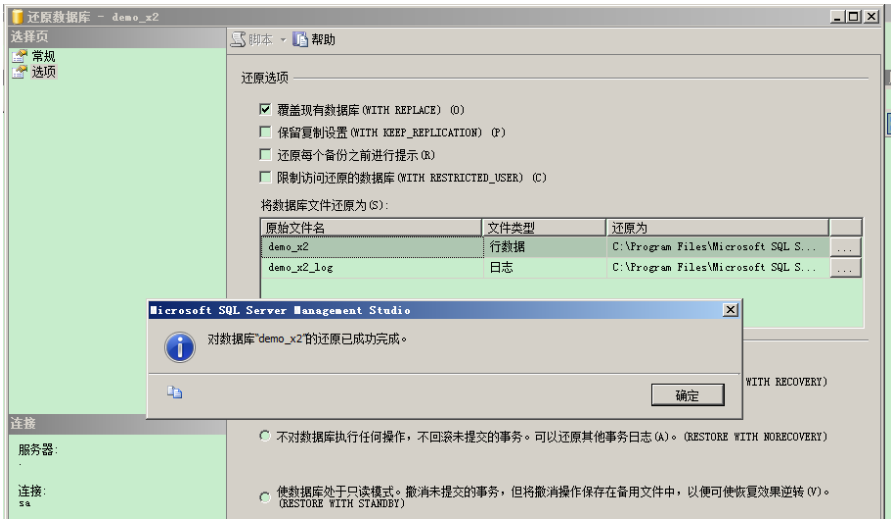

还原数据库 - demo_x2
选择页
常规
选项
脚本
帮助
还原的目标
为还原操作选择现有数据库的名称或键入新数据库名称。
指定备份
指定还原操作的备份介质及其位置。
备份介质(B):
文件
备份位置(L):
E:\code\NorthWind2\Data\Northwind.bak
添加(A)
删除(R)
内容(T)
确定(O)
取消
帮助
连接
服务器:
连接:
sa
查看连接属性
进度



选中选项标签，选中【覆盖现有数据库】，然后开始还原。



完成数据库还原。



分别在各个项目上右键进行重新生成，顺序为：

Northwind.Entity

Northwind.BLL.Contracts

Northwind.BLL.Host

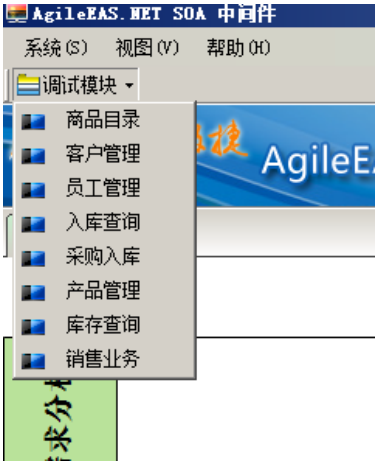
Northwind.WinUI

Northwind.AppStart

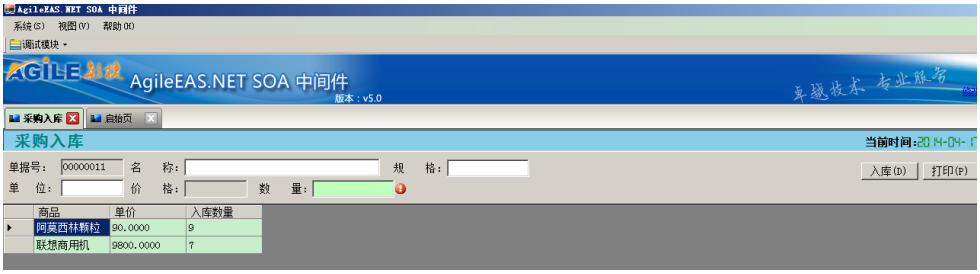
完成后，启动程序进行调试。

用户名：james，密码：sa

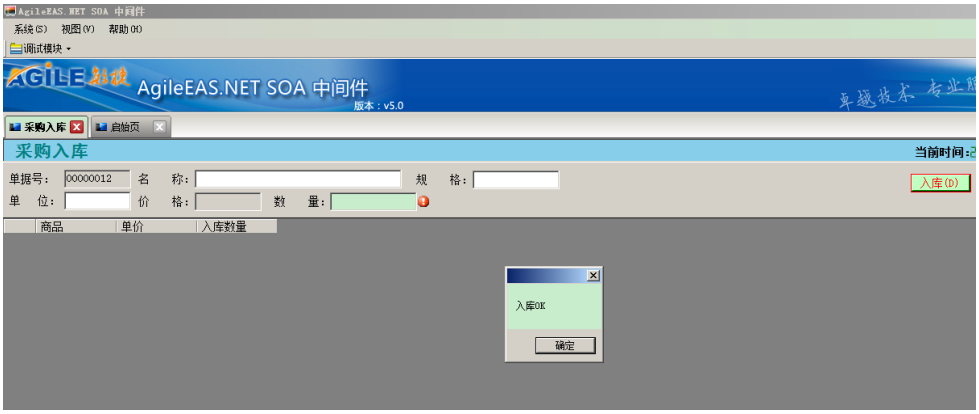
进入系统后，左上角有调试模块，点击后展开案例所有功能。



测试采购入库：在名称框中按空格键，弹出商品选择框，选中需要的商品，录入数量后回车，则入库单明细增加一条，如下图所示：



录入完成后，点击入库按钮，完成采购入库单。

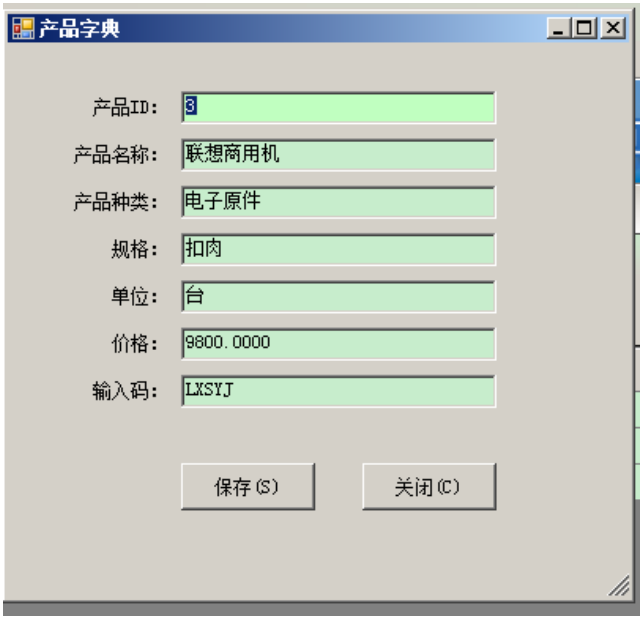


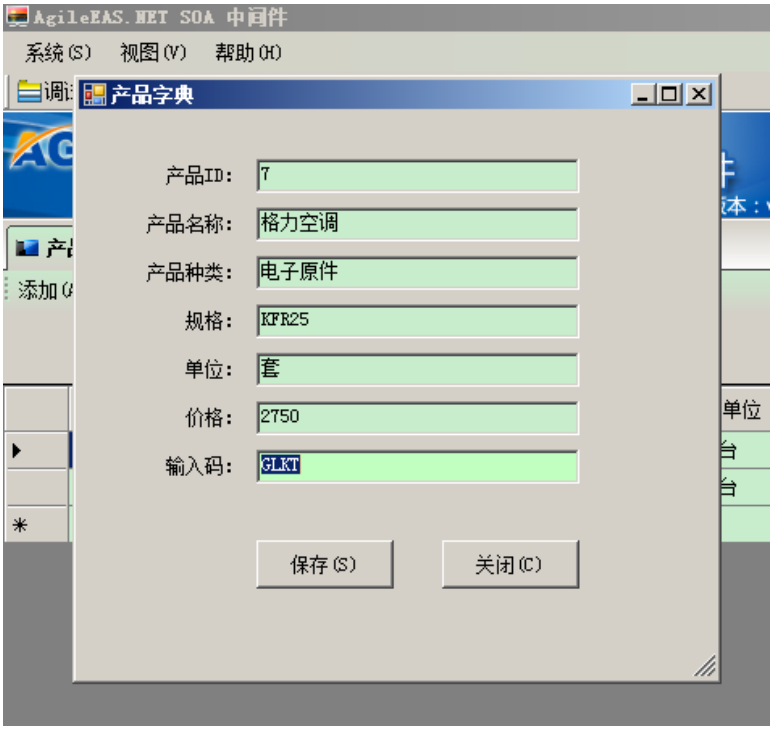
测试产品管理：

在此模块进行产品资料的维护，新增、修改、删除等。

进入模块后，首先点击查询按钮，系统会把已有的产品列出。

选中一条可以对其进行修改或删除。





3. 开始开发

建立开发目录：新建 E：\code\Northwind 目录，然后在 Northwind 下新建以下目录：

AppServer　　系统程序发布目录

Code　　程序代码

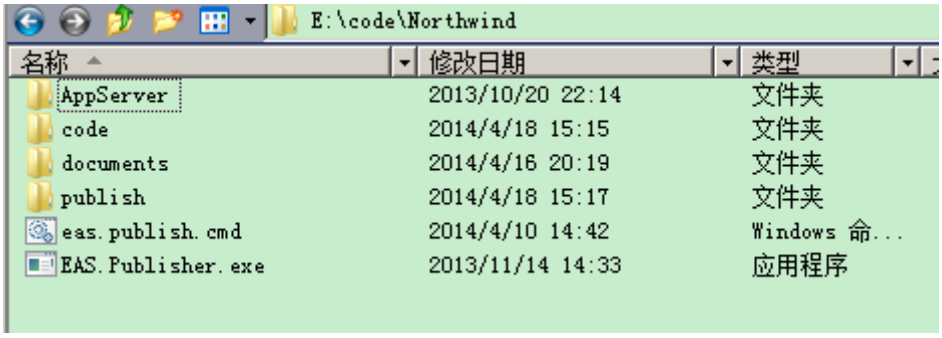
把案例代码 Northwind2 目录下的 eas.publish.cmd、EAS.Publisher.exe 复制到 Northwind 目录下；

把案例代码 Northwind2 目录下的 Documents、Publish 目录整个复制到 Northwind 目录下。

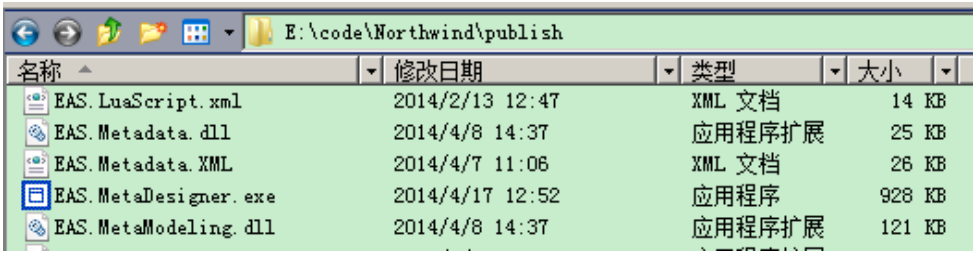
Documents　　文档

Publish　　引用的 dll、系统工具文件及开发编译后的程序文件

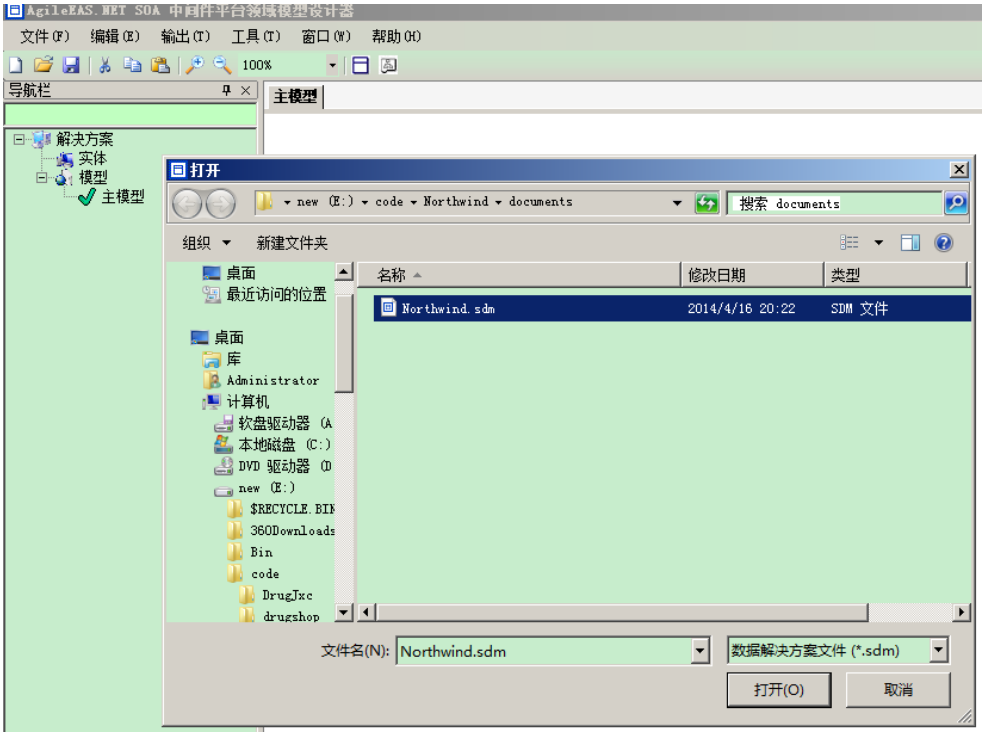
完成后的目录结构如下：



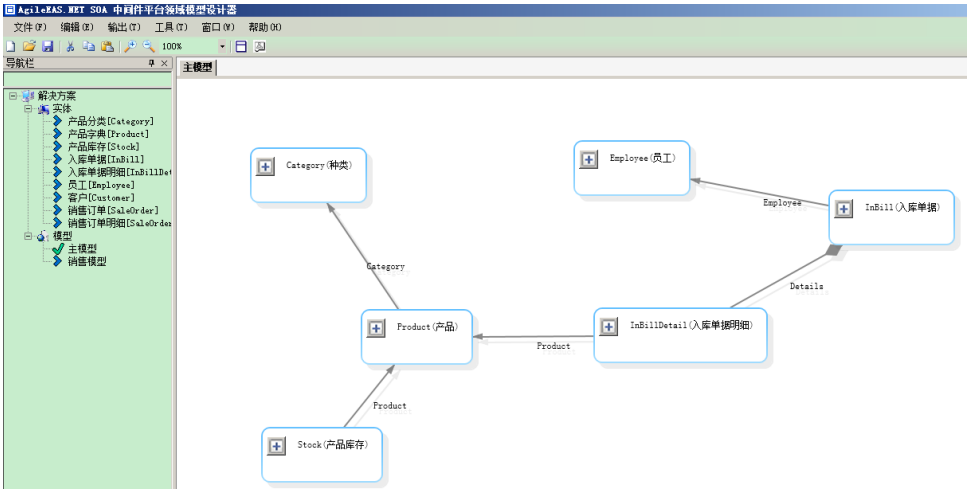
3.1. 生成数据实体代码



进入 publish 目录，双击 EAS.MetaDesigner.exe，打开领域模型设计器，在领域模型设计器中打开 documents 目录下的 Northwind.sdm 领域设计文档，如下图所示：



领域模型设计器可以展示各数据实体间的逻辑关系

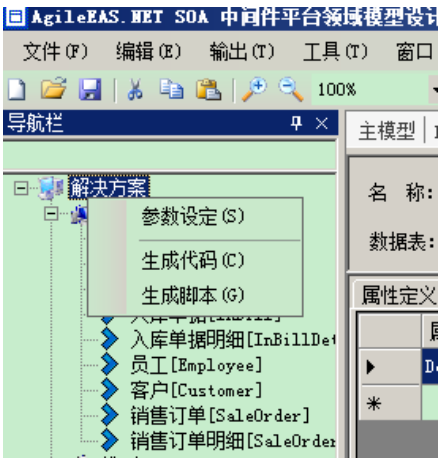

AgileEAS.NET SOA 中间件平台领域模型设计器
文件(F) 编辑(E) 输出(T) 工具(T) 窗口(W) 帮助(H)
100%
导航栏
解决方案
实体
产品分类[Category]
产品字典[Product]
产品库存[Stock]
入库单据[InBill]
入库单据明细[InBillDe
员工[Employee]
客户[Customer]
销售订单[SaleOrder]
销售订单明细[SaleOrde
模型
主模型
销售模型
主模型
Category(种类)
Employee(员工)
InBill(入库单据)
Category
Employee
Details
Product(产品)
InBillDetail(入库单据明细)
Product
Product
Stock(产品库存)

