

DotNet WinForm UI 开发框架

插件式通用版开发框架开发文档

*****文档信息*****

编写日期: 08/26/2013

更新日期: 08/26/2013

版 本: 13.08.26

作 者: Allen Chen

*****文档内容*****

1. 框架整体介绍
2. 解决方案说明
3. 演示如何进行二次开发
4. 如何实现发现新版本后客户端自动更新
5. 系统的注册码功能
6. 后台数据库相关的介绍
7. 部分功能截图预览
8. 用户使用过程中遇到的问题点以及解决方案

前言

在着手开发这个框架之前，作者在一家生产手机的公司做 Web 开发，主要是开发公司的一套 MES 系统，全称 Manufacturing Execution System，中文叫“生产制造执行系统”，一直从事 Web 方面的工作，比如 Asp.Net; JavaScript 等等，后来因故跳槽到现在的公司，现在的公司是一个集团，规模非常大，刚来到这家公司的时候看了看公司职员上班的情况，觉得自己在这里应该可以好好的发挥一下了。很快，一晃时间过了半年，事实证明作者当时的感觉是错误的，这个公司主要是来料加工，很多的客户，每一个客人都要求整理一个“系统”出来，并且每一个客人的要求都不一样，另外公司分 4 个厂区，每个厂区都有独立的服务器，有一些程序是 4 个厂区都要使用的，那么就是连接的服务器不一样，也就是说同一个程序要编译 4 个执行文件，如果是安装包的话还得做 4 个安装包，真的可以算是五花八门，有时候一个小小的 exe 文件就算是一个系统，源码都找不到，再加上很久以前的开发人员使用的语言各不一样，有 VB6, C++, Visual FoxPro, Asp, Delphi，也没有开发文档，维护起来真的是很痛苦；作者本来是搞 C# 的，面对这些语言我实在是感到迷茫，对于现在的工作感觉我们就是一个缝补衣服的角色，哪里破了就在哪里打个补丁，补丁上面破了就再打补丁，直到这个功能得以实现；并且上班不让上网，更别说 QQ 啊，MSN 啊等等，有时候很小的一点问题如果在一些比较专业的 QQ 群里面请教一下的话，几分钟就可以解决问题，可是在这里只能埋头摸索，实现不行就等着下班了回家在网上查查资料，吐槽一下，哪有搞 IT 的人不让上网的呢？开发人员被禁网，那这样的公司里面的技术面临的结果就是---OUT。

开发人员开发出来的程序都是放在一个文件服务器上，新建了一个 FTP 的目录，然后用户需要使用的时候便联系作者通过 FTP 复制一个执行文件到用户本地硬盘，然后给用户的电脑配置 ODBC 来连接服务器，后来看公司 FTP 服务器上面的 80 多个文件夹，每一个文件夹里面都算是一个系统，作者才想到要是有一个比较好的框架可以兼容这些不同的小程序就好了，将这些零零散散小程序归纳在一起，维护的时候好歹就这一个项目，不用搞得到处都是 exe 文件，这也是这套开发框架产生的背景。

作者承诺

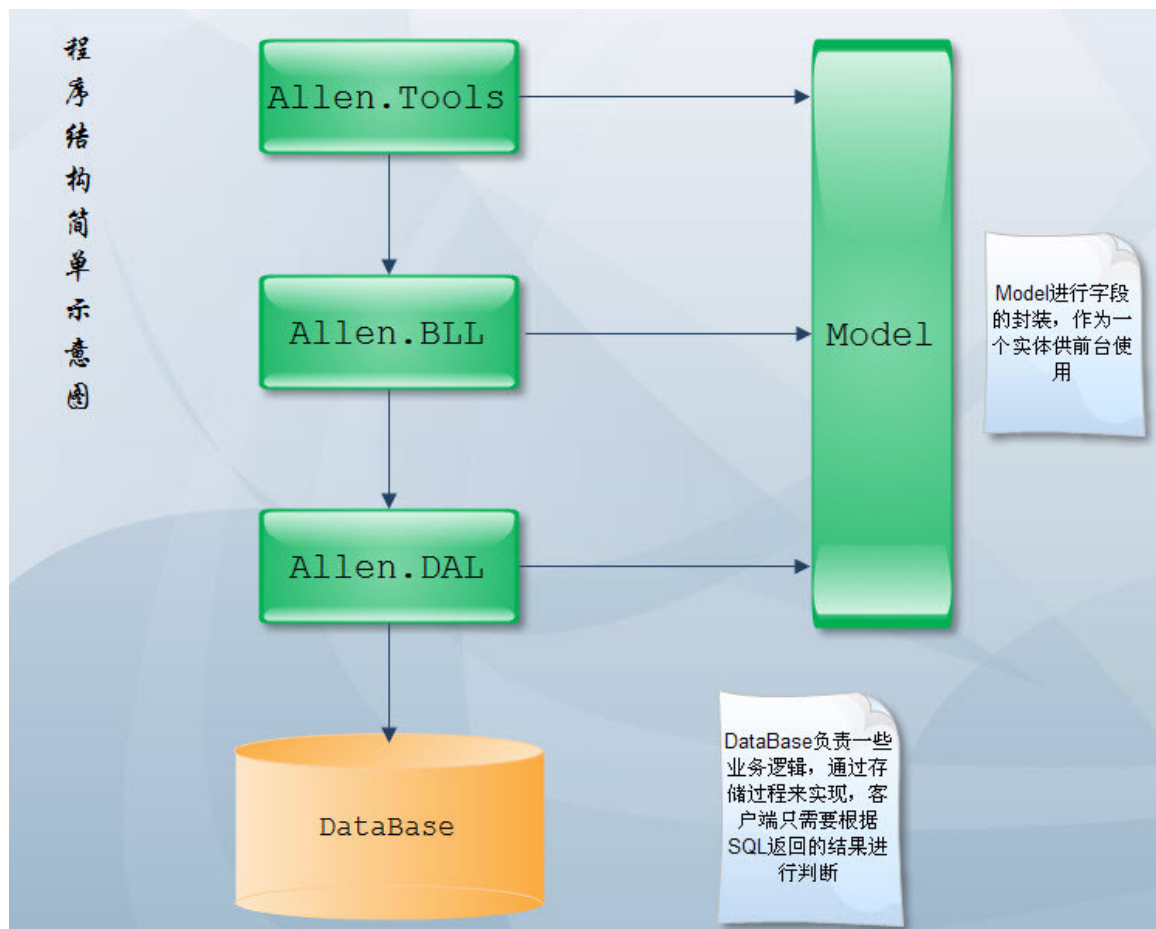
1. 购买之后帮您配置后台数据库，不管您的 PC 上面所安装的数据库是哪个版本，无任何影响。
2. 如果是 VS 版本不一致导致解决方案无法打开，作者协助您进行版本的转换工作。
3. 框架程序以源代码的形式发送给您。
4. 数据库文件分为数据库文件 (*.mdf) 以及 SQL 脚本文件，避免由于数据库版本不同导致附加不了数据库。
5. 测试的基本数据以 SQL 脚本的形式发送给您，您只需要按照顺序执行 SQL 语句之后即可体验实际效果。
6. 在使用过程中如有疑问作者可以提供 QQ 协助，帮您提出一些实现思路以及方法，但是不提供再次开发。