通过设计器可以设定数据对象的详细内容

可以设定数据对象之间的引用关系，如销售订单上引用的员工、客户等

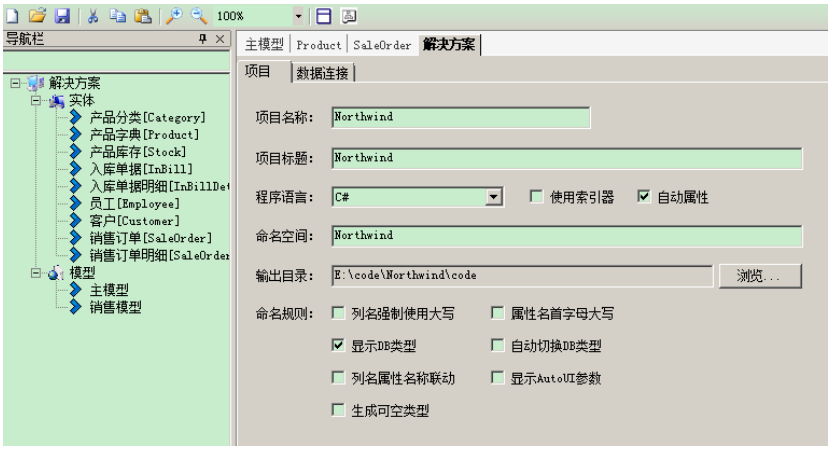
可以设定数据对象之间的父子关系，如销售订单和销售订单明细

鼠标右键点击解决方案，点击【参数设定】

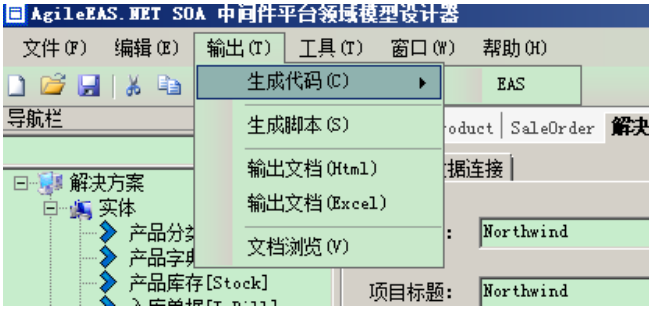


设定项目名称、项目标题都为 Northwind，程序语言选择 C#，选定自动属性，命名空间：

Northwind，设定输出目录为 E:\code\Northwind\code



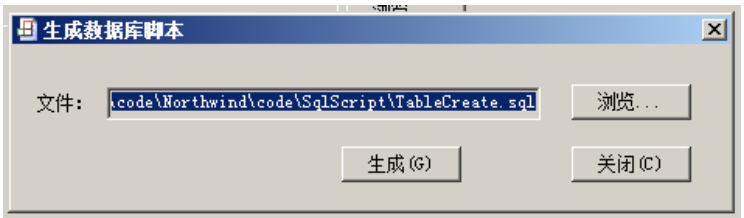
点击输出/生成代码/EAS 或在解决方案上右键，选择生成代码



选中生成解决方案，点击生成。

3.2. 生成数据库脚本

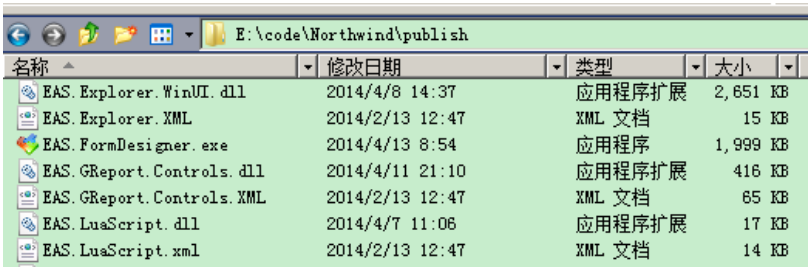
点击【生成脚本】按钮，创建数据库结构创建脚本。



3.3. 生成窗体代码

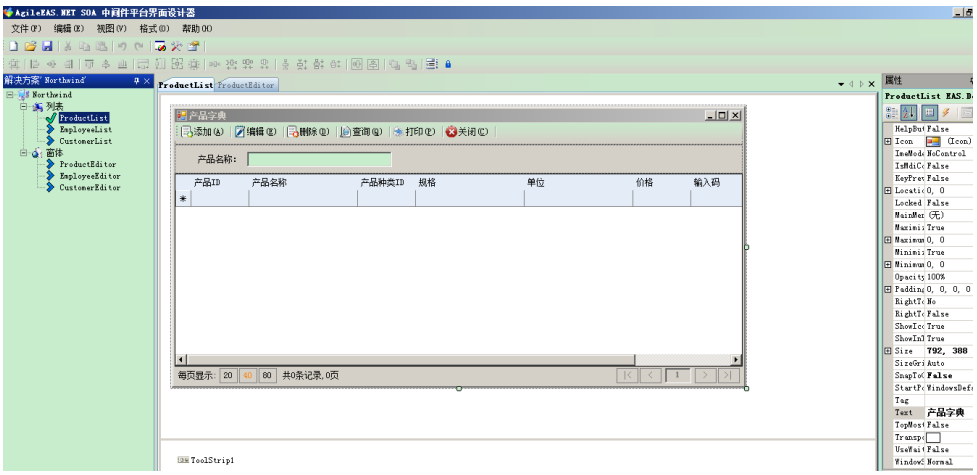
平台提供了窗体设计工具 EAS.FormDesigner，可以辅助快速生成列表类单据和录入类单据。案例中附带有例子，有兴趣的可以研究一下，本文操作采用手工创建窗体，手工编写代码方

式，便于读者能够把握使用平台时程序代码和普通程序代码的区别。

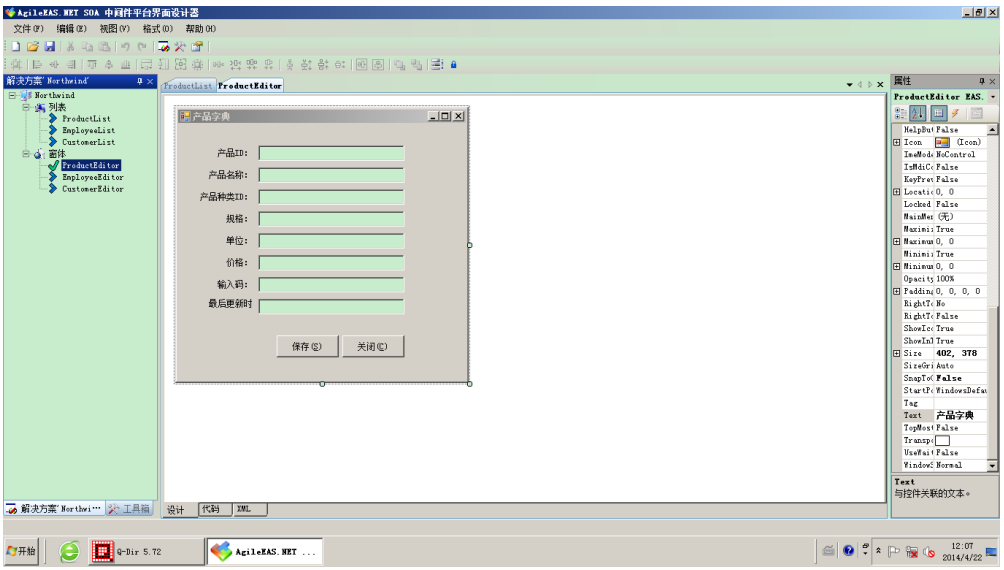


3.3.1. 列表窗体

启动 FormDesigner，打开 documents 下的 Northwind.sdm，结果如下：

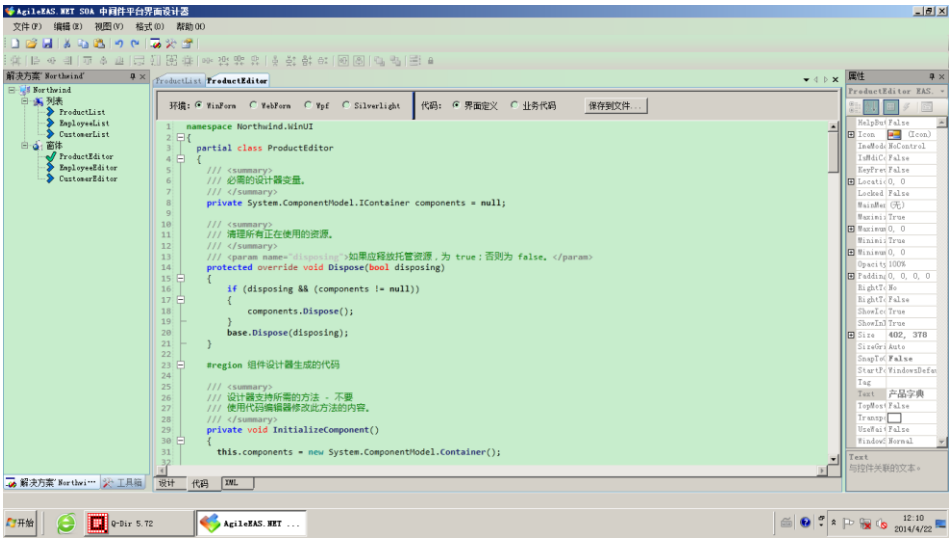


3.3.2. 编辑窗体

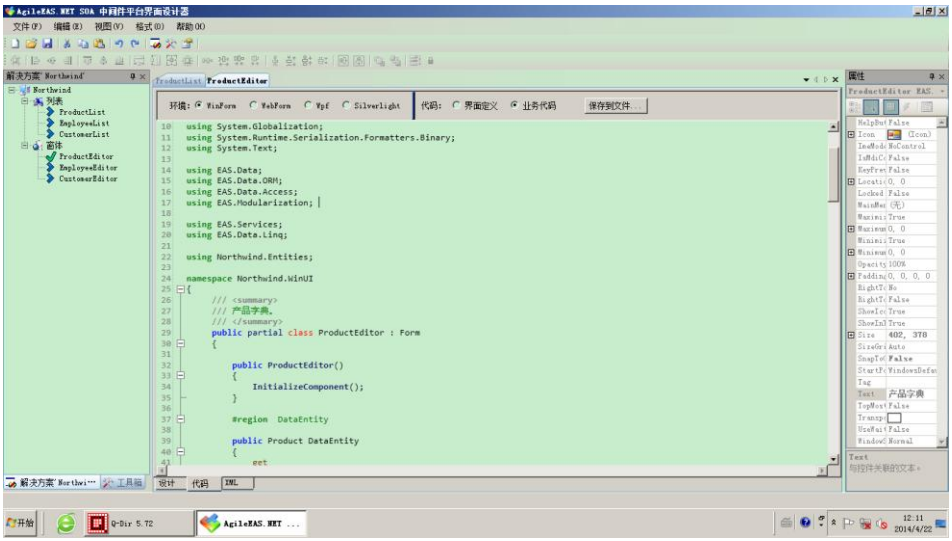


平台支持生成 winform、webform、wpf、silverlight 四种不同的界面代码。

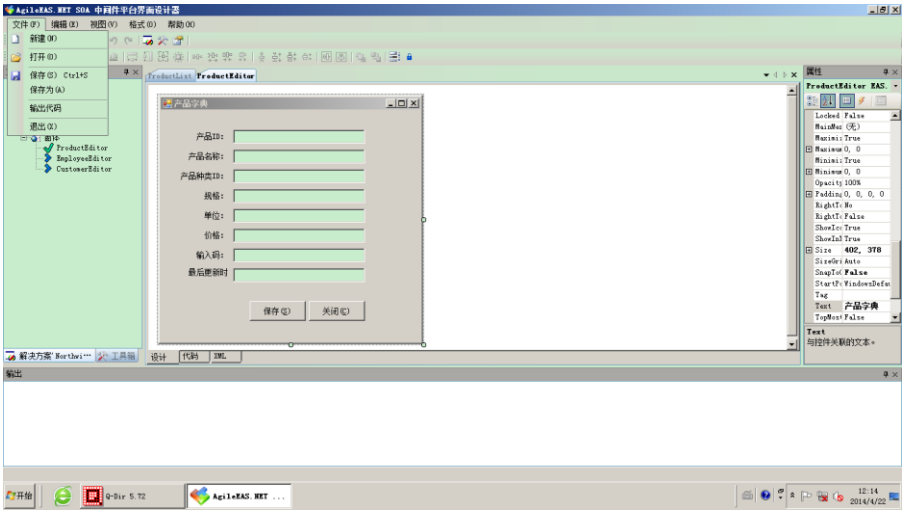
界面代码为系统自动生成：



在此可以进行业务代码的编写，完成后，



代码完成后，通过文件/输出代码，然后在 VS 中打开解决方案进行代码的调试和修改。



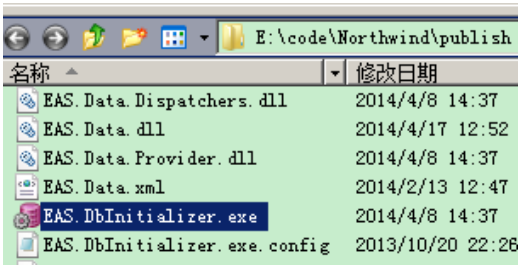
3.4. 创建数据库

3.4.1. 新建数据库

通过 sql server 管理器创建新数据库 Northwind。

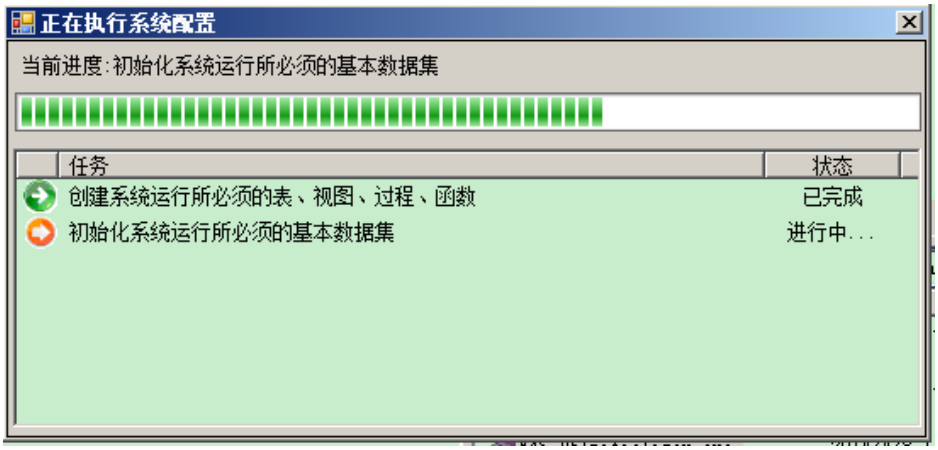
3.4.2. 维护数据库

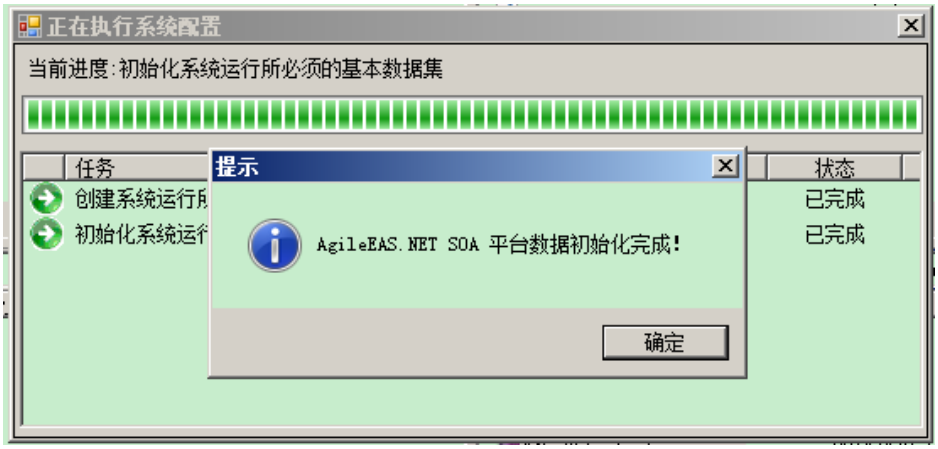
启动平台数据库初始化工具 EAS.DbInitializer



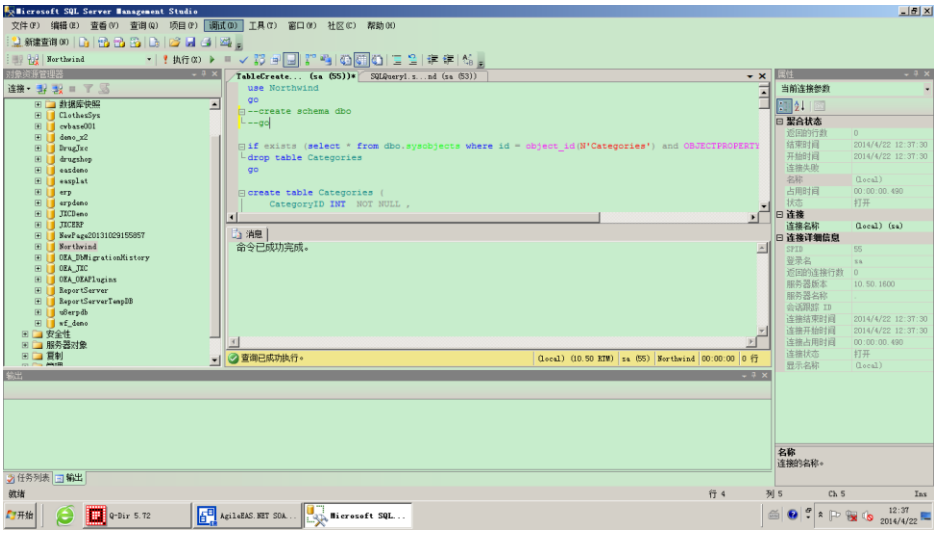
系统支持四种数据库系统，这里我们选择 sql server，下一步：

指定服务器，如果是本机则输入.即可。选中集成身份验证，点击测试链接，系统提示测试连接成功。然后，通过数据库下拉框选中新创建的 Northwind。也可以使用该工具直接建立新的数据库。点击完成，系统开始对数据库进行初始化。