框架简介

该框架采用逻辑上的三层模式开发，业务处理和数据处理完全分开，UI 层通过业务层来调用数据层的相关函数实现数据操作，前台界面方面使用的多文档的布局，类似选项卡的样式，可以随意的拖动窗体到任何地方，舍弃了 MDI 的窗体模式(试问现在谁还会采用 MDI 的方式呢?)，整个项目的工具栏比较统一，每一个界面都是一样的工具栏，这也为后续的权限控制做了一个铺垫；美化方面采用了 DevExpress 的风格，程序里面只是继承了 DevExpress，用到了其中的美化效果，客户端并不需要安装 DevExpress，只要程序运行的根目录里面存在 DevExpress 的几个 DLL 支持文件即可，如果需要增加新的功能，只需要将创建的窗体名称在菜单里面进行注册，然后通过权限管理来进行授权后即可使用，不需要其它特别的处理。

亮点：所有的 Form 都被编译成为一个相应的 DLL 文件，开发人员对此 DLL 文件进行自定义描述，比如 Name, GUID, Version, Manufacturer, Description, TCode, FrmUrl 等等，用户只需要将这些生成之后的 DLL 文件放置指定目录，主程序自动去读取这些 DLL 文件里面的信息，从而加载其内容，实现插件化模块管理，减少各个模块之间的依赖程度，便于维护和管理。

项目结构：



(图 1)

框架适用人群：

1. 处于学习阶段的同学，可以参考项目里面的编码风格、代码技术等等，项目里面用到了面向对象(封装、继承、多态)、反射、三层、文件 IO、存储过程、注册表、加密算法操作等等技术，可用来接外包等等。
2. 经常接外包的朋友，一般的中小型管理系统可以直接套用该框架，通用的功能作者全部都做完了，你只需要根据你的需求来开发实际的需求功能，简化你的操作，节约你的时间。

主界面预览



图(2)

顶部标志栏：可以设置公司的 Log 标志或者系统名称，图片可以随意更改。

中间工具栏：快捷菜单、窗体 Code 命令等等工具。

左侧工具箱：根据登录用户的权限动态将菜单加载到此工具箱，点击即可打开窗体。

中部工作区：每次打开的窗体都是加载到这里，多个窗体以选项卡形式呈现出来，可以拖拉。

下方状态栏：显示当前登录信息，以及时间和当前的版本状态。

解决方案介绍:

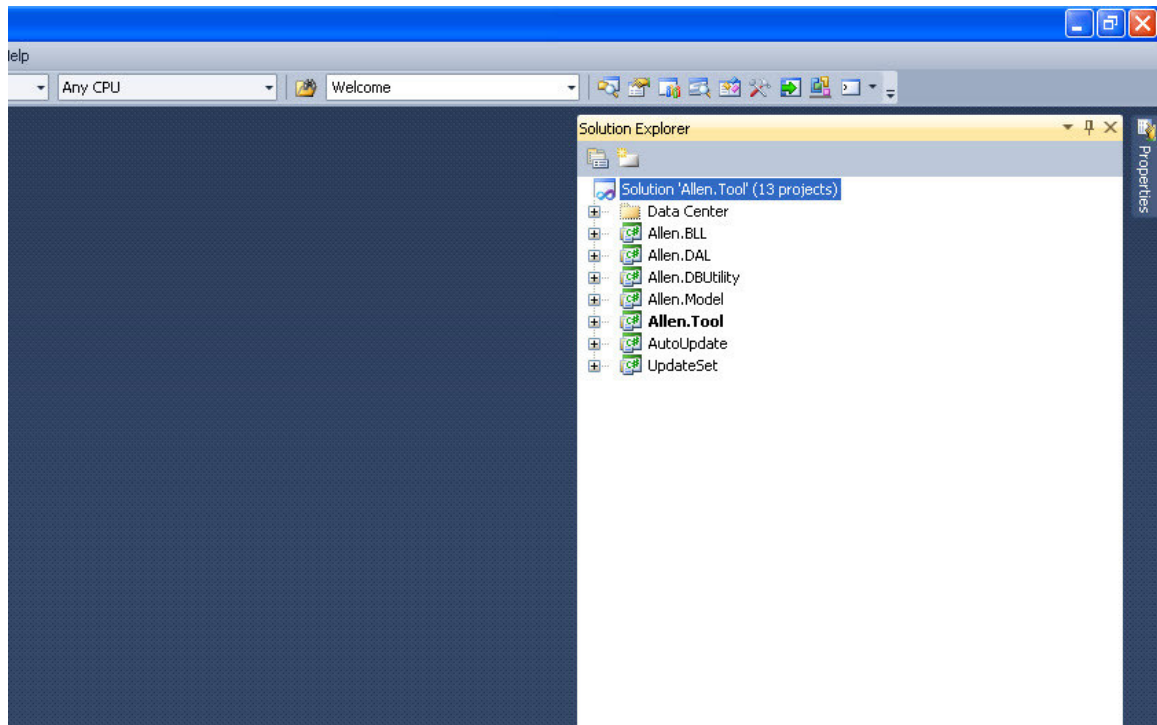


图 (3)

Data Center: 存放各个 Form, 也就是每一个类库。

Allen.DAL: 主要存放所有与数据库相关的操作, 对数据库的所有操作都在这里完成。

Allen.BLL: 将 Allen.DAL 里面的函数传递给 Allen.Tools, 单纯的传递工作。

Allen.Model: 保存着一些实体类, 都是通过封装的字段。

Allen.Tools: UI 层, 所有的前台控制都在这里实现, 需要操作数据库的时候就调用 Allen.BLL。

RegisterTool: 注册工具, 通过获得机器的 CPU 编号和硬盘编号来生成注册码, 如果程序没有注册则只能在试用期内使用, 过了试用期便无法继续使用, 只要通过这个程序生成的注册码注册以后才可以, 这一部分作者并没有花费太多的精力在上面, 在“破解无处不在”的大环境里面作者显得太渺小了。

AutoUpdate: 这是一个自动更新版本的部分, 软件的自动升级和更新是至关重要的, 作者认为 WinForm 的软件少了自动更新的话算不上一个软件; 作者现在的公司有很多 C++ 写的 DOS 程序, 现在更新程序的时候还得将硬盘拆下来, 通过非常复杂的命令来进行一个文件的拷贝。这个自动更新的程序可以自由设置更新的文件集合、更新文件的目录等等, 客户端检测到有新的版本出现则会立即启动这个更新的程序来进行文件更新, 至于是否更新到最新版本, 取决于用户的选择, 如果不更新最新版本, 则依旧使用旧版本的程序。

UpdateSet: 后台管理员配置程序更新的信息, 比如程序名称、最新版本、更新内容等等。

介绍如何进行二次开发:

1. 在项目里面增加你需要的窗体, 为了后续能够更好的维护以及逻辑更清楚, 作者建议在增加窗体文件的时候对窗体进行归类, 比如跟用户相关的界面都放在一个名称为“Users”的文件夹里面, 跟工作有关的界面都放在一个名称为“Work”的文件夹里面, 这样方便维护, 现在我们创建一个新的菜单, 用来演示具体的操作步骤。