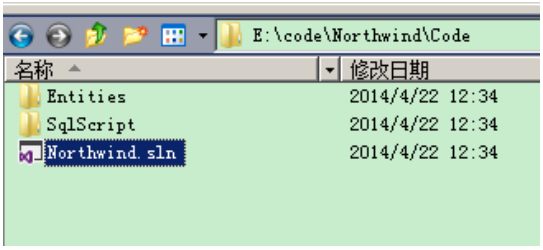


在查询分析其中执行 3.2.生成的数据库脚本，执行前注释掉第 3、4 行。



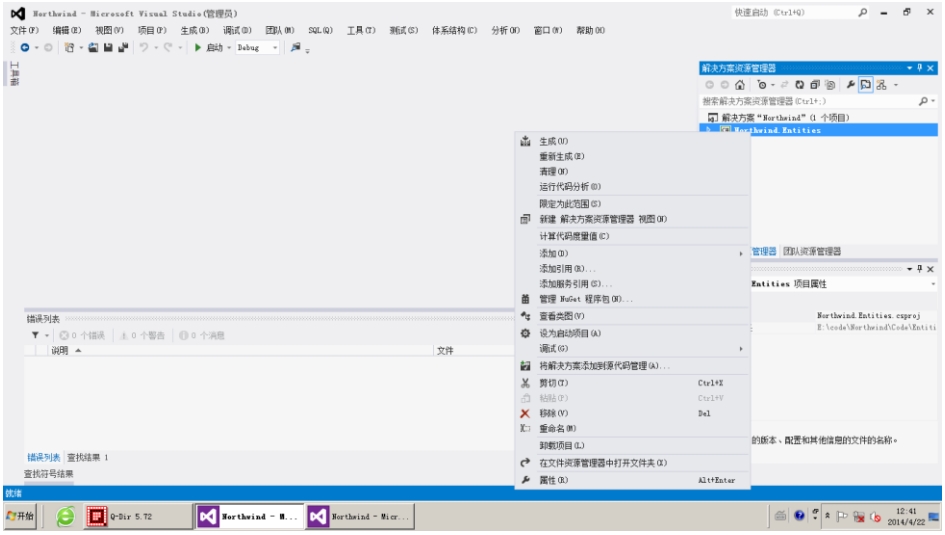
3.5.打开解决方案

进入 e:\code\Northwind\code，双击 Northwind.sln 打开解决方案。

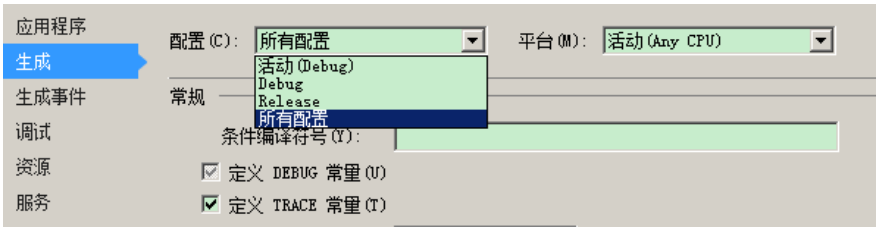


3.5.1. 设定项目编译生成路径

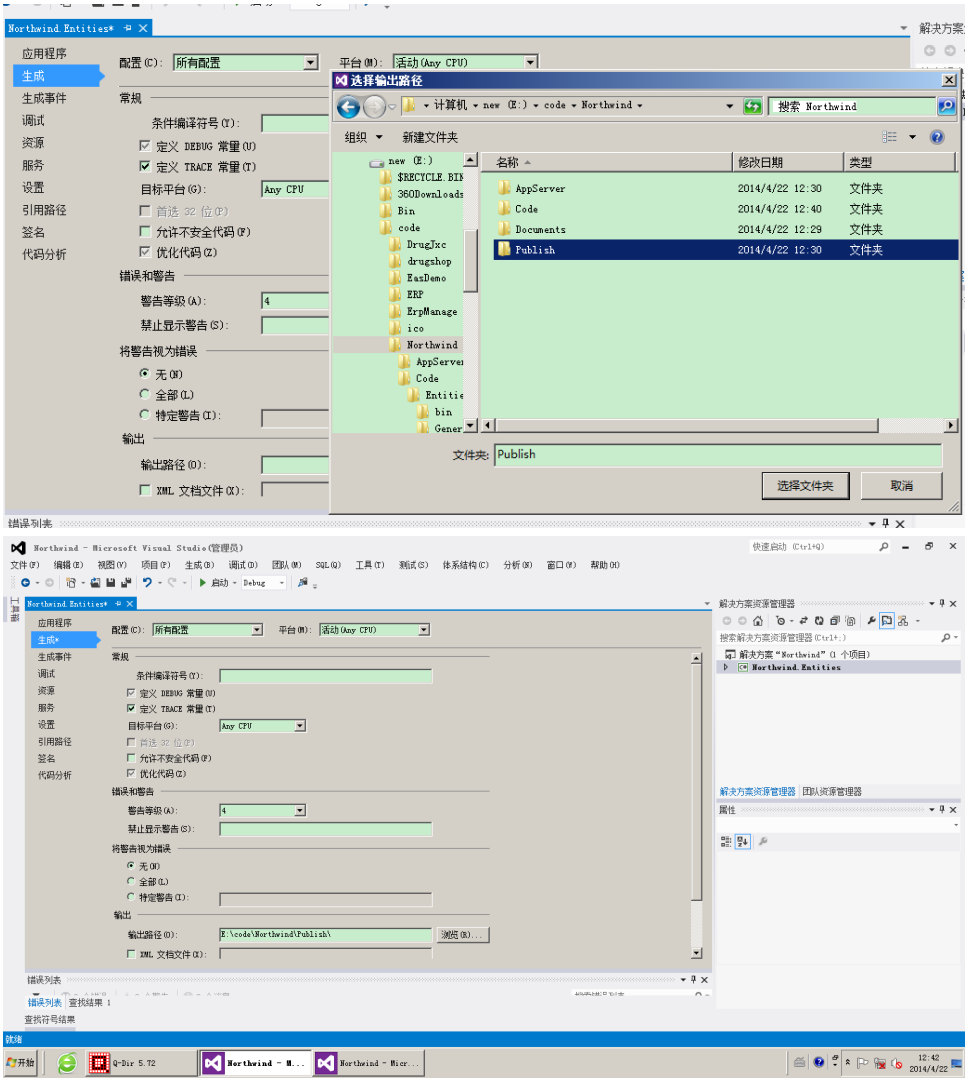
右键项目 Northwind.Entities，选择属性



设定生成/配置为所有配置



设定输出路径为：e:\code\Northwind\Publish

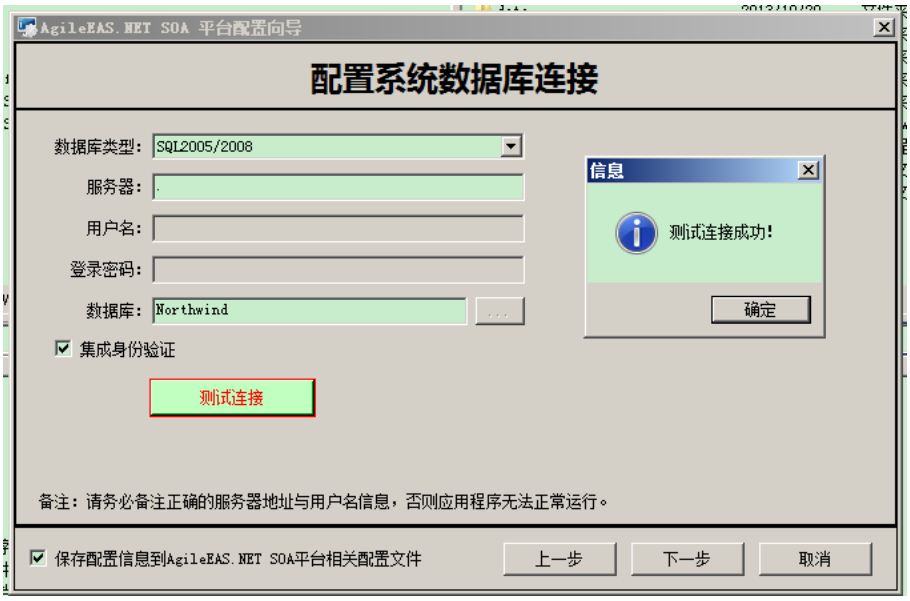


右键 Northwind.Entities，对项目进行重新生成。

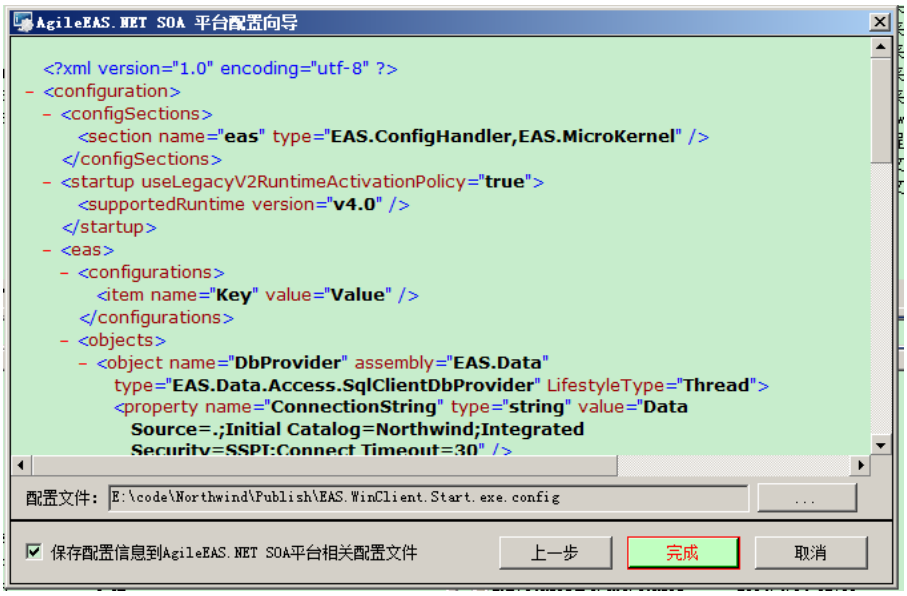
3.5.2. 维护数据字典

在进行进一步的编码工作之前，首先进行系统配置，并定义开发需要的输入字典。运行平台系统配置工具 EAS.Configure，设定连接数据库的方式等。

选择【生成客户端配置（数据库直连）】项，同时选中【保存配置信息到 AgileEAS.NET SOA 平台相关配置文件】。下一步



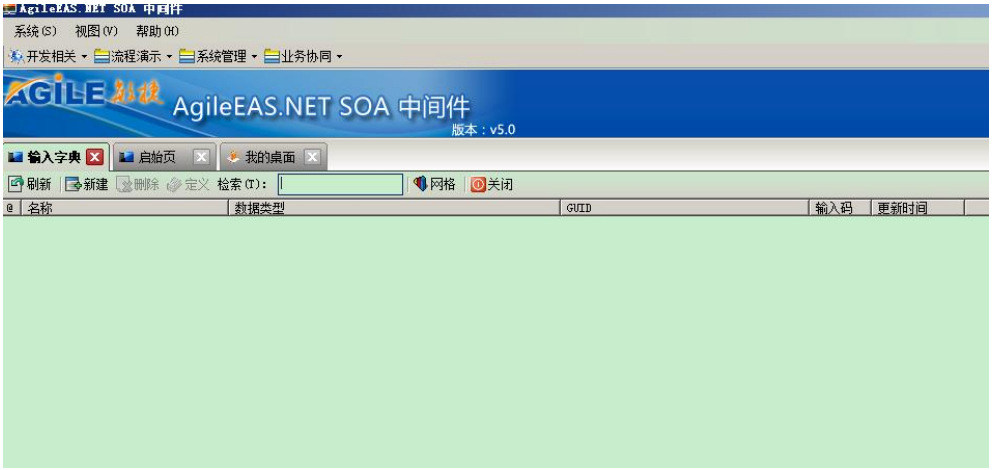
指定数据库服务器、身份验证信息、数据库名称等。完成后，点击下一步



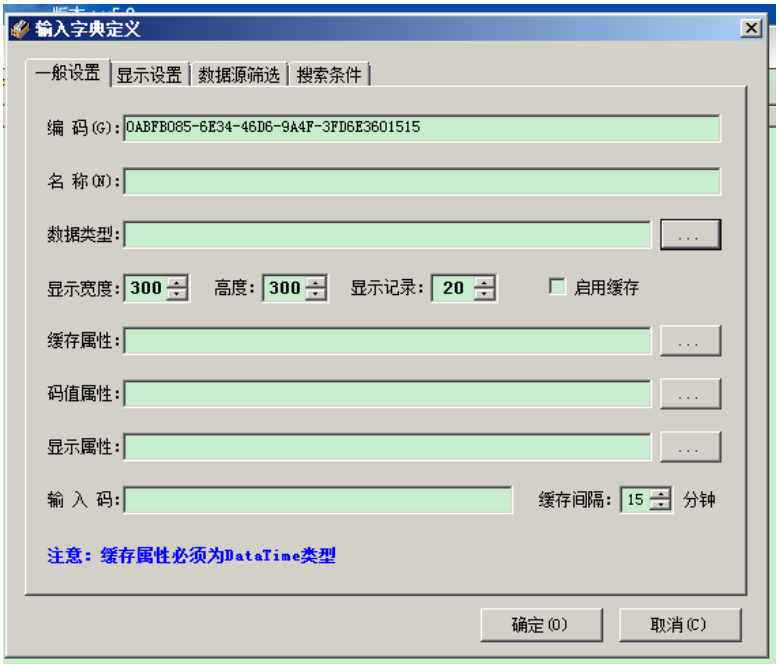
点击完成结束配置设置。

双击 EAS.WinClient.Start.exe 启动平台客户端，用户名：Administrator，密码：sa

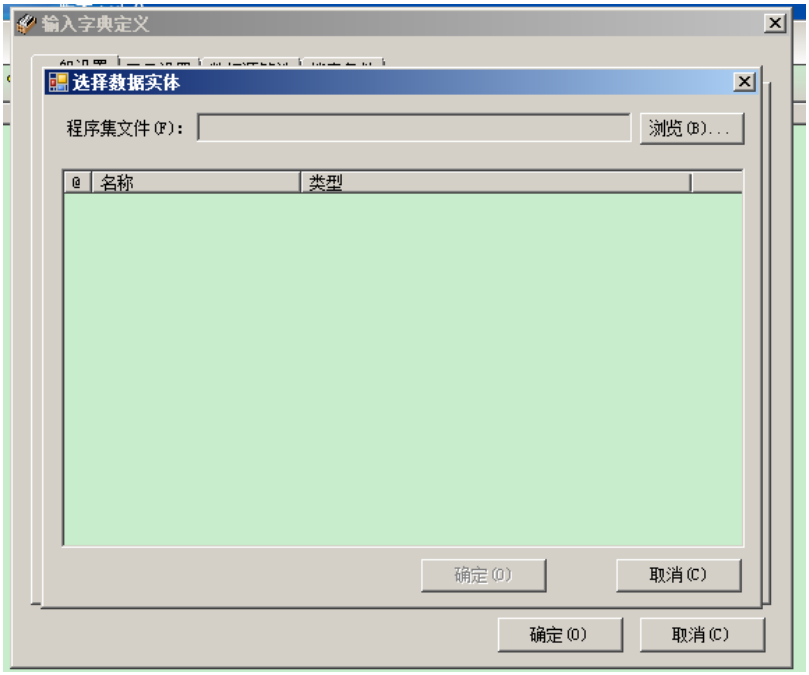
登录系统后选择菜单开发相关/输入字典



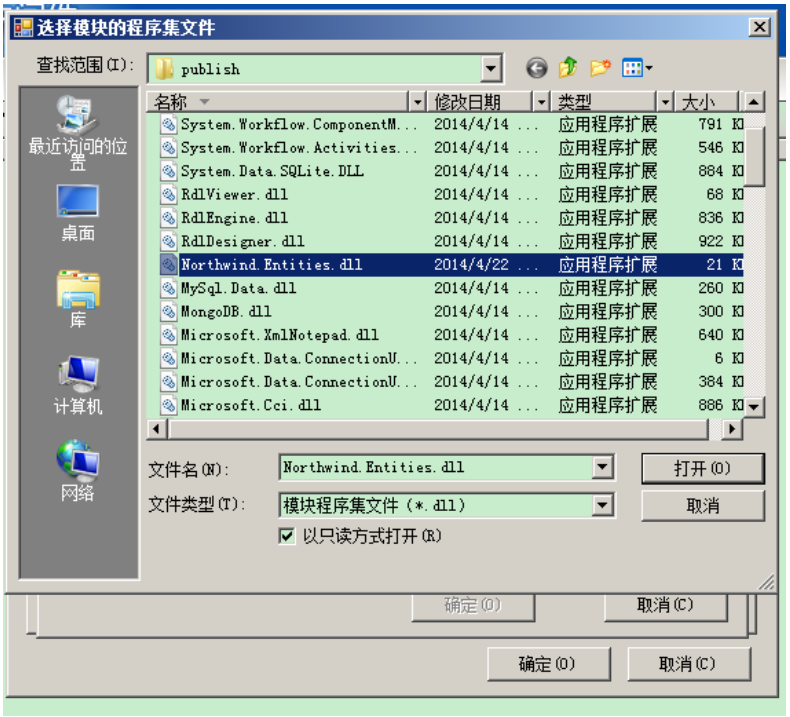
新建输入字典



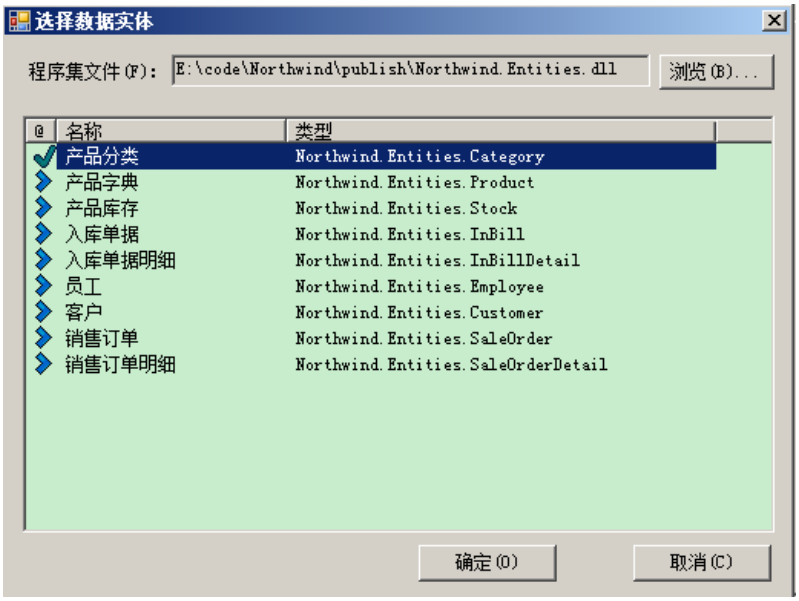
名称为产品分类，设定数据类型：



点击浏览，选择刚刚编译的 Northwind.Entities.dll

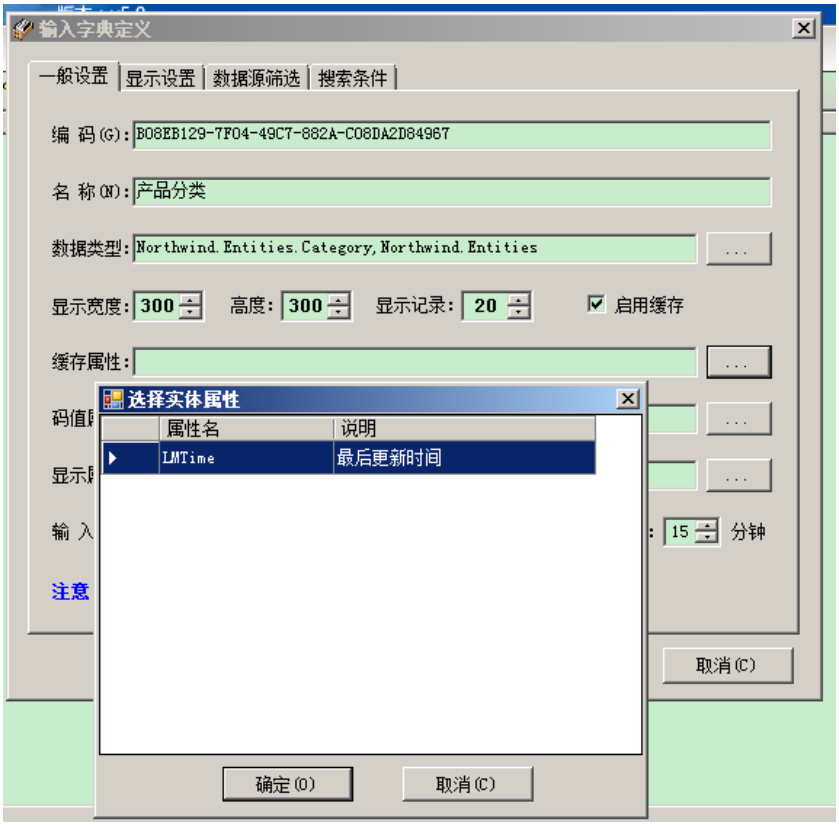


选择产品分类数据对象

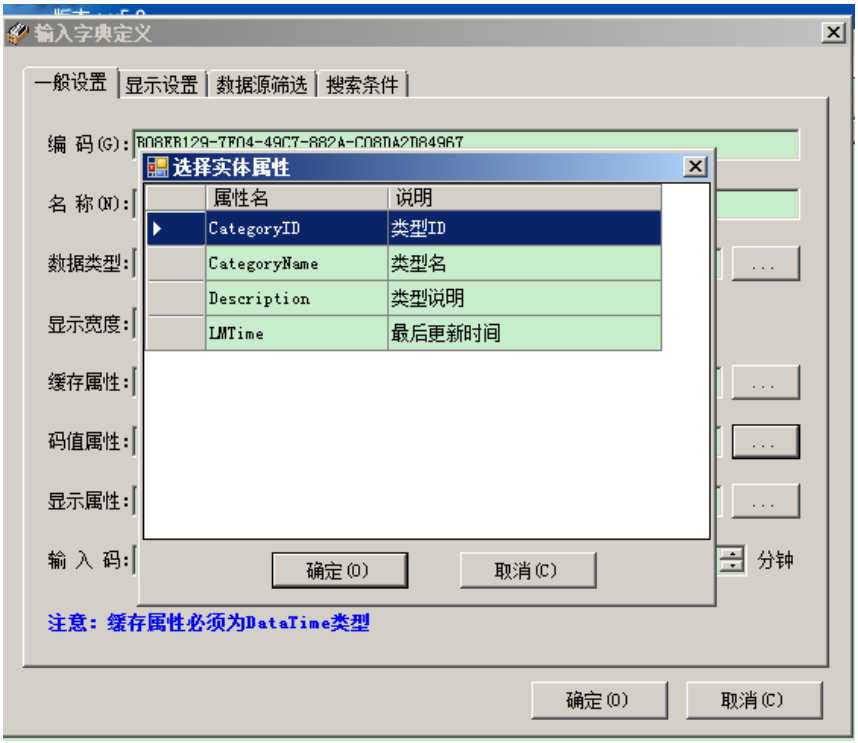


确定

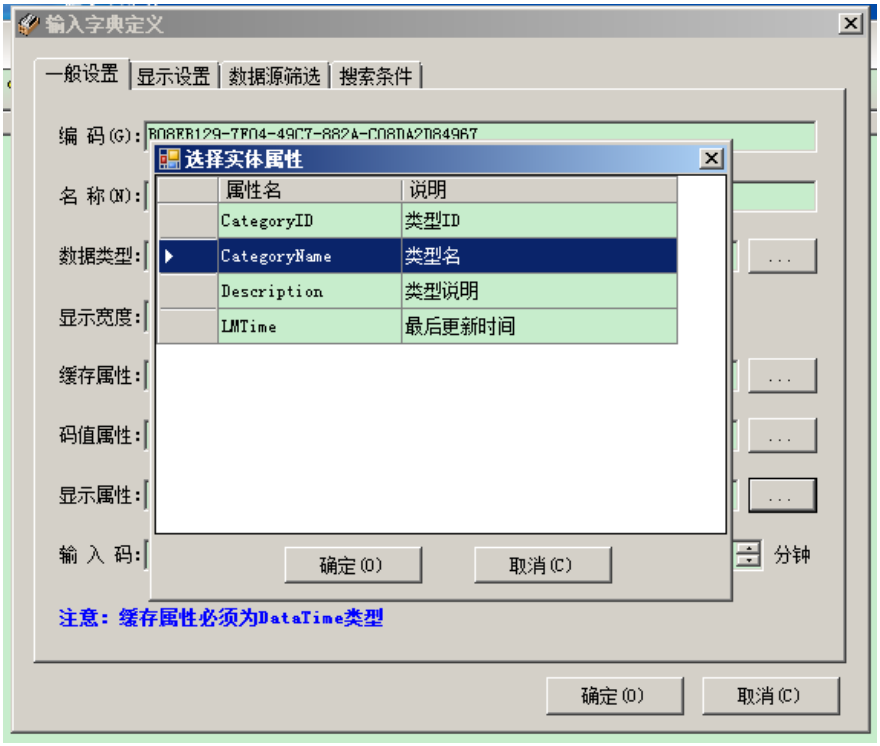
选中【启用缓存】，设定缓存属性为 LMTime

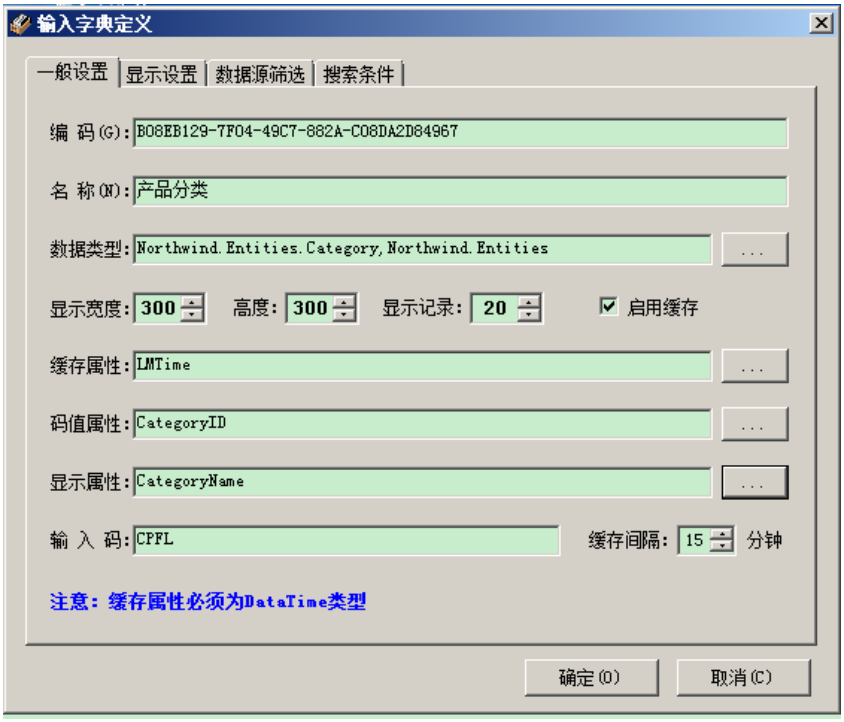


设定码值属性 CategoryID



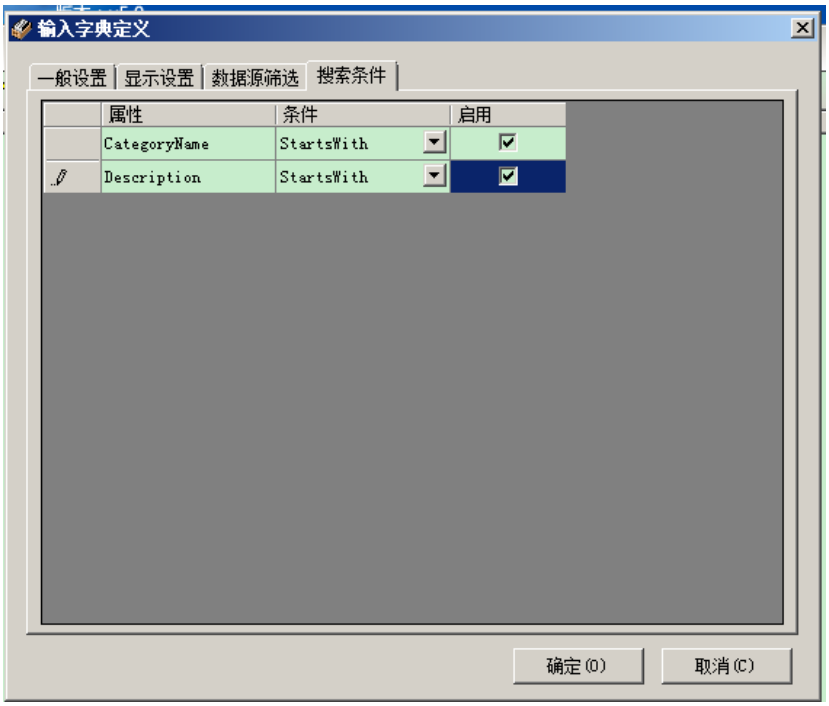
设定显示属性 CategoryName





显示设置：设定哪些项目在字典中显示

设定搜索条件

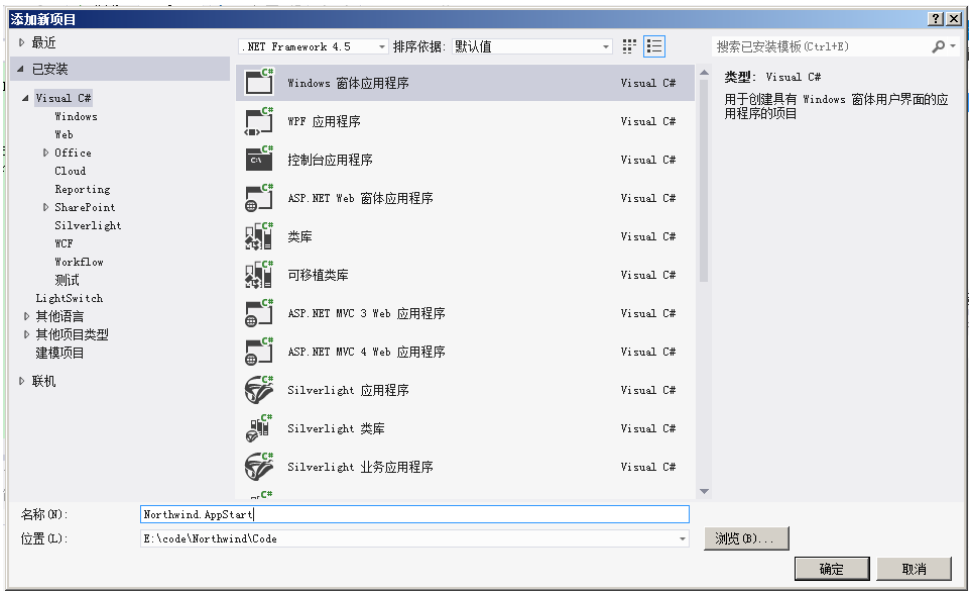


确定后完成，同样增加商品字典、客户字典、产品库存三个输入字典，具体可以参考案例中的输入字典进行设置。

商品字典对应 Northwind.Entities.Product 数据对象，客户字典对应 Northwind.Entities.Customer 数据对象，产品库存对应 Northwind.Entities.Stock 数据对象。

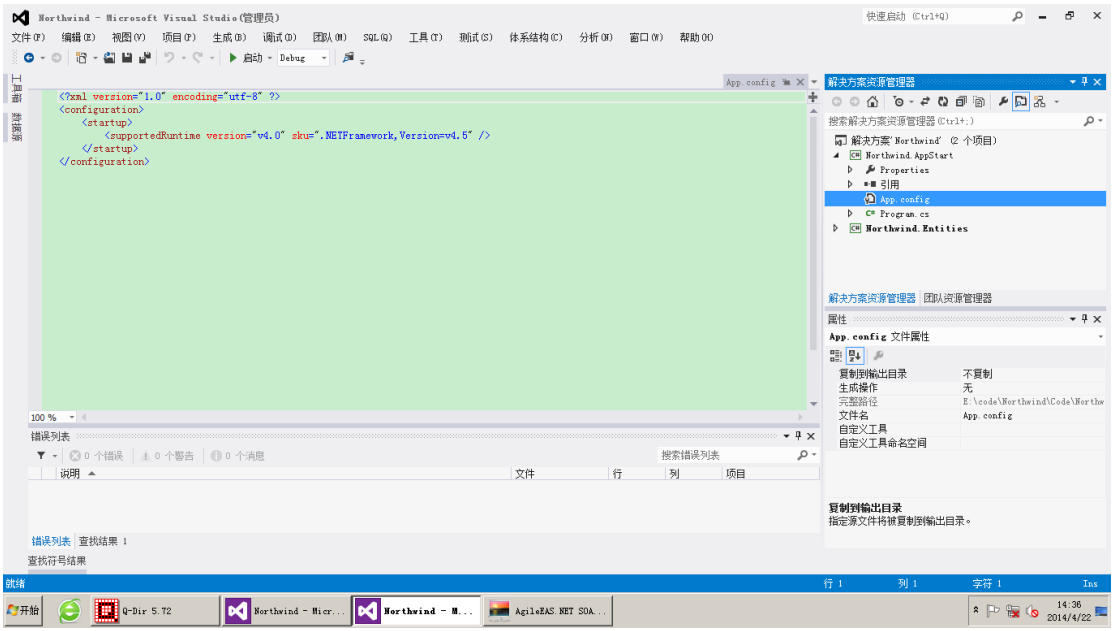
在进行正式的编码工作之前，还需要建立启动项：

在解决方案中添加项，类型为 winform 应用程序，名称为 Northwind.AppStart

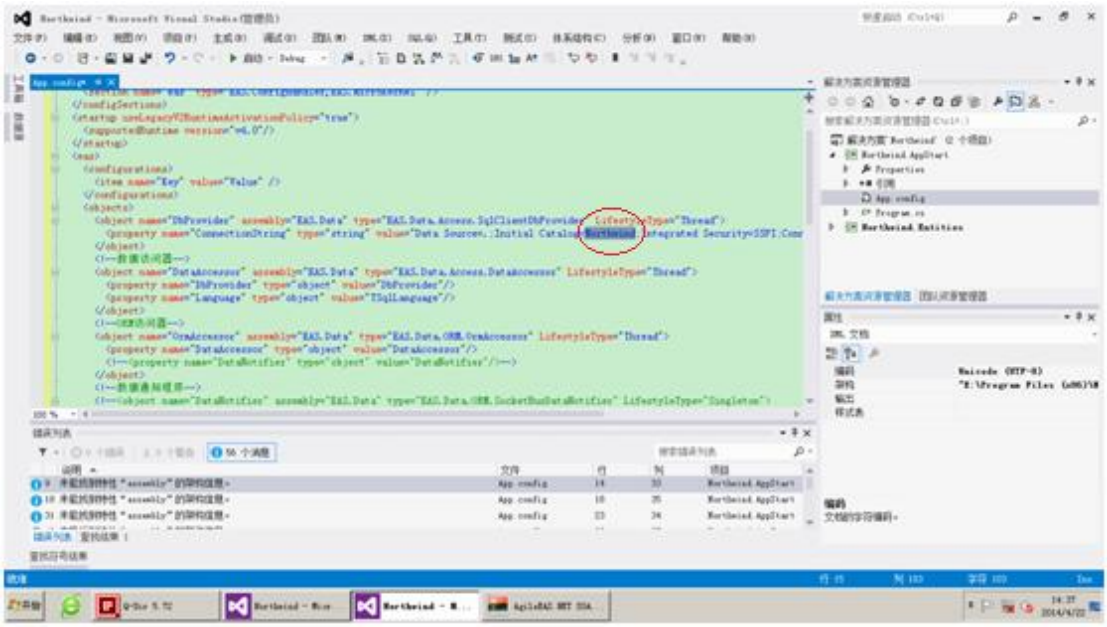


删除 form1.cs

修改 app.config

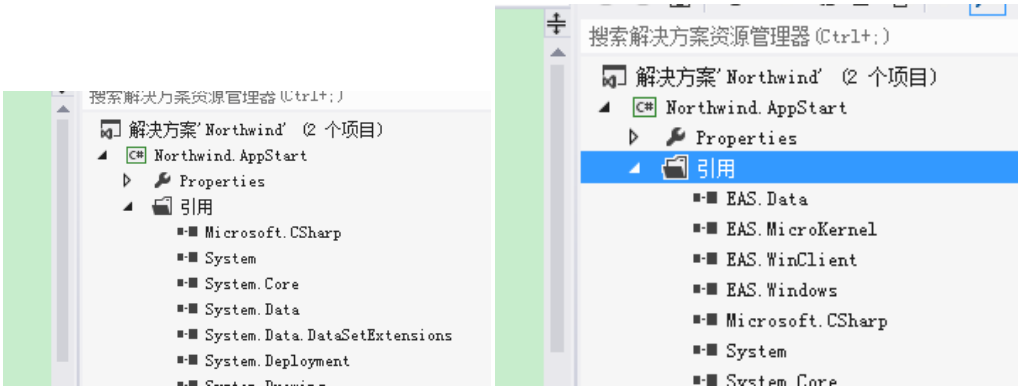


把案例相同文件中的内容复制过来，修改数据库名称为 Northwind



修改 Program.cs，从案例对应文件复制过来即可。

添加引用类库 EAS.Data、EAS.MicroKernel、EAS.WinClient、EAS.Windows，路径为 e:\code\Northwind\Publish。添加引用前后的不同。



把 Northwind.AppStart 设为启动项目。

注意：以下 2 步，所有新添加的项目都要进行设置。

Northwind.AppStart 鼠标右键，属性，设置目标框架为.net4（VS2012 默认为.net4.5）

设置输出路径为 e:\code\Northwind\Publish，编译。

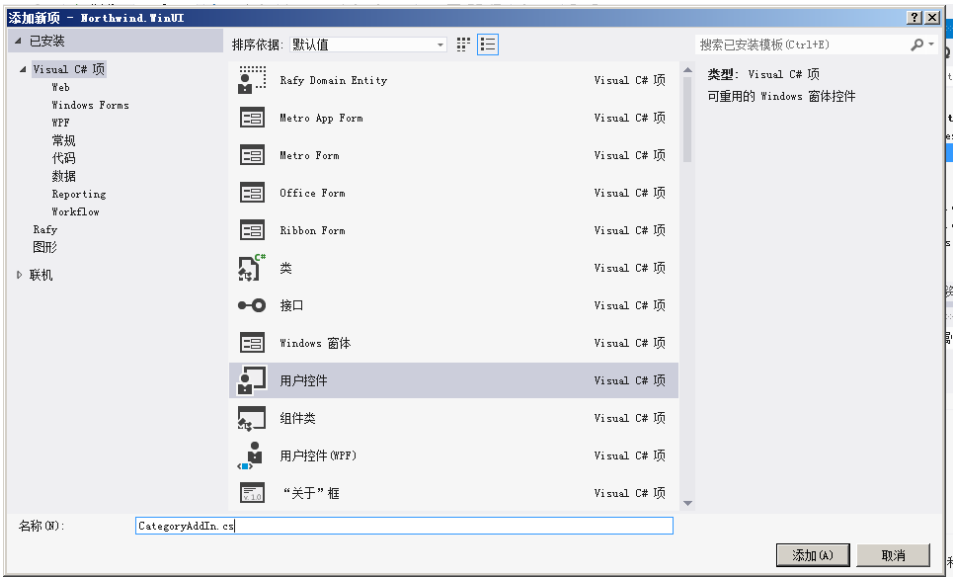
3.5.3. 两层（无业务层）模式开发、打印及按钮级权限控制

添加项目 Northwind.WinUI，类型为类库，删除 class1.cs 文件

添加引用类库 EAS.Data、EAS.MicroKernel、EAS.Data.Controls、EAS.Windows、EAS.Explorer、EAS.Explorer.Entities、EAS.GReport.Controls、EAS.Report.Controls、Northwind.Entities

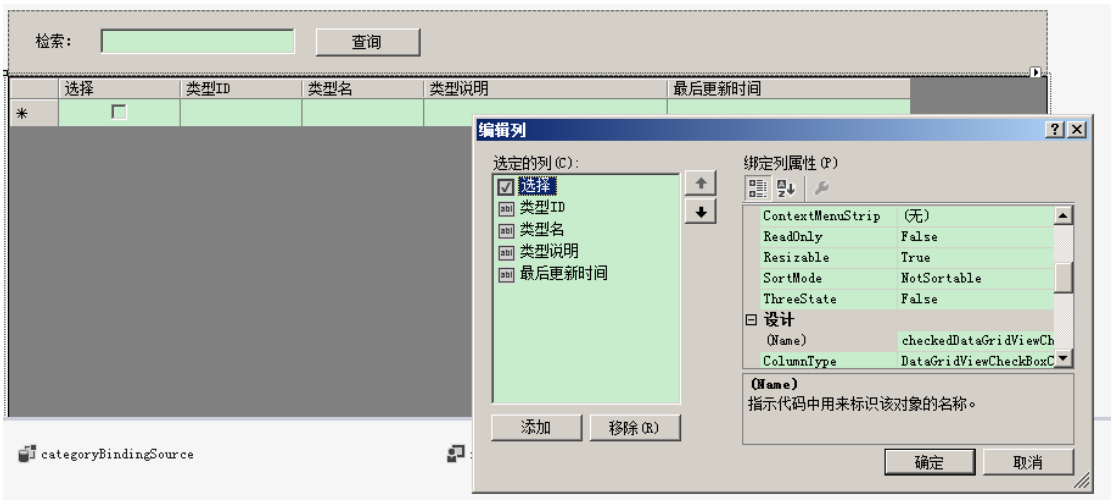
3.5.3.1. 用户控件类型窗体

在项目 Northwind.WinUI 添加用户控件 CategoryAddIn，产品分类
该类窗体可以作为平台插件，添加到平台功能树上，有 2 段代码为平台约定，必须添加。