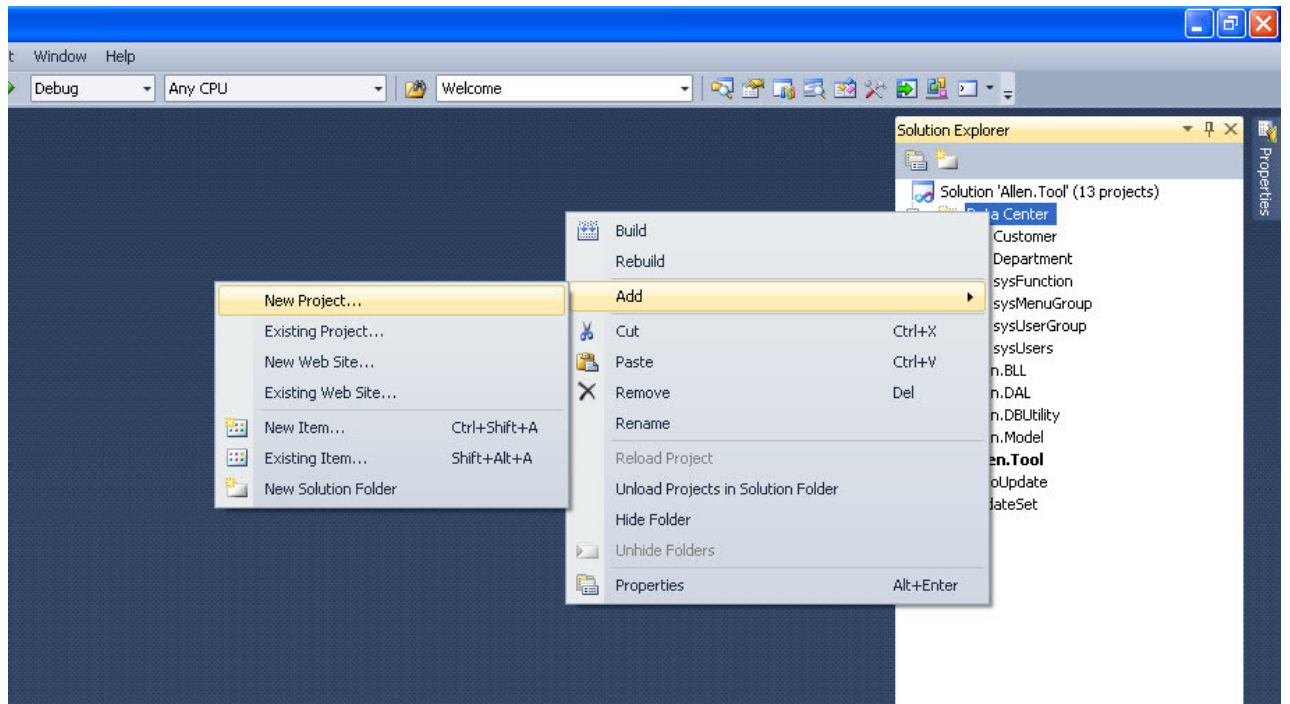
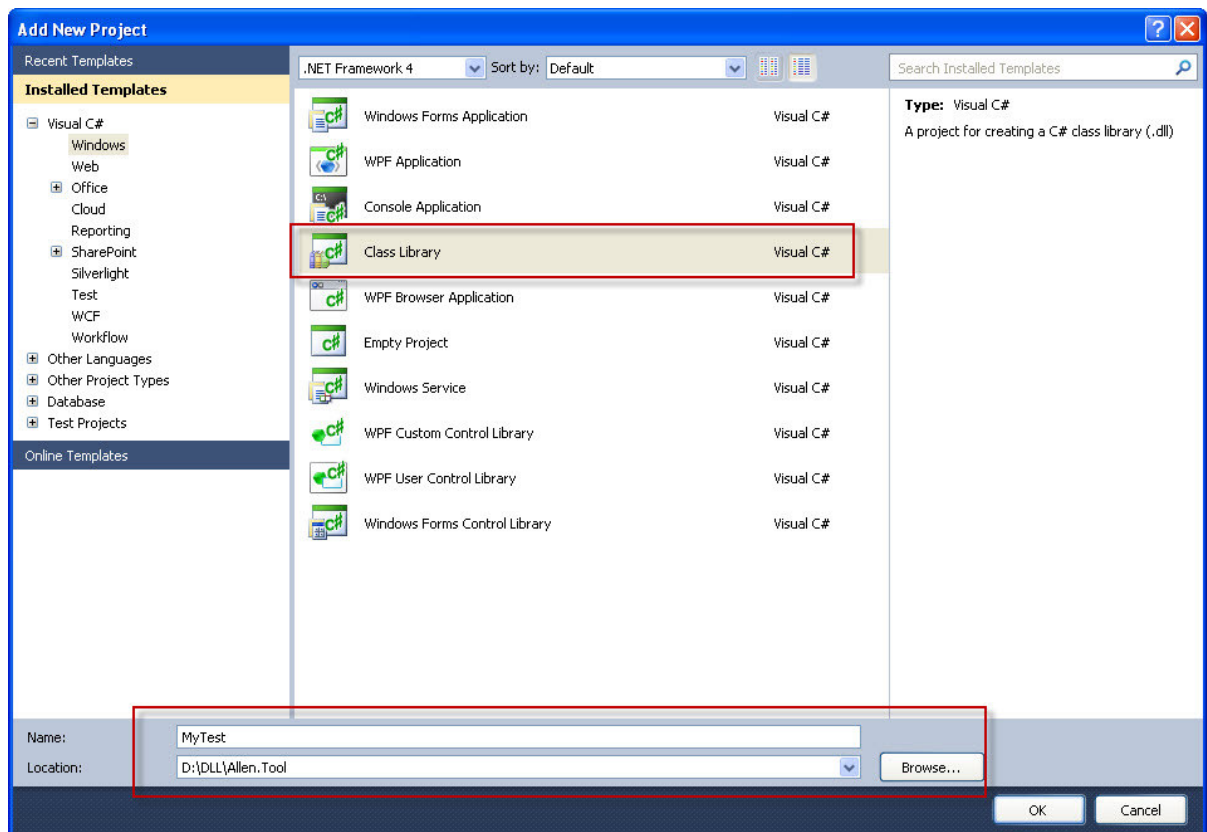
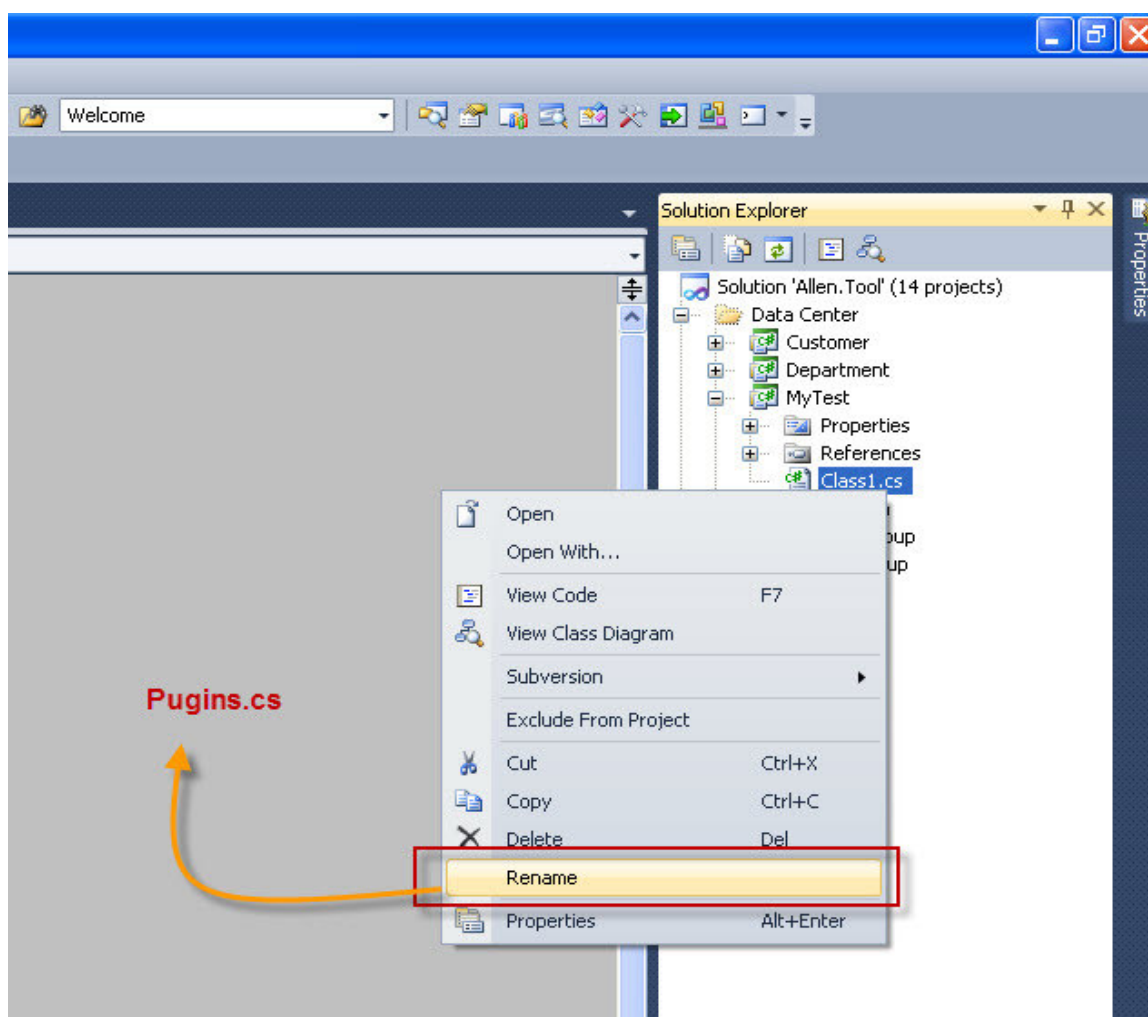