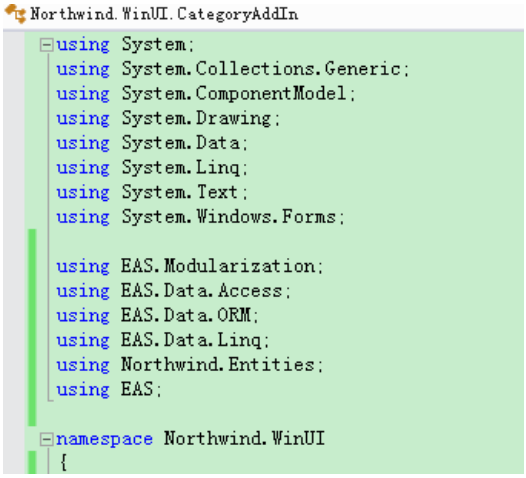
在窗体上添加控件（可以直接从案例对应窗体上复制，减少操作时间）

设定 dataGridView1 的 datasource 为 categoryBindingSource
dataGridView1 上鼠标右键，选择编辑列



在代码模式下添加引用

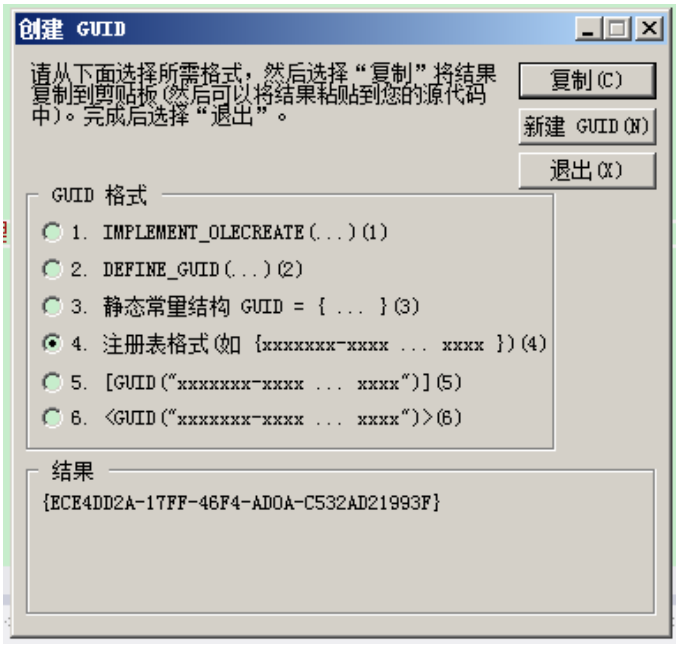
```
using EAS.Modularization;
using EAS.Data.Access;
using EAS.Data.ORM;
using EAS.Data.Linq;
using Northwind.Entities;
using EAS;
```



增加以下一行代码，此为平台约定的格式，必须有。

```
[Module("0A1745F3-7F95-4898-8E51-539818DAB95C", "产品分类", "")]
```

选中 GUID 字符串，然后选择工具、生成 GUID，生成新的 GUID，替换原 GUID：



替换后的结果如下：

```
using Northwind.Entities;
using EAS;

namespace Northwind.WinUI
{

    //以下内容为平台约定代码
    [Module("0A1745F3-7F95-4898-8E51-539818DAB95C", "产品分类", "")]
    //以下内容为平台约定代码
    public partial class CategoryAddIn : UserControl
    {
```

添加以下代码到指定位置

```
//以下代码为平台约定内容
        [ModuleStart]
        public void StartEx()
        {

        }
        //以上代码为平台约定内容
```

```
namespace Northwind.WinUI
{

    //以下内容为平台约定代码
    [Module("0A1745F3-7F95-4898-8E51-539818DAB95C", "产品分类", "")]
    //以下内容为平台约定代码
    public partial class CategoryAddIn : UserControl
    {
        //以下代码为平台约定内容
        [ModuleStart]
        public void StartEx()
        {
            this.LoadData();
        }
        //以下内容为平台约定代码
        public CategoryAddIn()
        {
```

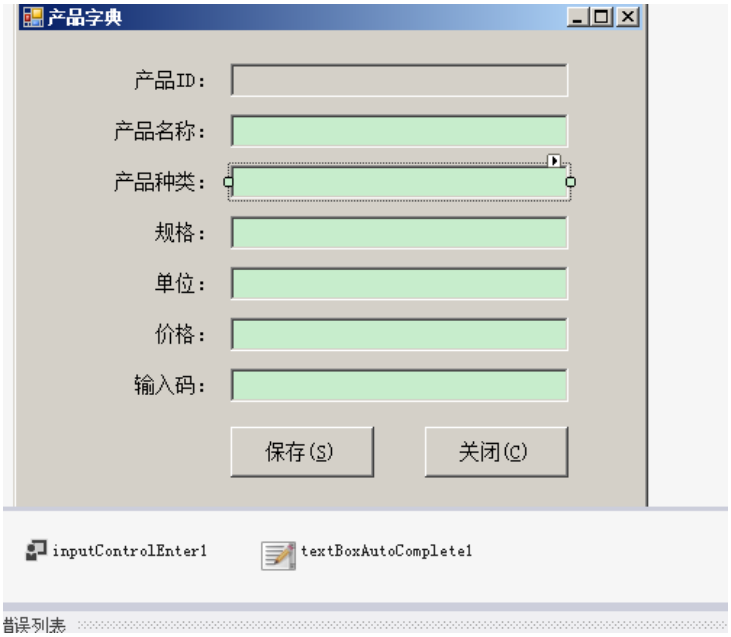
在 Northwind.Entities 数据对象 Category 中添加以下代码：

```
public int GetMaxID()
        {
                ParameterCollection pc = new ParameterCollection();
                pc.Add("ITEMKEY", this.DbTableName);

                return (int)this.DataAccessor.Query("exec GetIdentityValue @itemkey=?", pc);
        }

        public override string ToString()
        {
                return this.CategoryName;
        }
```

3.5.3.2. 普通 windows 窗体

此类型窗体的编程与普通程序相同，没有特殊要求。

在 Northwind.WinUI 增加 ProductEditor，产品录入/修改界面



界面控件和代码从案例中复制过来即可。

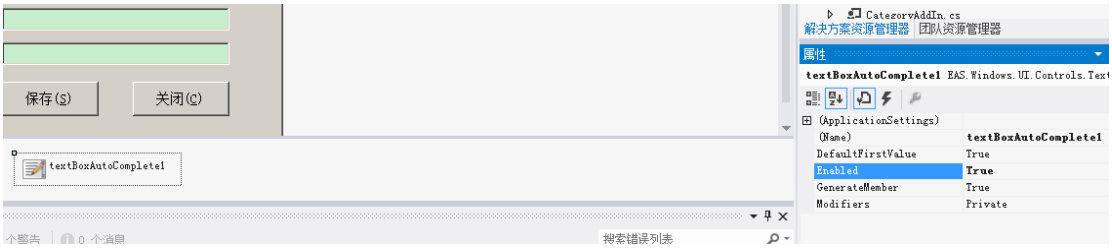
3.5.3.3. 输入字典设置

平台提供了输入字典通用组件，可以很方面的实现软件数据录入界面中常见的选择录入方式（例如在入库单上选择仓库、部门、人员、产品等），输入指点我们在 4.2.2 中已经维护，这里直接调用即可。

以下重点说明输入字典的设置：

选中产品种类后的文本框，把前面通过开发相关设置的产品分类输入字典的 GUID（上图红框中的内容）复制到文本框的属性【textBoxAutoComplete1 上的 MetadataID】，如下图所示。

选中 textBoxAutoComplete1 组件，设置其 Enabled 属性为 true，这样，就可以通过帮助字典的方式输入产品种类。



3.5.3.4. 打印设置

平台提供的打印功能使用起来非常的便捷，只需要 2 步就能够实现单据或报表的打印，而且打印可以在使用过程中进行修改，不需要重新修改代码及重新编译程序。

1、 通过平台的报表管理工具设计打印格式；（详细操作参见 4.2.5.报表及数据导出）

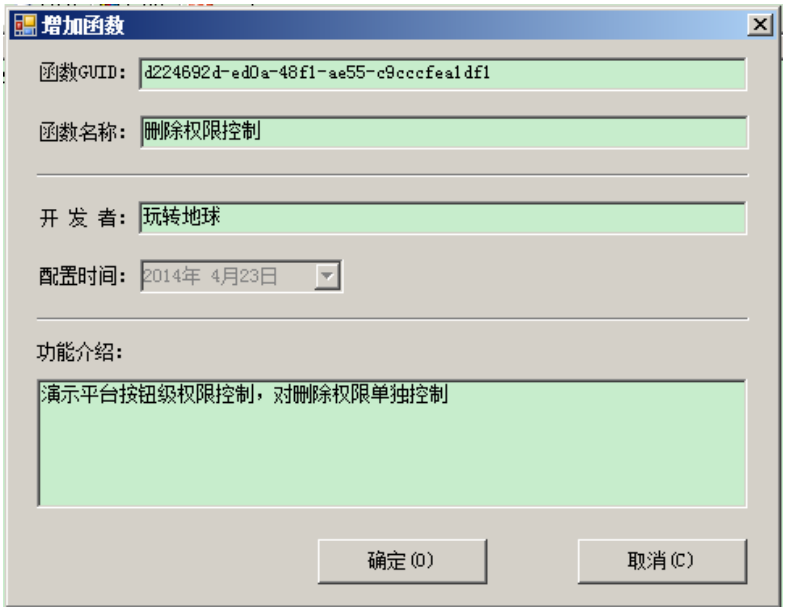
2、 在需要打印的地方通过报表名称直接调用（案例中提供了打印的样例代码）。
这里我们通过报表管理定义了一张打印格式【产品列表】，然后，把报表名称填入打印按钮对应的代码中即完成了打印的调用，如下：
this.Report.Name = "产品列表"; //在此处修改报表名称

```
private void btnPrint_Click(object sender, EventArgs e)
{
    this.Report.Name = "产品列表"; //在此处修改报表名称
    this.Report.Refresh();
    if (!this.Report.Exists)
    {
        MessageBox.Show("没有找到'" + this.Report.Name +"' 报表的定义。请联系
        return;
    }

    this.PrintForm.Report = this.Report;
    this.PrintForm.DataObject = this.vList.ToList();
    this.PrintForm.PrintPreview();
}
```

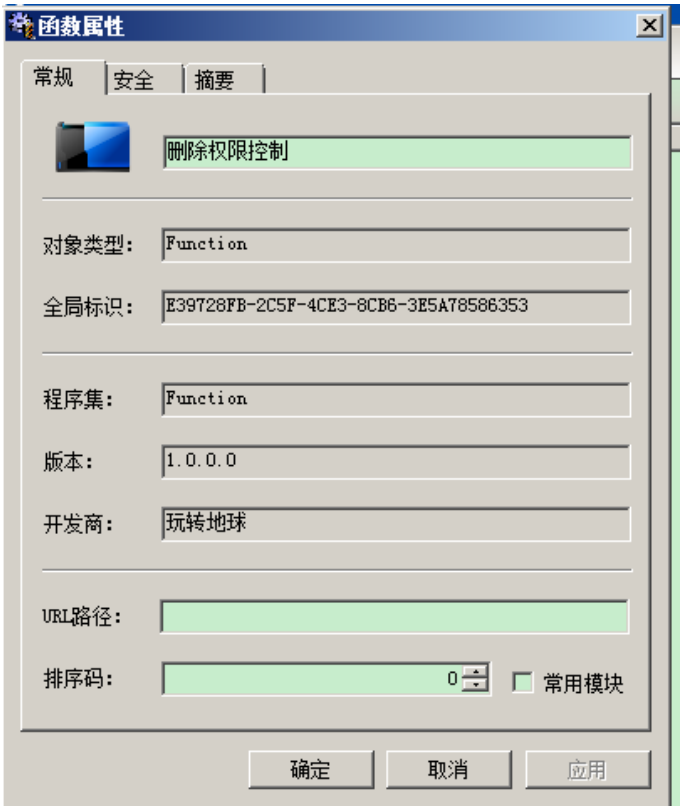
3.5.3.5. 按钮级权限控制

处理功能模块级的控制外，平台还提供了更加严格的权限控制——按钮级权限控制。
平台以函数管理的形式提供了按钮级的权限控制，具体操作如下：
1、通过平台系统管理/函数管理增加需要进行管理的按钮权限控制函数。

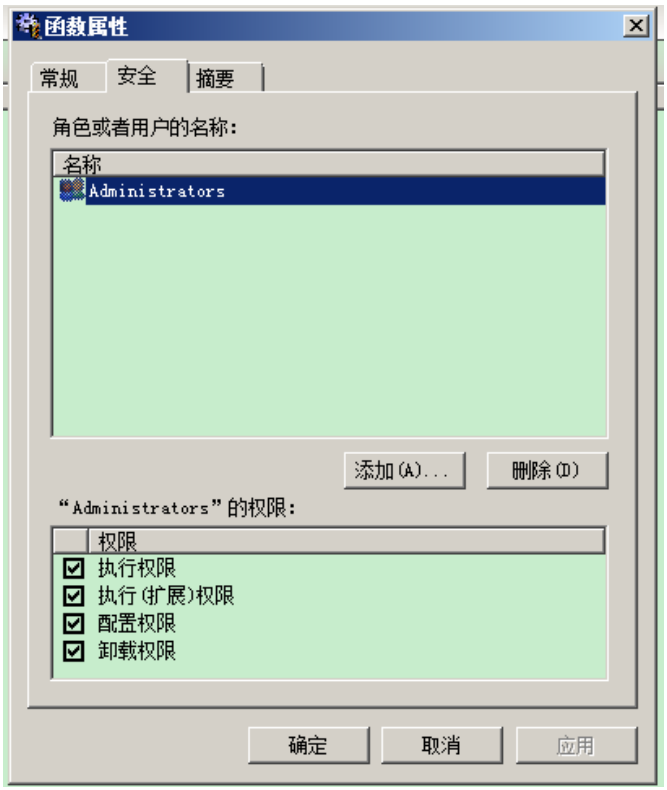
2、添加函数【删除权限控制】
对产品管理列表的删除按钮权限进行控制，如无权限则提示无权限，不允许删除。
函数的调用时通过 GUID 进行的，在需要控制的按钮相应的代码中把 GUID 填入。



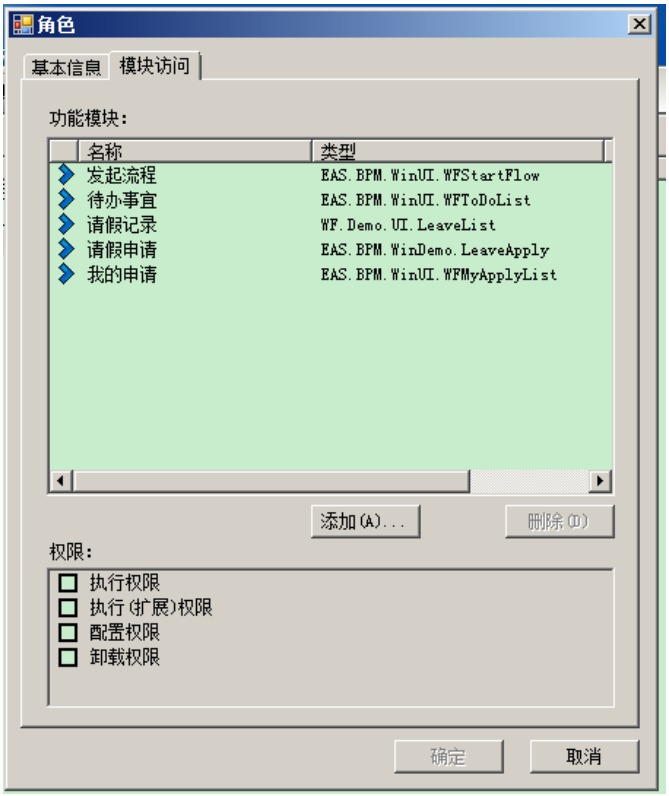
完成后，确定。然后选中函数并点击属性：

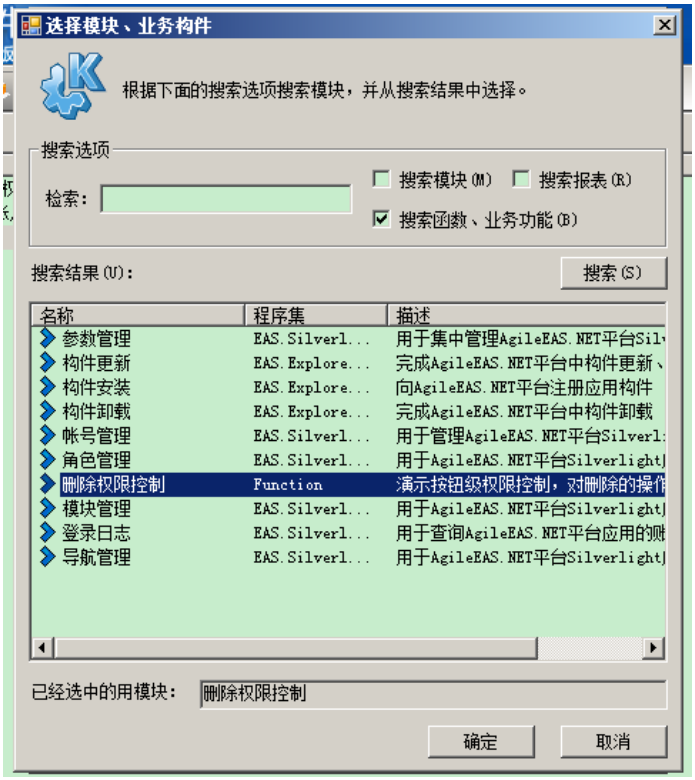


选择【安全】页签进行函数的授权。



也可以通过角色管理进行授权。如下图所示：





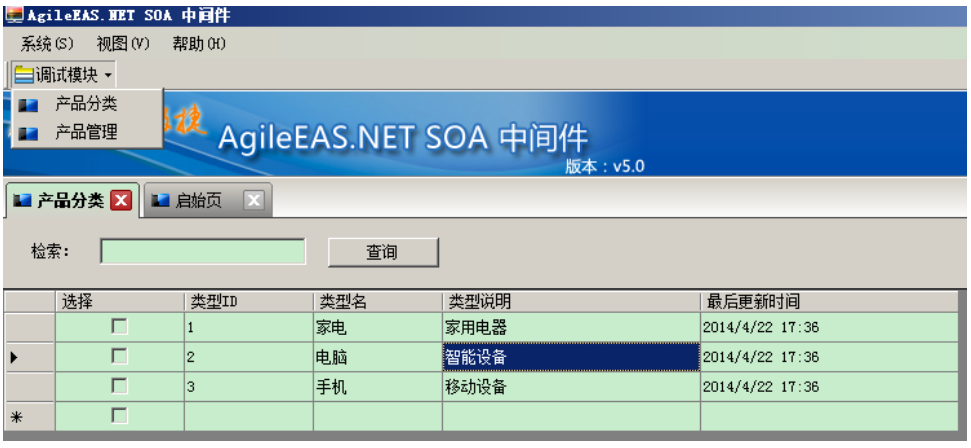
设置完成后，在需要控制的按钮代码中增加如下 2 行代码即完成按钮级权限控制的设计开发。

```
private void btnDelete_Click(object sender, EventArgs e)
{
    //验证按钮级别权限。
    if (!EAS.Application.Instance.Demand(new Guid("E39728FB-2C5F-4CE3-8CB6-3E5A78586353")))
    {
        MessageBox.Show("您未获得删除产品的权限，请与系统管理员联系！");
        return;
    }
```

3.5.3.6. 程序测试

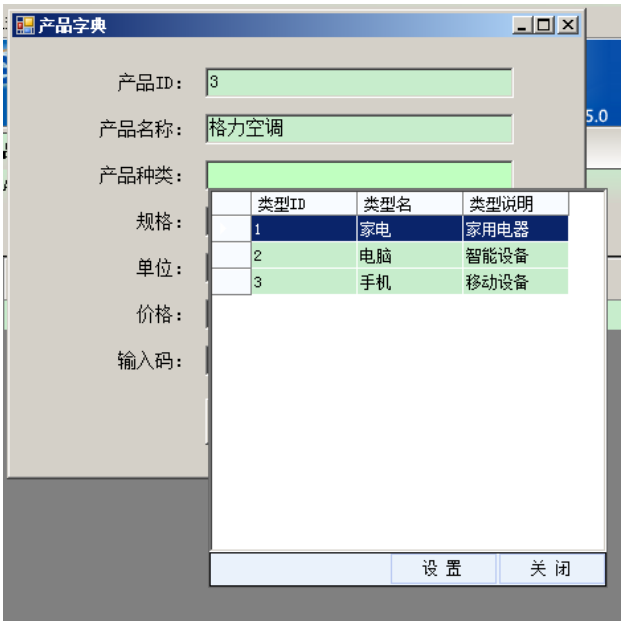
启动程序，登录系统，用户名：Administrator，密码：sa

录入产品分类

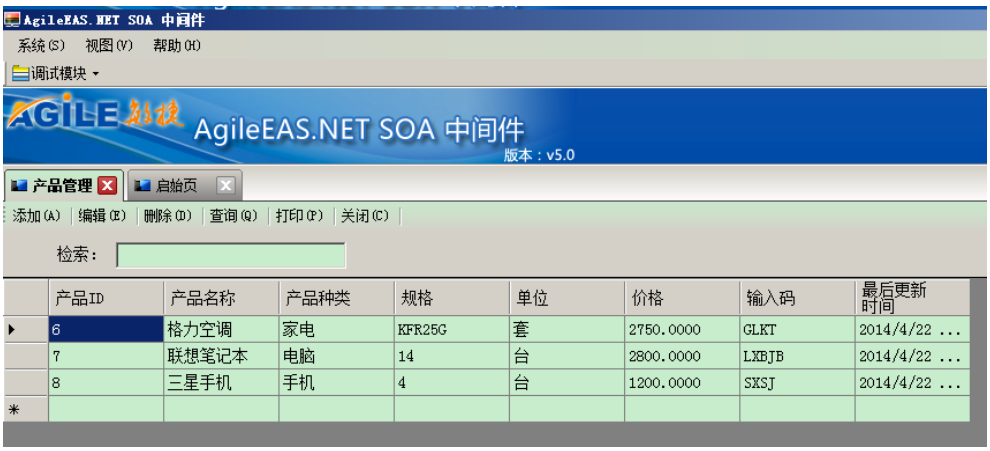


录入产品：

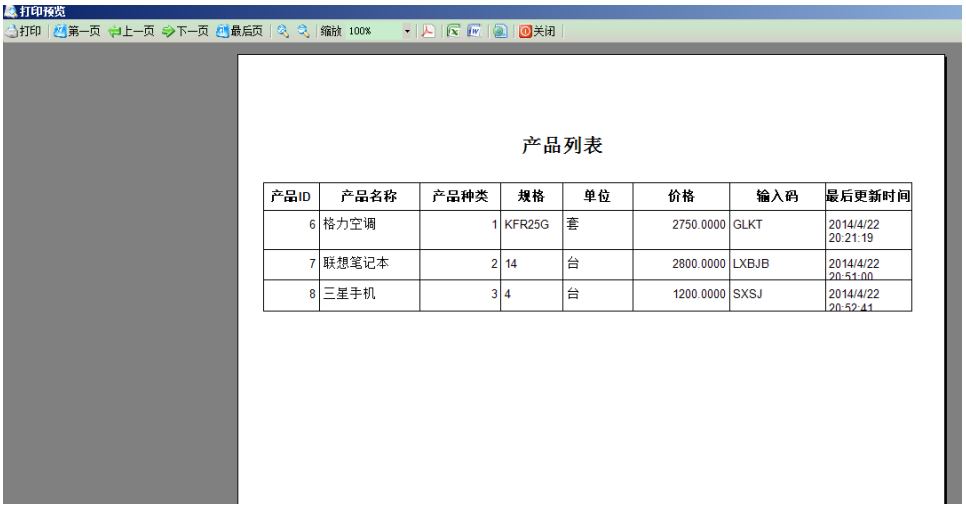
在产品种类文本框单击空格键，弹出产品分类输入字典，选择对应的分类即可。



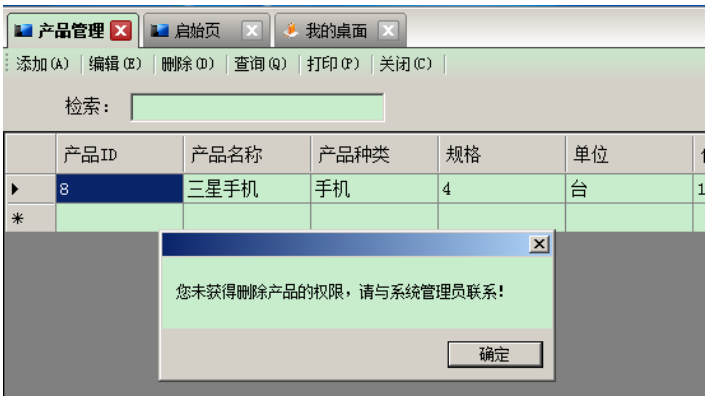
录入测试数据



打印测试：点击【打印】按钮，系统弹出打印预览窗口



删除按钮权限控制测试：在未经授权的情况下结果如下图所示：（此功能在调试模式下无效，需要进行模块安装、授权后才能够正常）



3.5.4. 多层（有业务层）模式开发

参照案例代码，添加两个项目
Northwind.BLL.Contracts 作为服务端与客户端的接口
Northwind.BLL.Host　　服务端业务处理逻辑
设定目标框架为.net 4.0，目标平台为 Any CPU，输出路径为 Northwind 下的 Publish

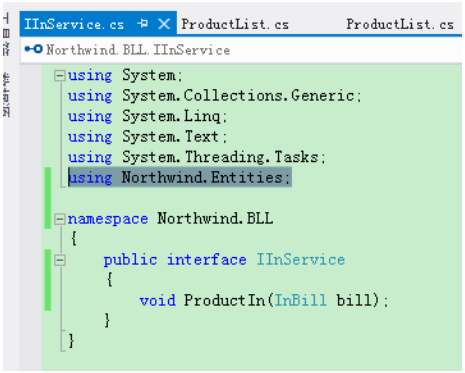
3.5.4.1.　接口层开发

Northwind.BLL.Contracts：
添加类库引用：EAS.Data、EAS.MicroKernel、Northwind.Entities

新建接口 IInService，处理采购入库业务。
添加代码

```
using Northwind.Entities;
```

其它代码如图所示，详见案例对应代码：

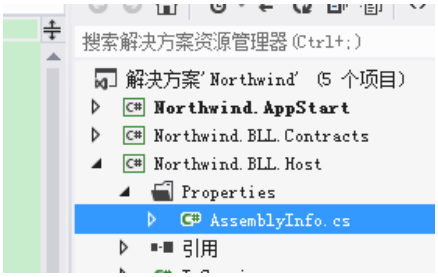


编译生成 Northwind.BLL.Contracts.dll

3.5.4.2. 业务逻辑层开发

对应的，在 Northwind.BLL.Host 中添加类库引用 EAS.Data、EAS.MicroKernel、Northwind.Entities、Northwind.BLL.Contracts

修改属性文件 Properties 目录下的 AssemblyInfo



修改前：

```
using System.Reflection;
using System.Runtime.CompilerServices;
using System.Runtime.InteropServices;

// 有关程序集的常规信息通过以下
// 特性集控制。更改这些特性值可修改
// 与程序集关联的信息。
[assembly: AssemblyTitle("Northwind.BLL.Host")]
[assembly: AssemblyDescription("")]
[assembly: AssemblyConfiguration("")]
[assembly: AssemblyCompany("Microsoft")]
[assembly: AssemblyProduct("Northwind.BLL.Host")]
[assembly: AssemblyCopyright("Copyright © Microsoft 2014")]
[assembly: AssemblyTrademark("")]
[assembly: AssemblyCulture("")]

// 将 ComVisible 设置为 false 使此程序集中的类型
// 对 COM 组件不可见。如果需要从 COM 访问此程序集中的类型，
// 则将该类型上的 ComVisible 特性设置为 true。
[assembly: ComVisible(false)]

// 如果此项目向 COM 公开，则下列 GUID 用于类型库的 ID
[assembly: Guid("1fa32d4f-d168-4870-bcbd-1744ade45699")]

// 程序集的版本信息由下面四个值组成:
//
```

修改后增加了红框中的 2 行内容：

```
using System.Reflection;
using System.Runtime.CompilerServices;
using System.Runtime.InteropServices;
//下面一行为EAS.NET平台约定
using EAS.Services;
// 有关程序集的常规信息通过以下
// 特性集控制。更改这些特性值可修改
// 与程序集关联的信息。
[assembly: AssemblyTitle("Northwind.BLL.Host")]
[assembly: AssemblyDescription("")]
[assembly: AssemblyConfiguration("")]
[assembly: AssemblyCompany("Microsoft")]
[assembly: AssemblyProduct("Northwind.BLL.Host")]
[assembly: AssemblyCopyright("Copyright © Microsoft 2014")]
[assembly: AssemblyTrademark("")]
[assembly: AssemblyCulture("")]
//下面一行为EAS.NET平台约定
[assembly: ServiceAssembly()]
// 将 ComVisible 设置为 false 使此程序集中的类型
// 对 COM 组件不可见。如果需要从 COM 访问此程序集中的类型，
// 则将该类型上的 ComVisible 特性设置为 true。
[assembly: ComVisible(false)]

// 如果此项目向 COM 公开，则下列 GUID 用于类型库的 ID
```

添加 InService 类，具体实现 Northwind.BLL.Contracts 中定义的接口

```
Northwind.BLL.InService
using System;
using System.Collections.Generic;
using System.Linq;
using System.Text;
using System.Threading.Tasks;

namespace Northwind.BLL
{
    public class InService
    {

    }
}
```

添加类库引用代码

using EAS.Data.Linq;

using EAS.Data.ORM;

using Northwind.Entities;

using EAS.Services;

修改代码如下：

```
Northwind.BLL.InService
using System;
using System.Collections.Generic;
using System.Linq;
using System.Text;
using EAS.Data.Linq;
using EAS.Data.ORM;
using Northwind.Entities;
using EAS.Services;

namespace Northwind.BLL
{
    [ServiceObject("采购入库服务")]
    [ServiceBind(typeof(IInService))]
    public class InService : ServiceObject, IInService
    {
        #region IInService 成员

        public void ProductIn(Entities.InBill bill)...
```

到此就完成了接口层和业务逻辑层的开发。

3.5.4.3. UI 层开发

首先在 Northwind.WinUI 中添加对 Northwind.BLL.Contracts 的应用

添加用户组件 ProductInAddIn，用于录入采购入库单

界面参照案例设计、布局（可以直接复制过来，以节省时间）

参照案例添加对应的代码

参照案例，在 Northwind.Entities 的数据对象 InBill、InBillDetail 添加 GetMaxID()，完成重新编译。

参考案例添加 LoginInfo 类。

为文本框 tbName 的属性【textBoxAutoComplete1 上的 MedaDataID】填入 4.2.3.3 中设置的输入字典【商品字典】的编码。

为文本框控件 tbNumber 绑定 tbNumber_KeyDown 事件；

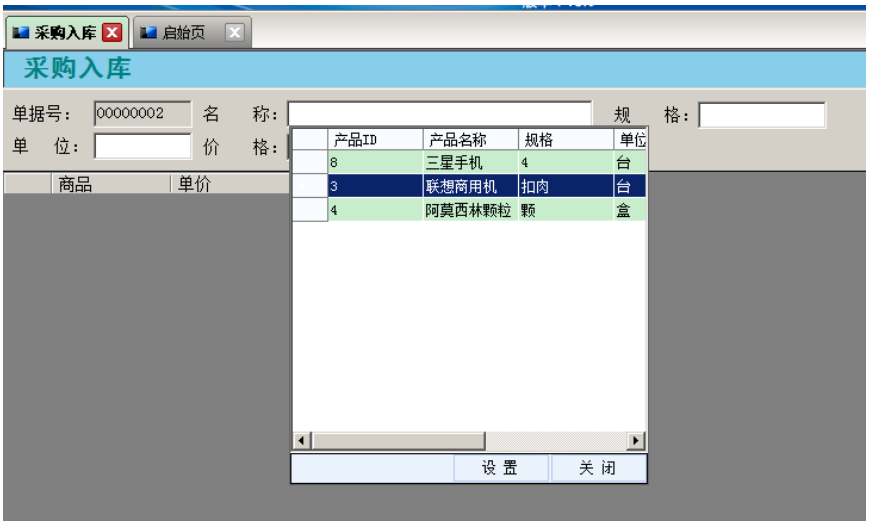
为按钮控件 btnOK 绑定 btnOK_Click 事件；

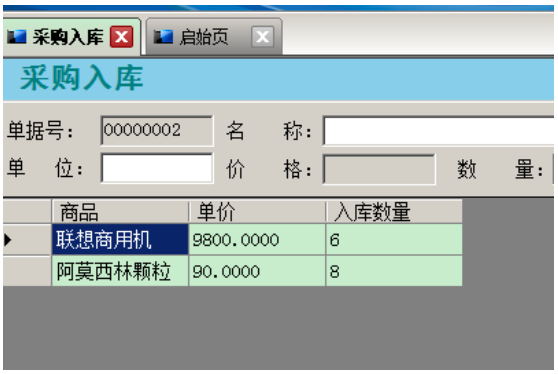
为数据字典控件 textBoxAutoComplete1 绑定 textBoxAutoComplete1_InputComplete 事件。

完成后重新编译 UI 层。

运行测试：

采购入库菜单已经出现，点击开始录入数据





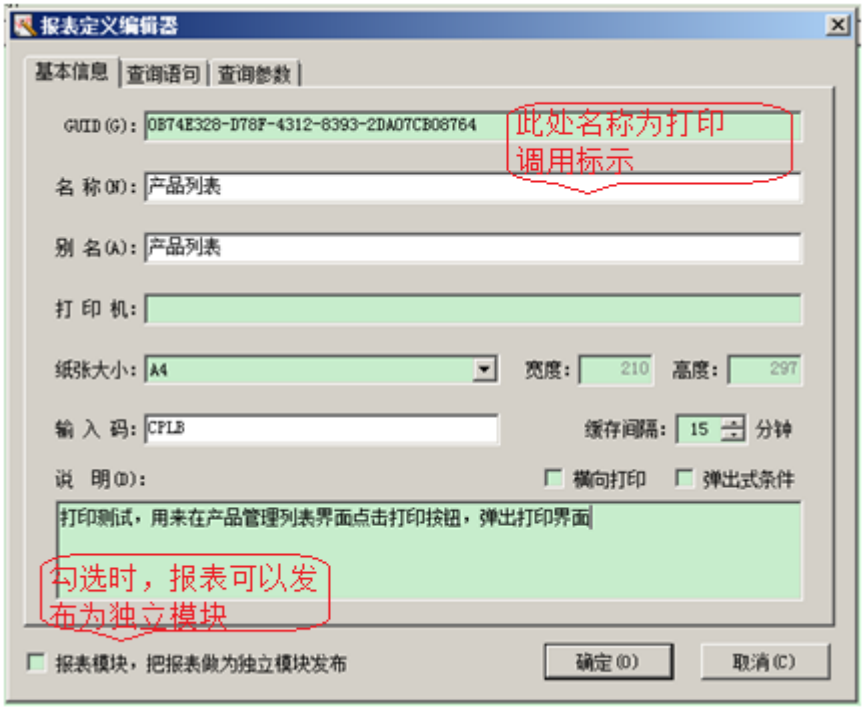
到此，所有的开发和编码工作基本完成。
其它功能模块的开发可以参考上述步骤及案例全部做一遍，以加强对平台的掌握。

3.5.5. 报表及数据导出设计开发

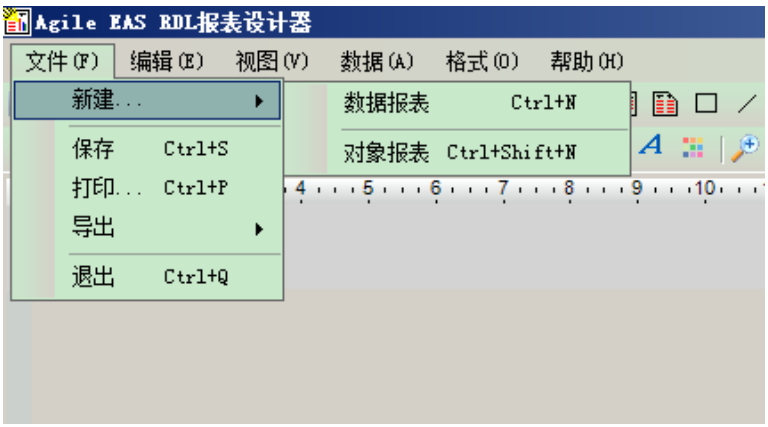
平台提供了强大的打印设计、报表设计及自定义报表。
为了前面我们测试的打印功能，我们在此设计【产品列表】报表，以 Administrator 账户进入系统，选择开发相关/报表管理（注意：此处是报表管理而不是 GR 报表管理）：