图 (4)



(图 5)

2. 这里我们选择新建的项目类型为 Class Library，命名为 MyTest，点击 OK，



(图 6)

3. 新创建的类库默认产生一个 Class1.cs 的类文件，我们将这个类文件重命名为 Pugins.cs

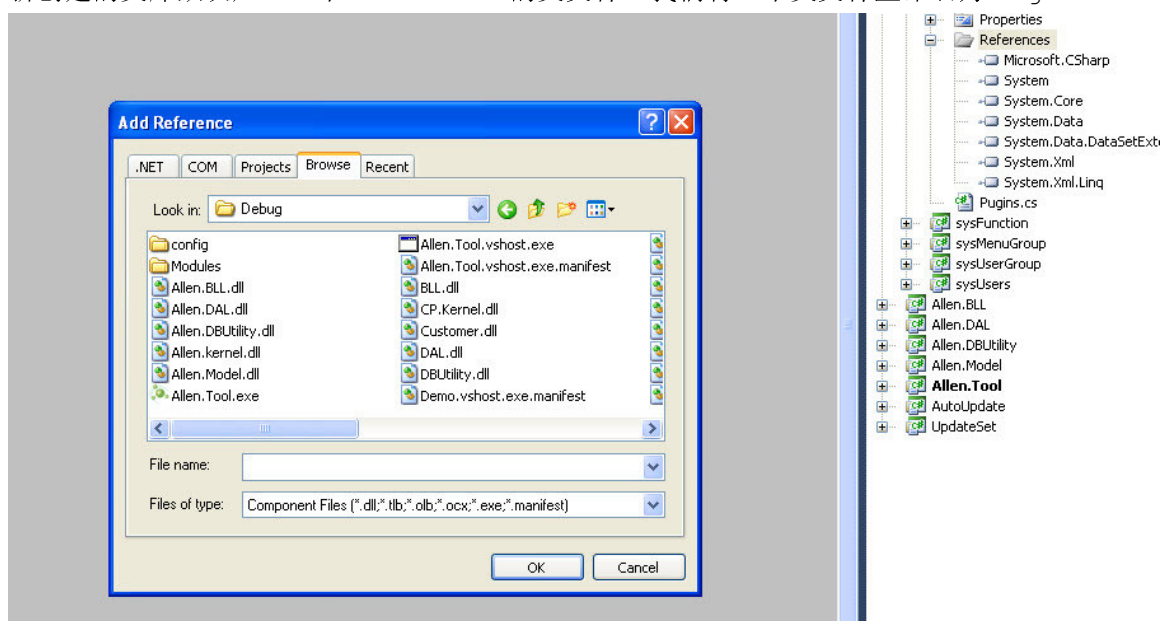
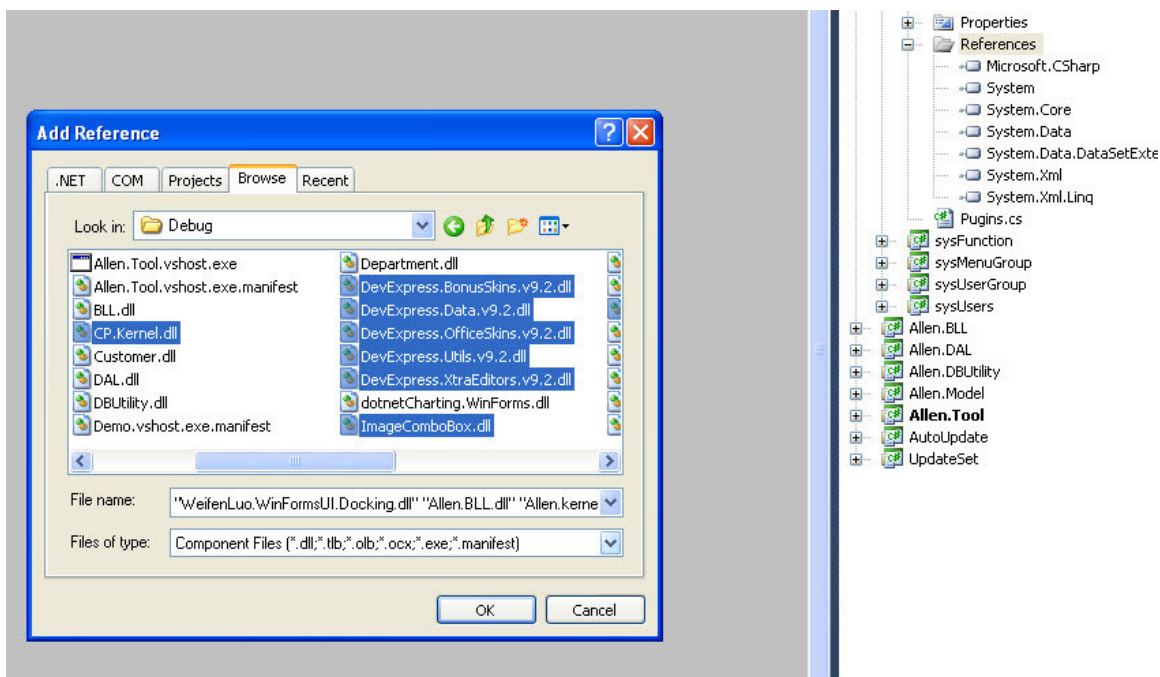