点击添加，新建报表

录入报表名称：产品列表（平台要求报表名称必须唯一，作为打印调用的标识），查询语句、查询参数在此不做处理，确定

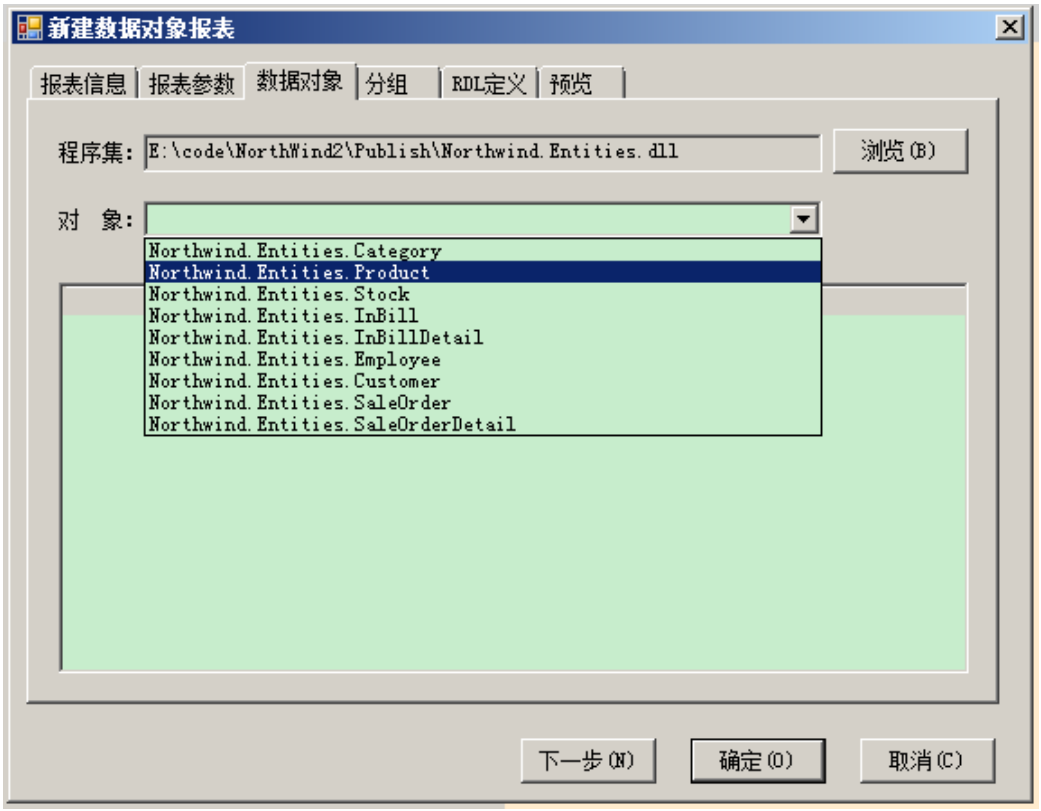


选中报表然后点击设计按钮，进行报表内容和格式的设计，在文件/新建/选择对象报表

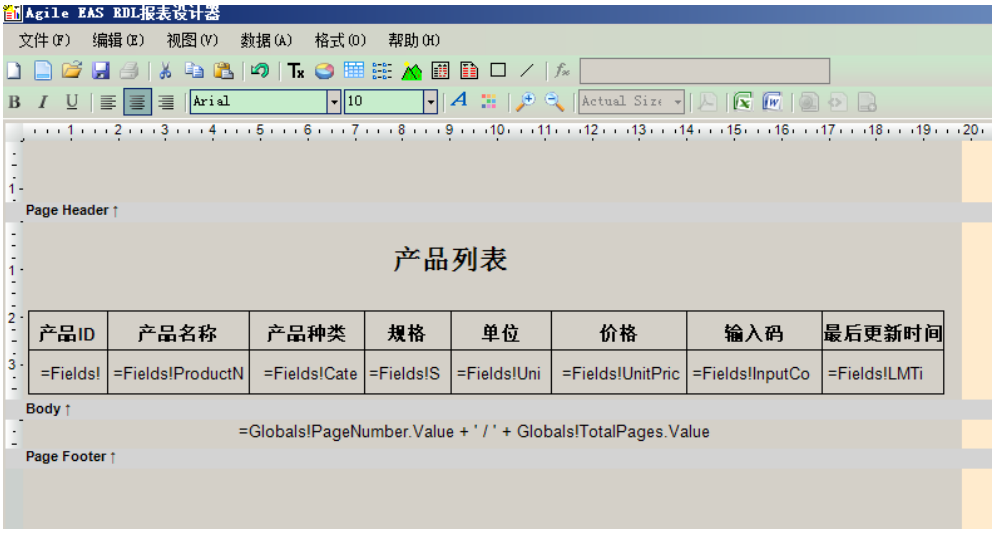


在报表信息页签录入报表名称：产品列表及其他信息

在数据对象页签程序集选择 e:\code\Northwind\Publish\Northwind.Entities.dll 对象选择 Product，如下图所示

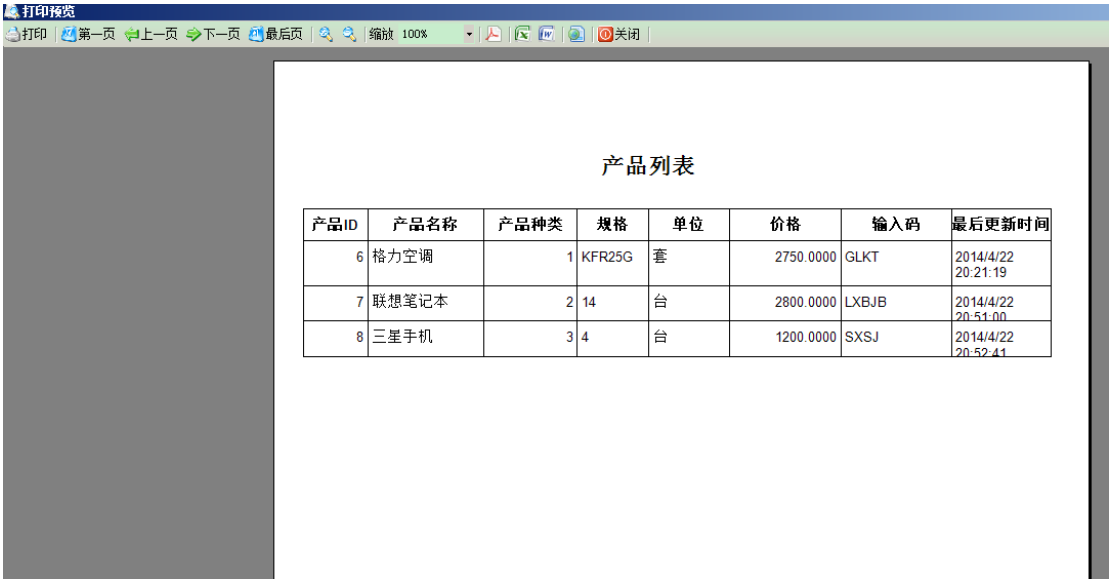


调整格式后，结果如下：

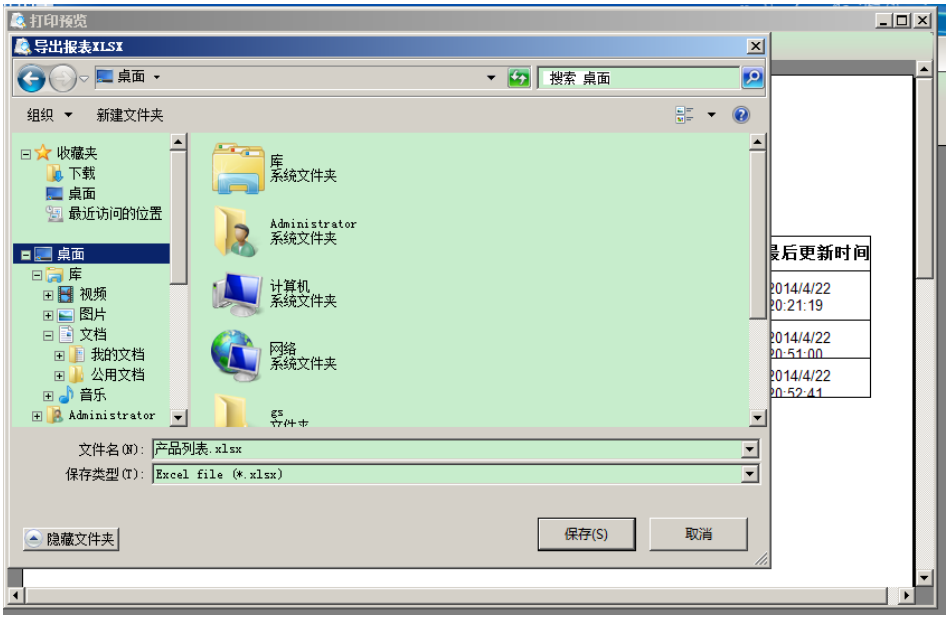


保存，完成报表的设计。

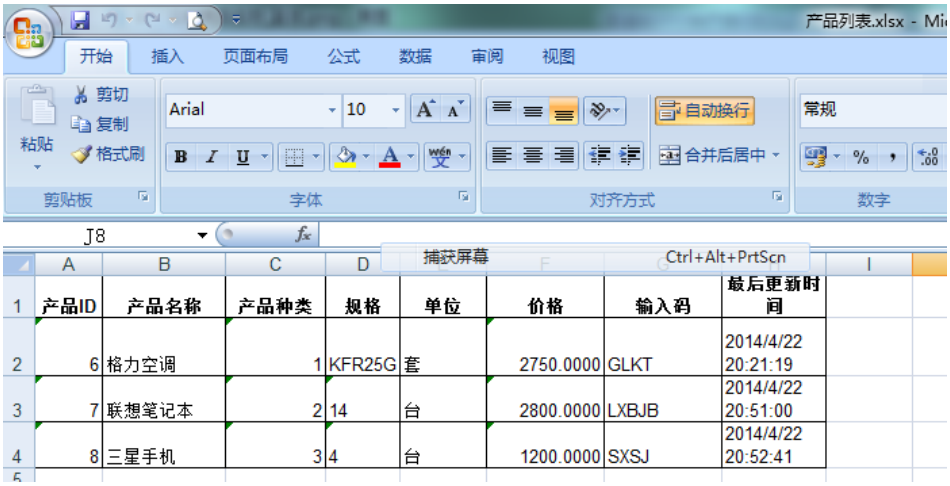
报表实际效果。报表支持 4 中数据导出格式：Excel、Word、Html、Pdf。



如导出为 excel：



导出后格式如下：

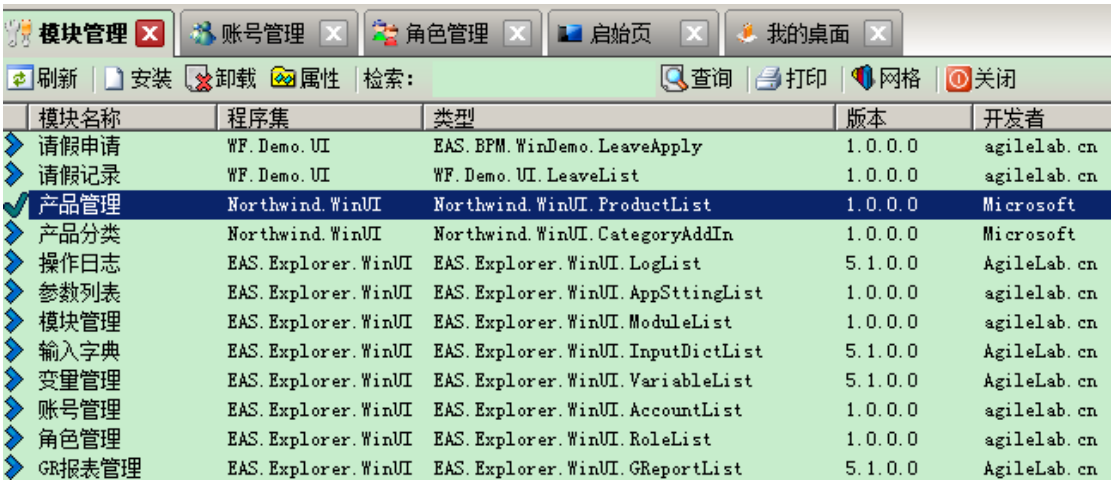


	A	B	C	D	E	F	G	H	I
1	产品ID	产品名称	产品种类	规格	单位	价格	输入码	最后更新时间	
2	6	格力空调	1	KFR25G	套	2750.0000	GLKT	2014/4/22 20:21:19	
3	7	联想笔记本	2	14	台	2800.0000	LXBJB	2014/4/22 20:51:00	
4	8	三星手机	3	4	台	1200.0000	SXSJ	2014/4/22 20:52:41	

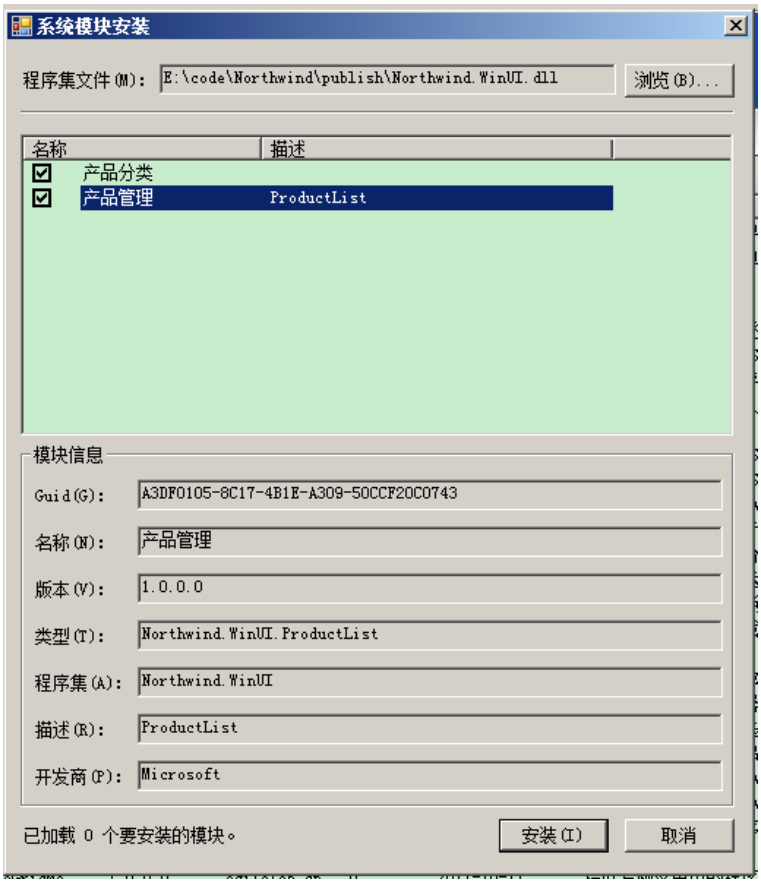
4. 正式运行系统

4.1. 系统部署

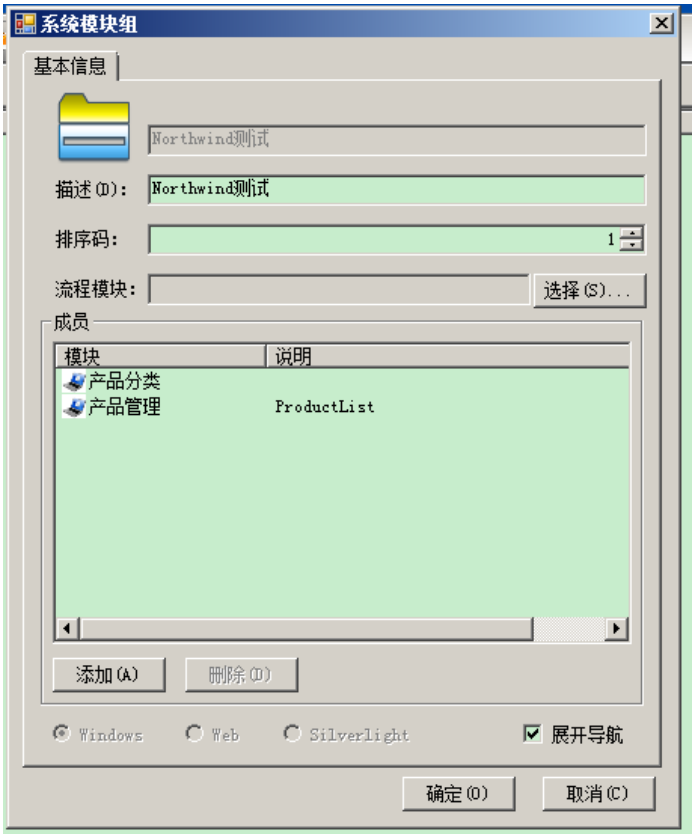
4.1.1. 模块安装



模块名称	程序集	类型	版本	开发者
请假申请	WF.Demo.UI	EAS.BPM.WinDemo.LeaveApply	1.0.0.0	agilelab.cn
请假记录	WF.Demo.UI	WF.Demo.UI.LeaveList	1.0.0.0	agilelab.cn
产品管理	Northwind.WinUI	Northwind.WinUI.ProductList	1.0.0.0	Microsoft
产品分类	Northwind.WinUI	Northwind.WinUI.CategoryAddIn	1.0.0.0	Microsoft
操作日志	EAS.Explorer.WinUI	EAS.Explorer.WinUI.LogList	5.1.0.0	AgileLab.cn
参数列表	EAS.Explorer.WinUI	EAS.Explorer.WinUI.AppSttingList	1.0.0.0	agilelab.cn
模块管理	EAS.Explorer.WinUI	EAS.Explorer.WinUI.ModuleList	1.0.0.0	agilelab.cn
输入字典	EAS.Explorer.WinUI	EAS.Explorer.WinUI.InputDictList	5.1.0.0	AgileLab.cn
变量管理	EAS.Explorer.WinUI	EAS.Explorer.WinUI.VariableList	5.1.0.0	AgileLab.cn
账号管理	EAS.Explorer.WinUI	EAS.Explorer.WinUI.AccountList	1.0.0.0	agilelab.cn
角色管理	EAS.Explorer.WinUI	EAS.Explorer.WinUI.RoleList	1.0.0.0	agilelab.cn
GR报表管理	EAS.Explorer.WinUI	EAS.Explorer.WinUI.GReportList	5.1.0.0	AgileLab.cn


系统模块安装
程序集文件(M):
E:\code\Northwind\publish\Northwind.WinUI.dll
浏览(B)...
名称
描述
产品分类
产品管理
ProductList
模块信息
Guid(G):
A3DF0105-8C17-4B1E-A309-50CCF20C0743
名称(N):
产品管理
版本(V):
1.0.0.0
类型(T):
Northwind.WinUI.ProductList
程序集(A):
Northwind.WinUI
描述(R):
ProductList
开发商(P):
Microsoft
已加载 0 个要安装的模块。
安装(I)
取消