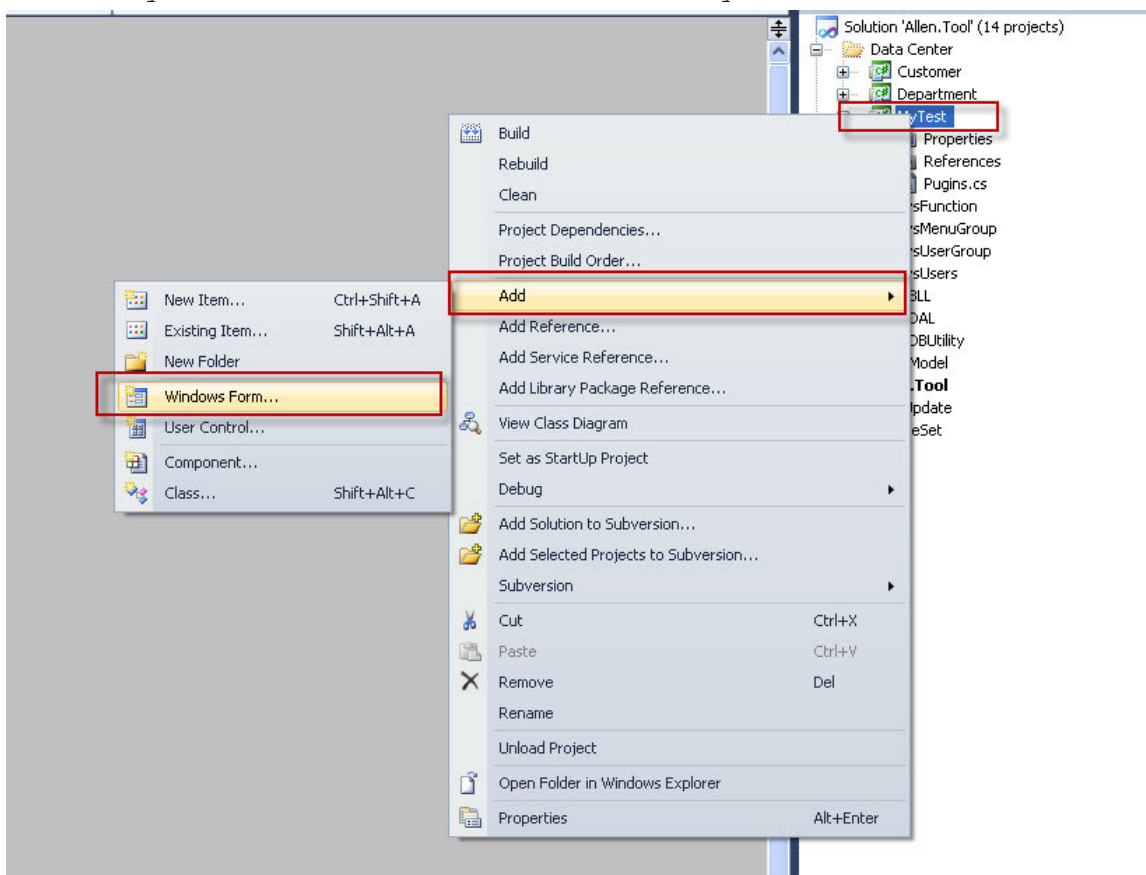
图 (7)

4. 添加引用，项目里面所用到的一些 DLL 文件都在根目录里面，我们浏览到相应的目录，然后将其引用到项目里面来，如下图：

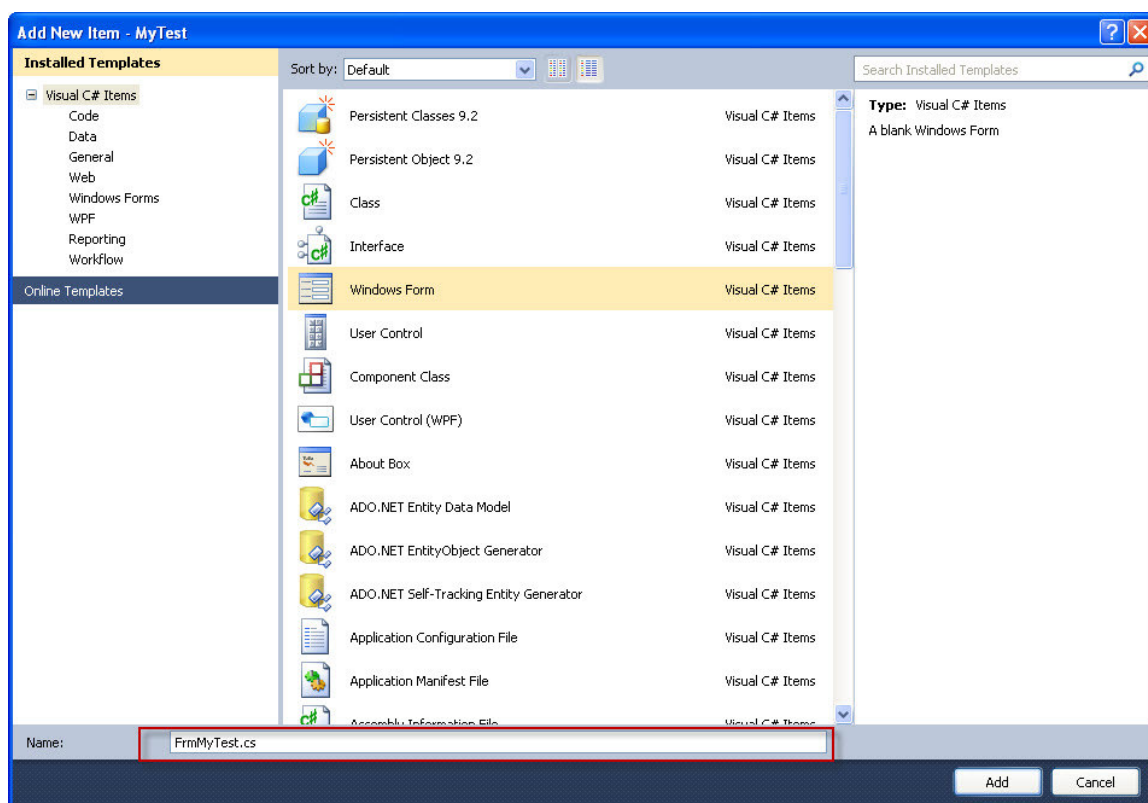


(图 8)

5. 在新创建的 MyTest 类库中创建一个新的 Form，名称为 FrmMyTest

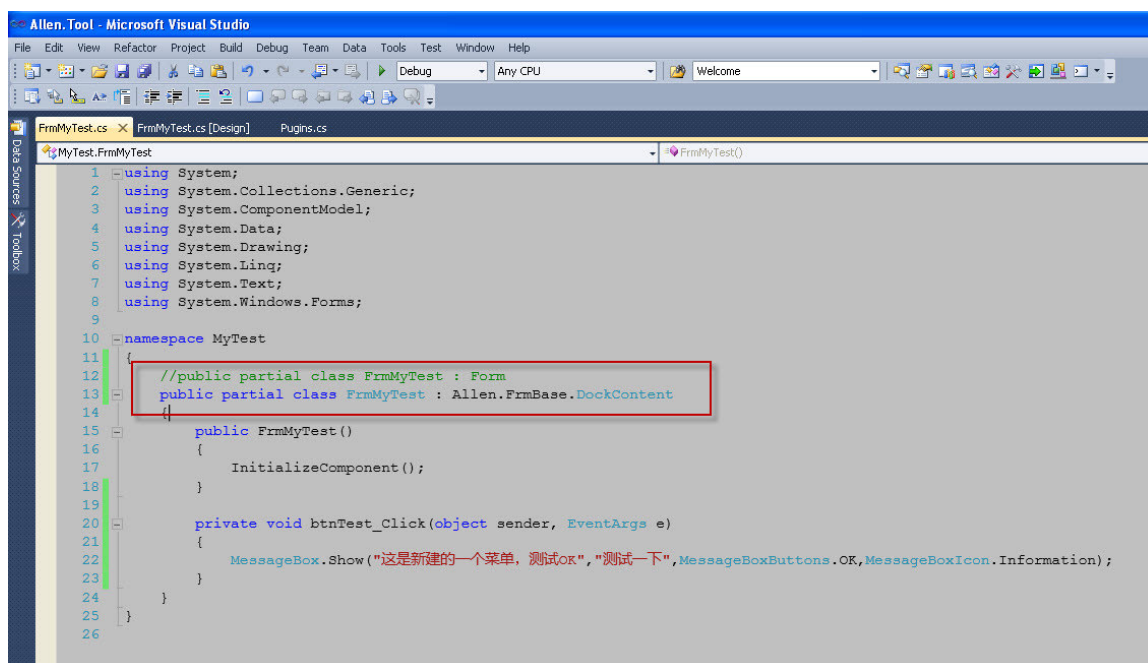


(图 9)