4.1.2. 建立导航



4.1.3. 发布为分布式模式

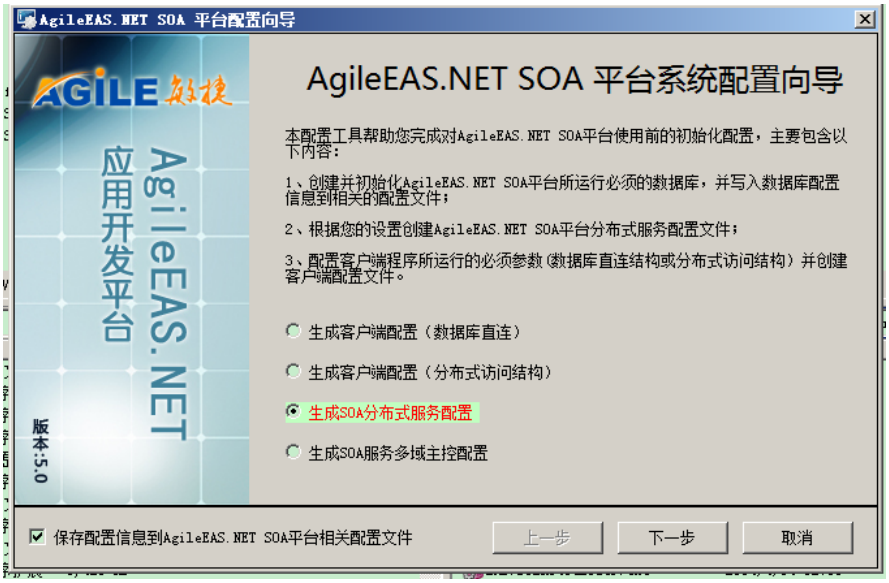
修改 E:\code\Northwind 下的 eas.publish.cmd 批处理文件，把以下内容添加到文件后面：

```
@rem Northwind文件部署
copy @RootDirectory\Publish\Northwind.Entities.dll @RootDirectory\appserver\xclient\files\Northwind.Entities.dll /y
copy @RootDirectory\Publish\Northwind.BLL.Contracts.dll @RootDirectory\appserver\xclient\files\Northwind.BLL.Contracts.dll /y
copy @RootDirectory\Publish\Northwind.Main.exe @RootDirectory\appserver\xclient\files\Northwind.Main.exe /y
copy @RootDirectory\Publish\Northwind.Main.exe.config @RootDirectory\appserver\xclient\files\Northwind.Main.exe.config /y
copy @RootDirectory\Publish\Northwind.WinUI.dll @RootDirectory\appserver\xclient\files\Northwind.WinUI.dll /y

@rem 复制到SOA服务Bin目录
copy @RootDirectory\Publish\Northwind.Entities.dll @RootDirectory\appserver\bin\Northwind.Entities.dll /y
copy @RootDirectory\Publish\Northwind.BLL.Contracts.dll @RootDirectory\appserver\bin\Northwind.BLL.Contracts.dll /y
copy @RootDirectory\Publish\Northwind.BLL.Host.dll @RootDirectory\appserver\bin\Northwind.BLL.Host.dll /y
pause
```

运行 E:\code\Northwind 下的 EAS.Publisher.exe 完成服务端及客户端程序文件的发布

配置服务端

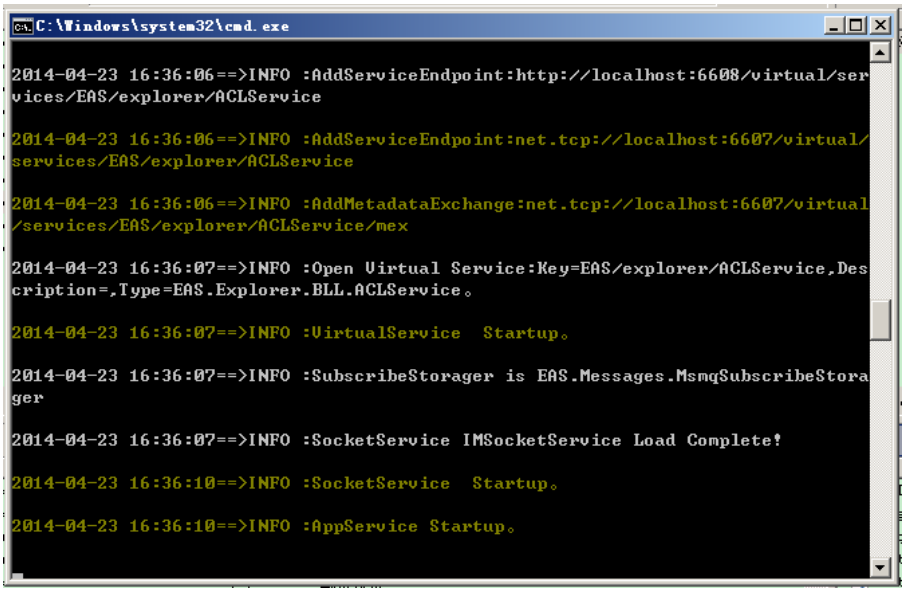
运行 E:\code\Northwind\AppServer\Bin 下的 EAS.Configure.exe，配置为分布式模式，选择第三项【生成 SOA 分布式服务配置】如图


AgileEAS.NET SOA 平台配置向导
AgileEAS.NET SOA 平台系统配置向导
本配置工具帮助您完成对AgileEAS.NET SOA平台使用前的初始化配置，主要包含以下内容：
1、创建并初始化AgileEAS.NET SOA平台所运行必须的数据库，并写入数据库配置信息到相关的配置文件；
2、根据您的设置创建AgileEAS.NET SOA平台分布式服务配置文件；
3、配置客户端程序所运行的必须参数（数据库直连结构或分布式访问结构）并创建客户端配置文件。
生成客户端配置（数据库直连）
生成客户端配置（分布式访问结构）
生成SOA分布式服务配置
生成SOA服务多域主控配置
保存配置信息到AgileEAS.NET SOA平台相关配置文件
上一步
下一步
取消


AgileEAS.NET SOA 平台配置向导
配置应用程序服务器
WCF Service
tcp端口号: 6607
http端口号: 6608
Socket Service
tcp端口号: 6606
数据库类型: SQL2005/2008
服务器: .
用户名:
登录密码:
数据库: Northwind
集成身份验证
订阅消息存储: Msmq
开启WCF服务
6606
缓存更新通知
使用通信令牌
记录通信日志
测试连接
自动更新目录:
浏览
保存配置信息到AgileEAS.NET SOA平台相关配置文件
上一步
下一步
取消

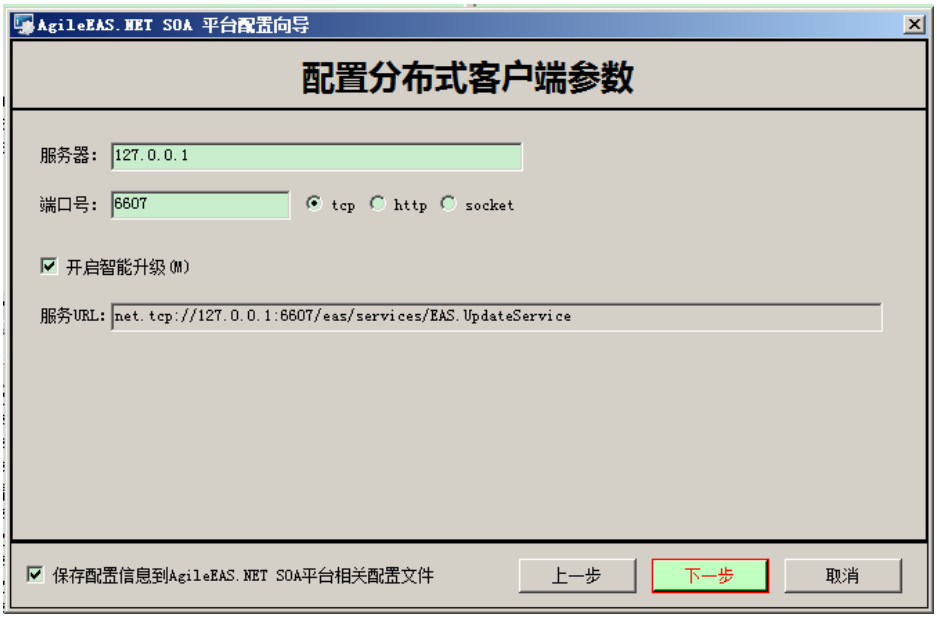

AgileEAS.NET SOA 平台配置向导
<?xml version="1.0" encoding="utf-8" ?>
<configuration>
<configSections>
<section name="eas" type="EAS.ConfigHandler,EAS.MicroKernel" />
</configSections>
<startup useLegacyV2RuntimeActivationPolicy="true">
<supportedRuntime version="v4.0" />
</startup>
<eas>
<configurations>
<item name="Key" value="Value" />
</configurations>
<appserver>
<channel>
<wcf enable="true">
<config tcpPort="6607" httpPort="6608" />
<serviceThrottling maxConcurrentCalls="128"
配置文件: E:\code\Northwind\AppServer\Bin\EAS.SOA.Server.exe.config
保存配置信息到AgileEAS.NET SOA平台相关配置文件
上一步
完成
取消

运行同目录下的 soaserver.start.bat 文件，启动 SOA 服务，如图所示

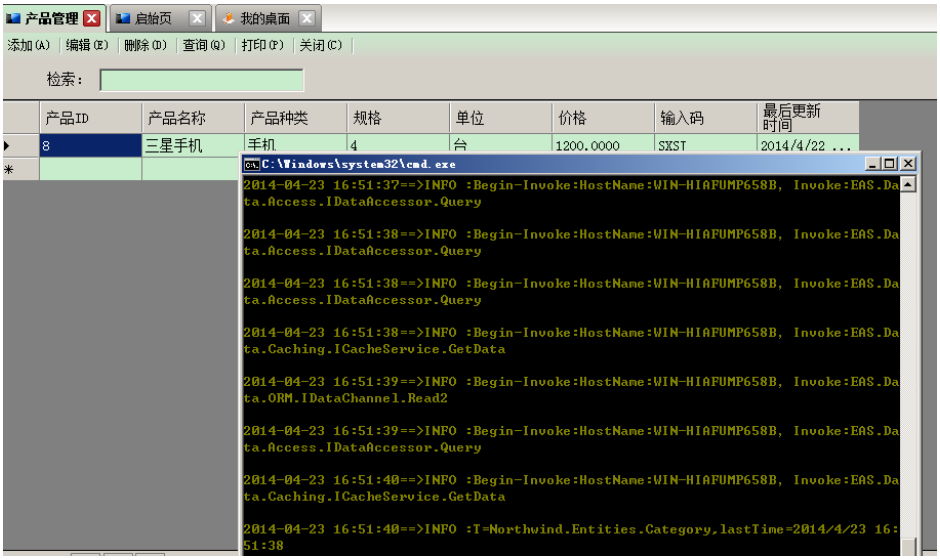


配置客户端

运行 E:\code\Northwind\AppServer\xClient\files 下的 EAS.Configure.exe 进行客户端配置

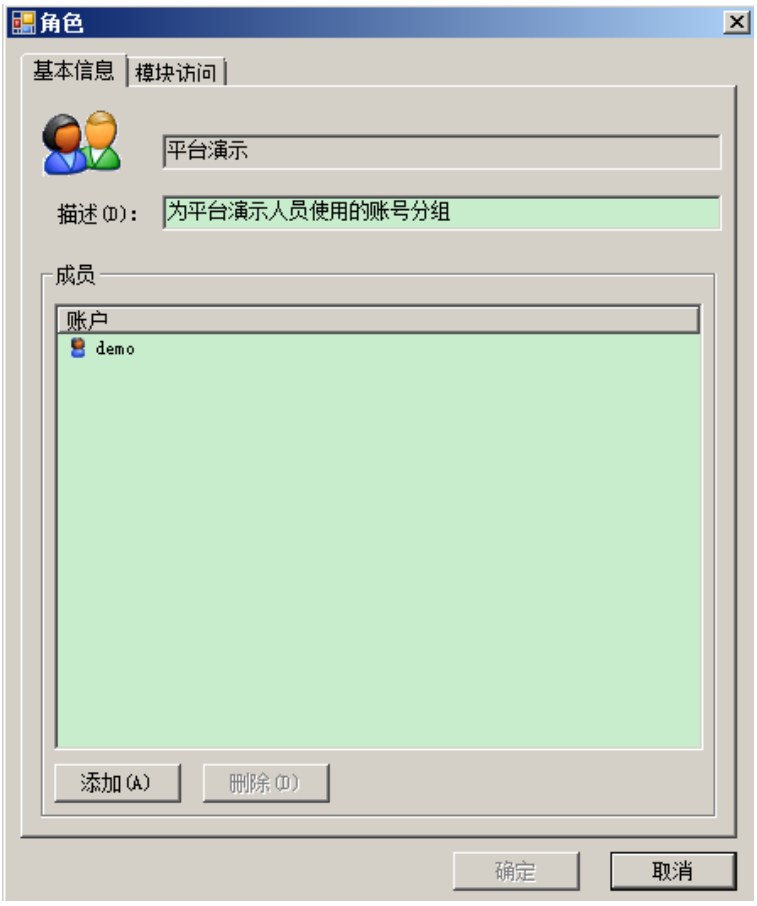


运行同目录下的 EAS.WinClient.Start.exe，启动客户端，登录系统，运行正常



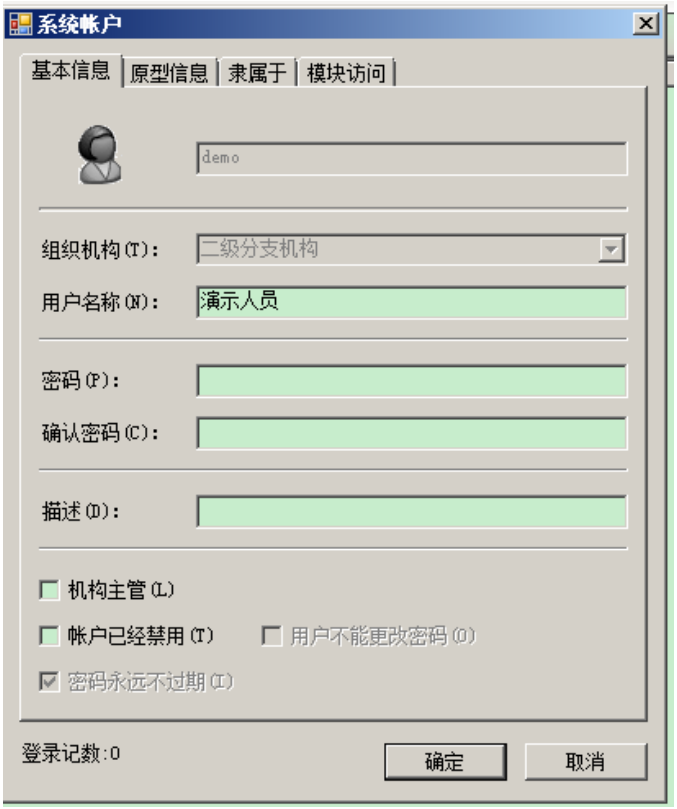
4.2. 角色维护

以 Administrator 账户登录，进行角色维护
建立角色
分配权限
分配成员



4.3. 账户维护

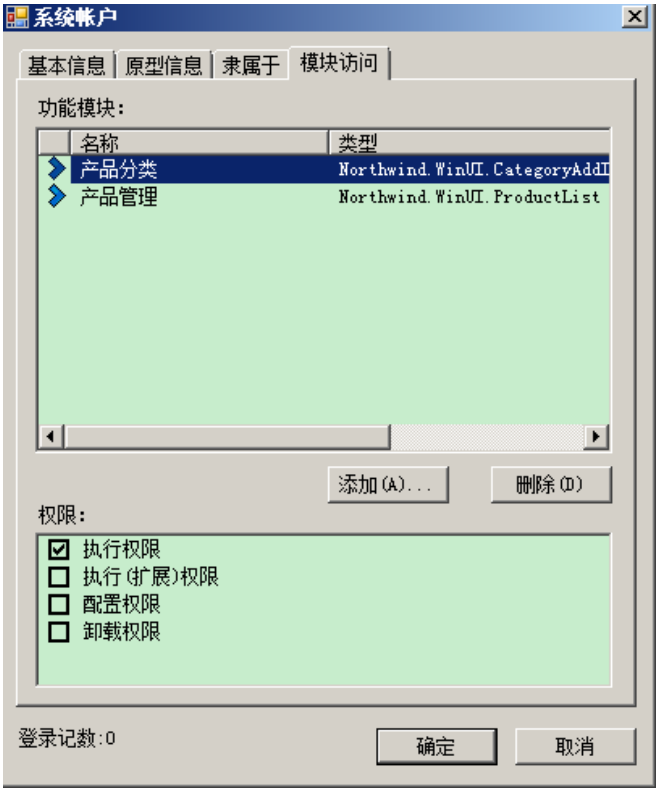
新建账户
制定隶属角色
单独权限分配



4.4. 权限分配

平台既可以在角色管理中对角色统一授权

平台又可以在账户管理中对账户单独授权，管理比较灵活方便。



5. 应用领域

结合我对平台的理解，我认为平台适用面还是很广的，总结为以下四个方面：

5.1. 全新软件项目的快速开发平台

对于全新的软件项目，无论是产品化开发还是项目化开发，都可以考虑用 EAS.NET 平台，可以减少系统管理方面的开发。

5.2. 已有系统的快速二次开发平台

对于在已有成熟系统上进行二次开发来讲，平台更有价值，特别是平台的输入字典可以充分利用原有系统的数据字典，只需要开发新增的录入单据即可，单据上的所有参照项只需要通过平台设置输入字典就可以实现与原有系统的数据字典共享了，报表既可以用原有系统的报表工具（如果有的话）也可以用系统的 GR 报表管理设计报表并发布为功能模块，实现快速二次开发。

5.3. 老旧系统的快速升级开发平台

对于一些技术相对落后、维护工作量大、客户应用较深的系统，也可以在 EAS.NET 平台的基

础上，分模块逐步升级，所有的业务逻辑仍然可以保持与老系统一致，但系统的扩展性、稳定性、可维护性方面则有很大的改进。
另一种系统由于原来主要是针对 sql server 开发的，现在有项目需要支持 oracle 数据库，也可以考虑使用 EAS.NET 对老系统进行扩展升级，如果原来就是基于.net 技术，用 C#开发的，整体开发工作量相对比较小。

5.4. 多系统应用快速整合开发平台

对于一些大型集团企业，内部使用了许多系统，但作为集团领导经常需要结合多个系统的综合数据，通过 EAS.NET 平台可以搭建企业内部的数据交换中心和数据仓库，并可以在此基础上建立一套决策分析系统。如果能够充分发挥 EAS.NET 工作流平台的优势，还可以通过工作流把各个系统有机联系起来。

6. 适用对象

此处的分类方式以对.net 开发技术掌握的程度为标准。

6.1. 零基础

主要针对那些小微软件企业的老板们，他们多数擅长业务，不懂技术，或者只是简单的了解技术，没有实力建立一个强大的开发队伍，多数只有 1~2 名技术人员，这些既负责一些简单的开发工作，还要负责大量的客户服务。这些公司可以考虑使用 EAS.NET 平台作为基础开发平台，技术人员只要能够会用 C#写窗体和简单的业务逻辑即可。

6.2. 初级水平

主要针对一些解除开发 1~3 年的开发人员，已经基本能够做一个模块级的开发，但由于没有时间积累，没有自己的整体开发框架。这类人员也可以考虑把 EAS.NET 平台掌握后跳槽到新的公司做他们的技术负责人，把 EAS.NET 平台作为公司的开发平台，为公司快速带来效益，也可以使自己的收入在短期内实现翻番。

6.3. 中级水平

主要针对已经从事开发工作多年的开发人员或者项目团队的负责人，他们多数有一定的技术水平，也形成了自己的开发框架，但框架存在着这样或者那样的不足，为解决框架问题而忙碌不休。建议这些朋友可以考虑选择 EAS.NET 平台作为自己或者项目组的开发平台，提高开发交付效率，解放自己，用更多的时间去休闲，去陪陪家人、朋友。

7. 结束语

以上为个人对 EAS.NET 平台的认识和理解，EAS.NET 平台还有可以改进提高的地方，希望喜欢这个平台以及使用这个平台的朋友们能够发挥各自的优势，让平台更强大、更好用、更有效，为我们大家带来更大的回报。在此谢谢作者的无私奉献，最后，预祝 EAS.NET 平台发展的越来越好。

欢迎大家沟通交流：QQ：709565426；邮箱：wy_erp@126.com