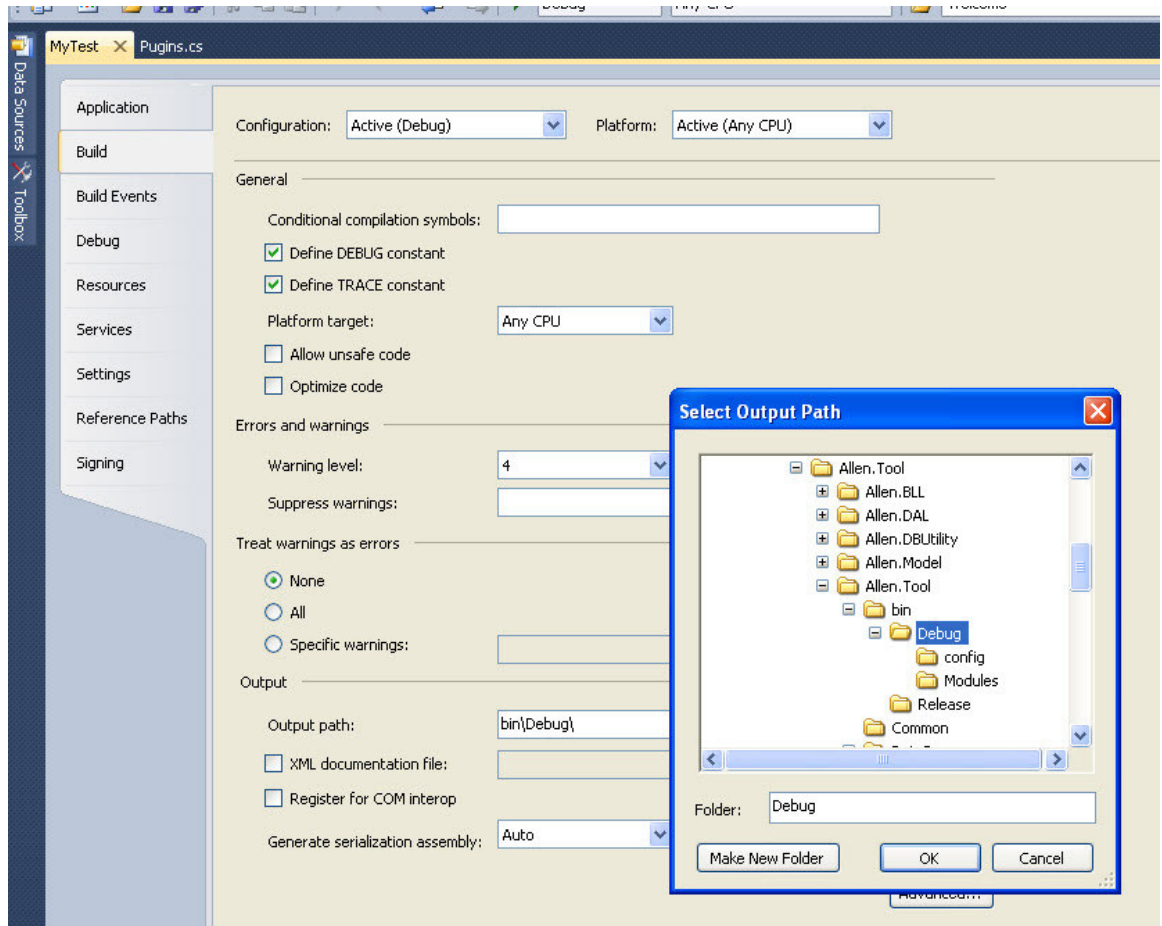
(图 10)

6. 在代码视图里面更改 Form 的继承来源，如下图所示：



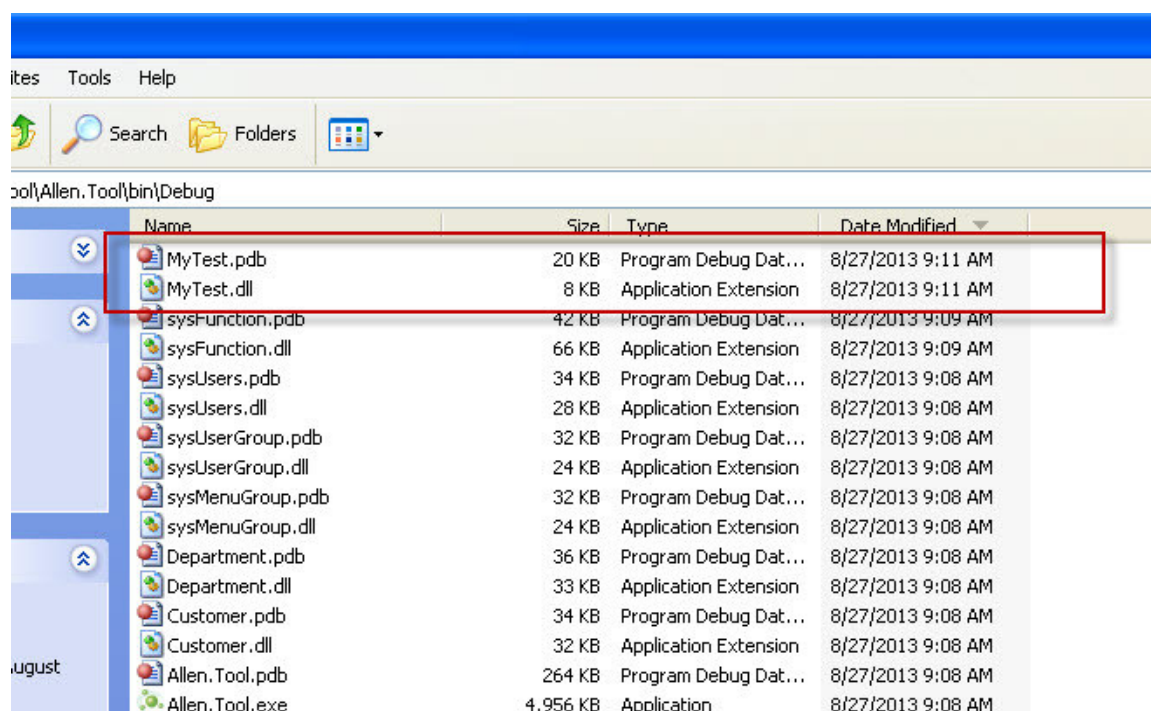
(图 11)

7. 更改 MyTest 这个类库的生成路径，右键点击 MyTest 类库，选择属性，找到 Build 页面，将输出路径修改为我们的主目录的 DeBug 文件夹，本示例上面显示的是 Allen.Tool.bin.Debug，然后点击保存。

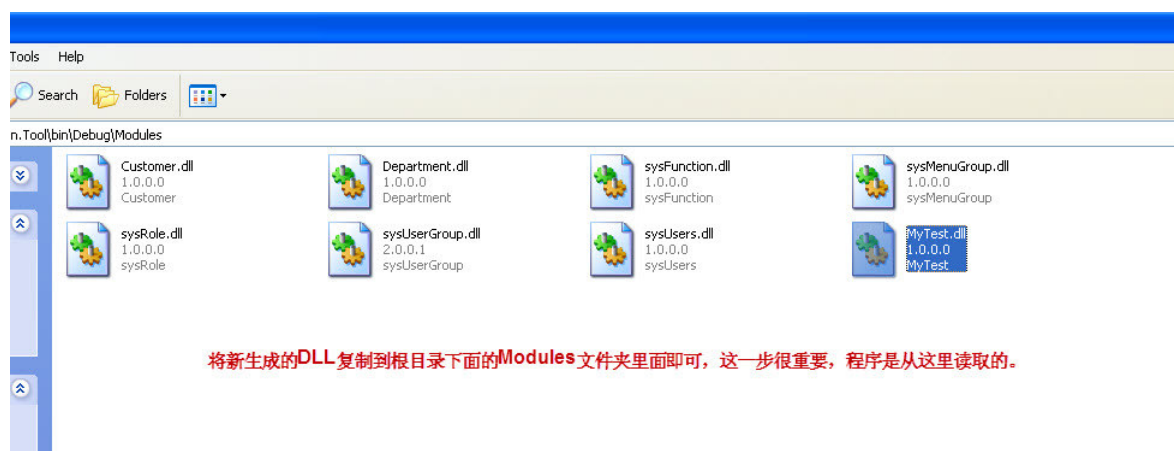


(图 12)

8. 我们可以发现在主目录的 DeBug 文件夹里面多出来 MyTest.dll 这个文件，我们将其复制到 Modules 文件夹里面。

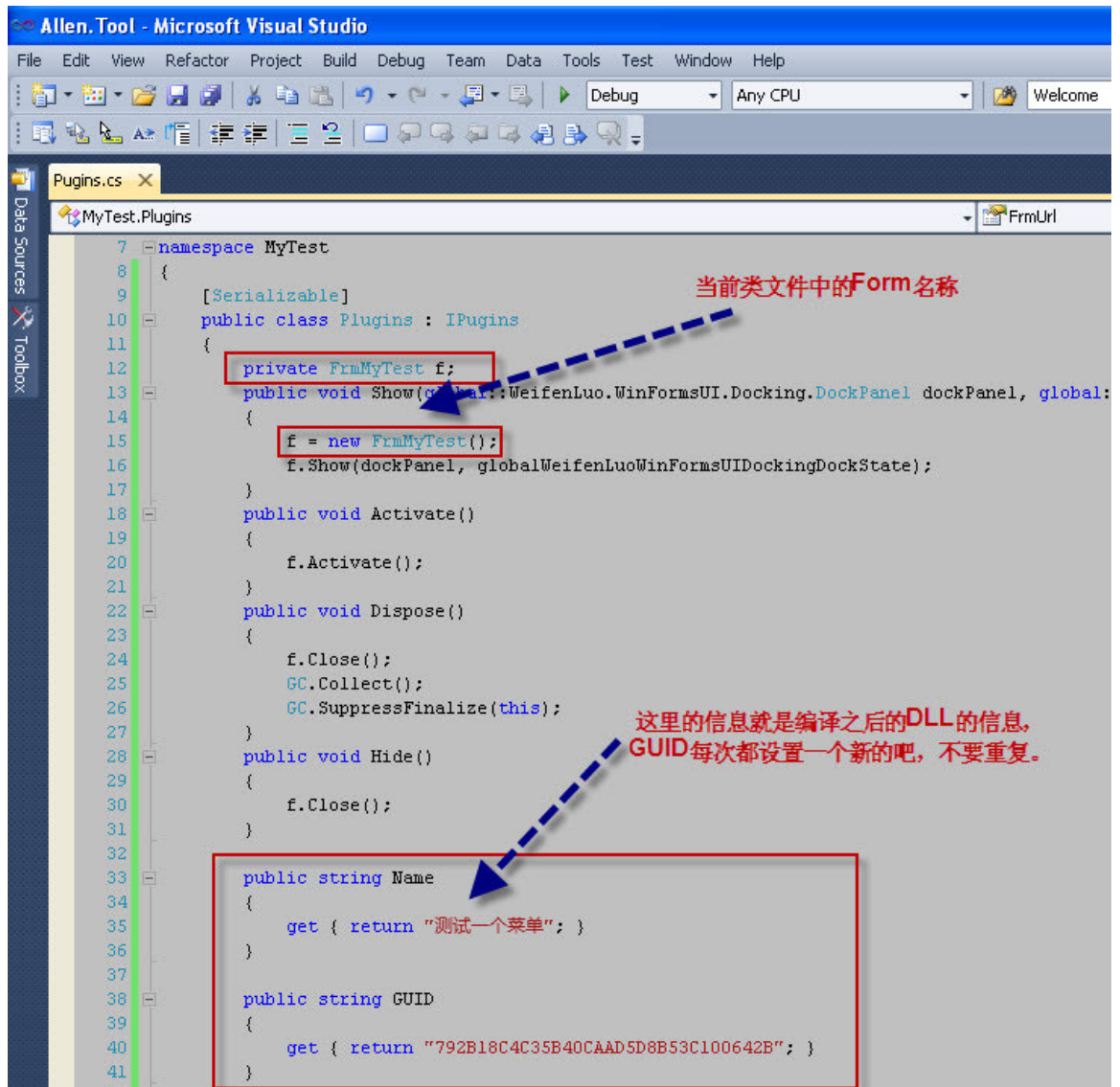


(图 13)

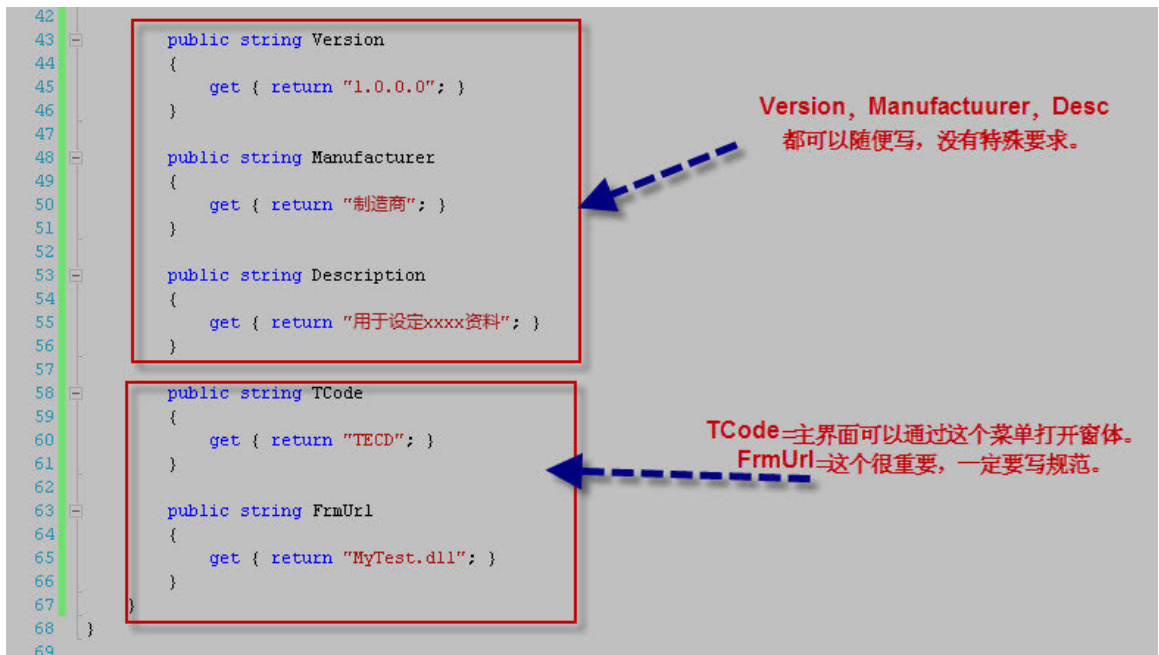


(图 14)

9. 修改我们的 Plugins 这个类文件，将其它类库里面的这个 Plugins 内容直接复制过来即可，需要更改以下内容，窗体名称要改为当前的窗体，名称，GUID，Version, Description, TCode, FrmUrl 等等，这些信息都将显示在相关的界面上，比如菜单构建等，所以请认真填写，其中 GUID 请通过 VS 中的 Tool-> Create Guid 来获取，



(图 15)



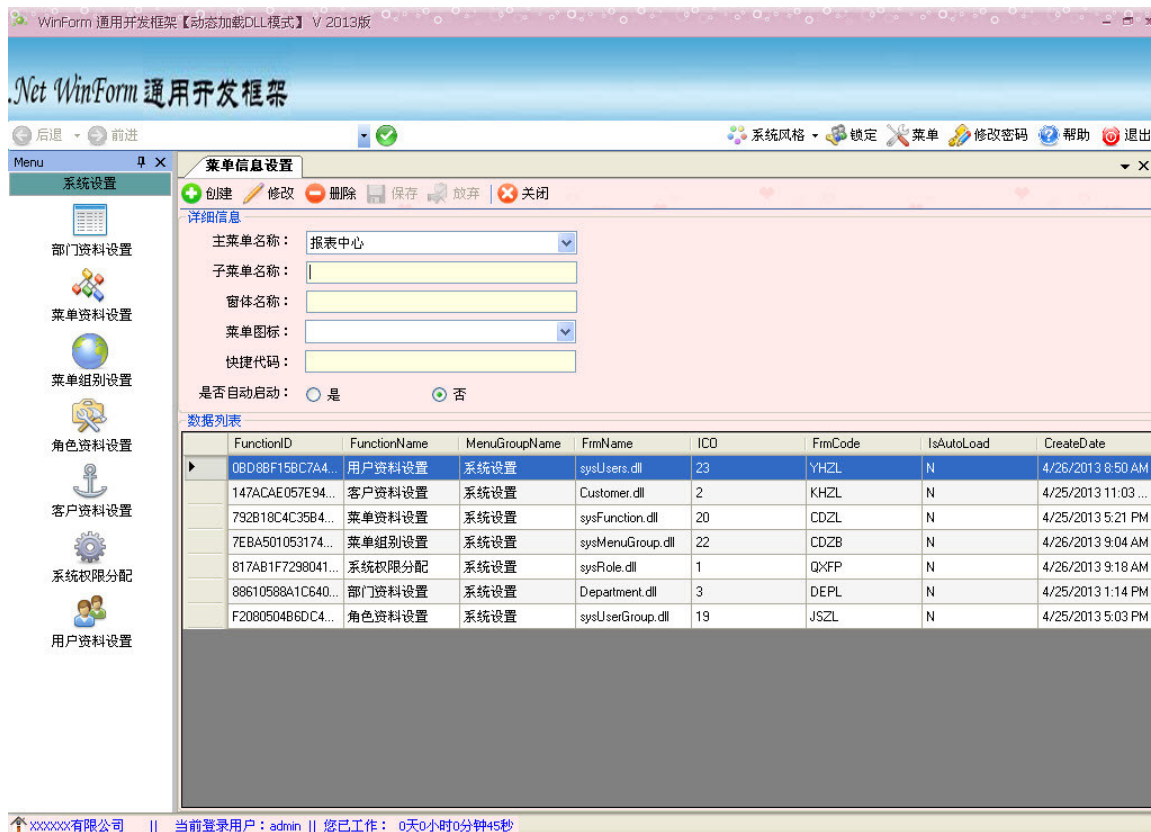
(图 16)

10. 现在我们的窗体创建完成, 我们运行一下看看效果, 首先我们必须创建菜单, 将这个 DLL 注册到系统中, 进入系统, 以下是主界面。



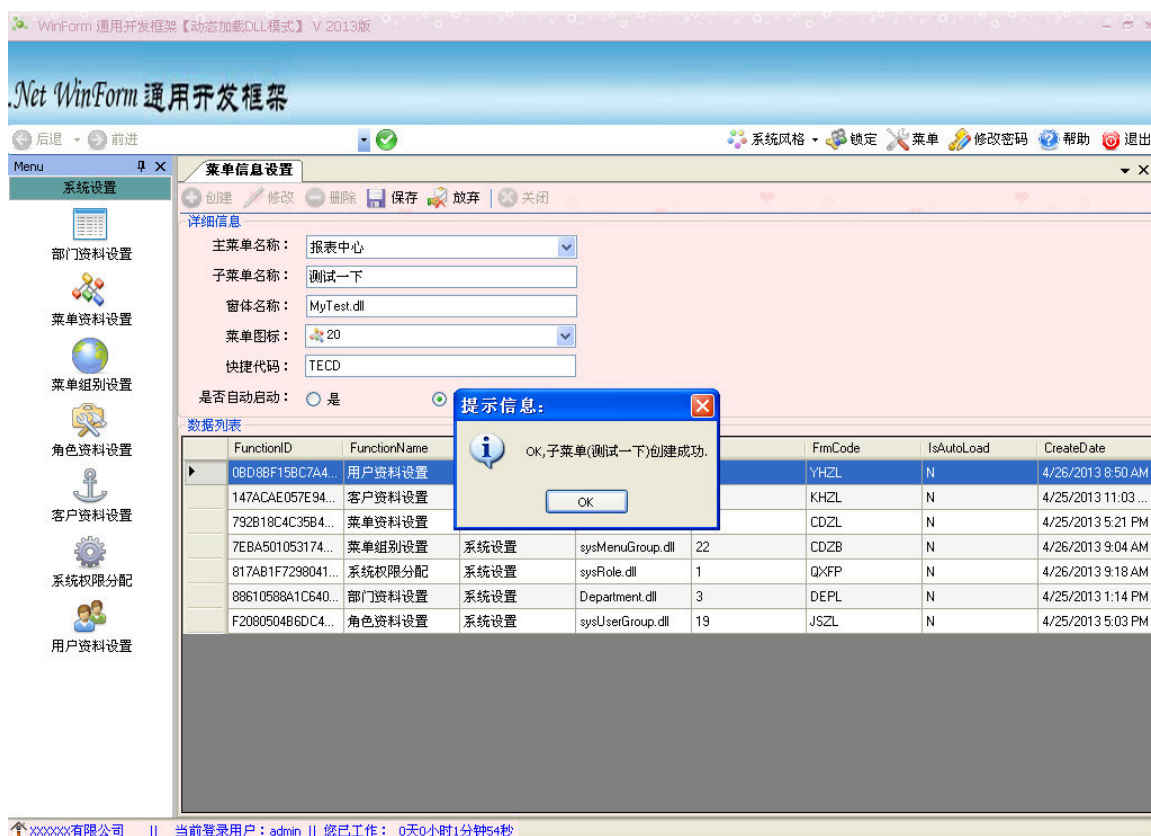
(图 17)

11. 我们创建一个菜单，点击左侧工具箱里面的“菜单资料设置”，这些名称都可以随便更改。



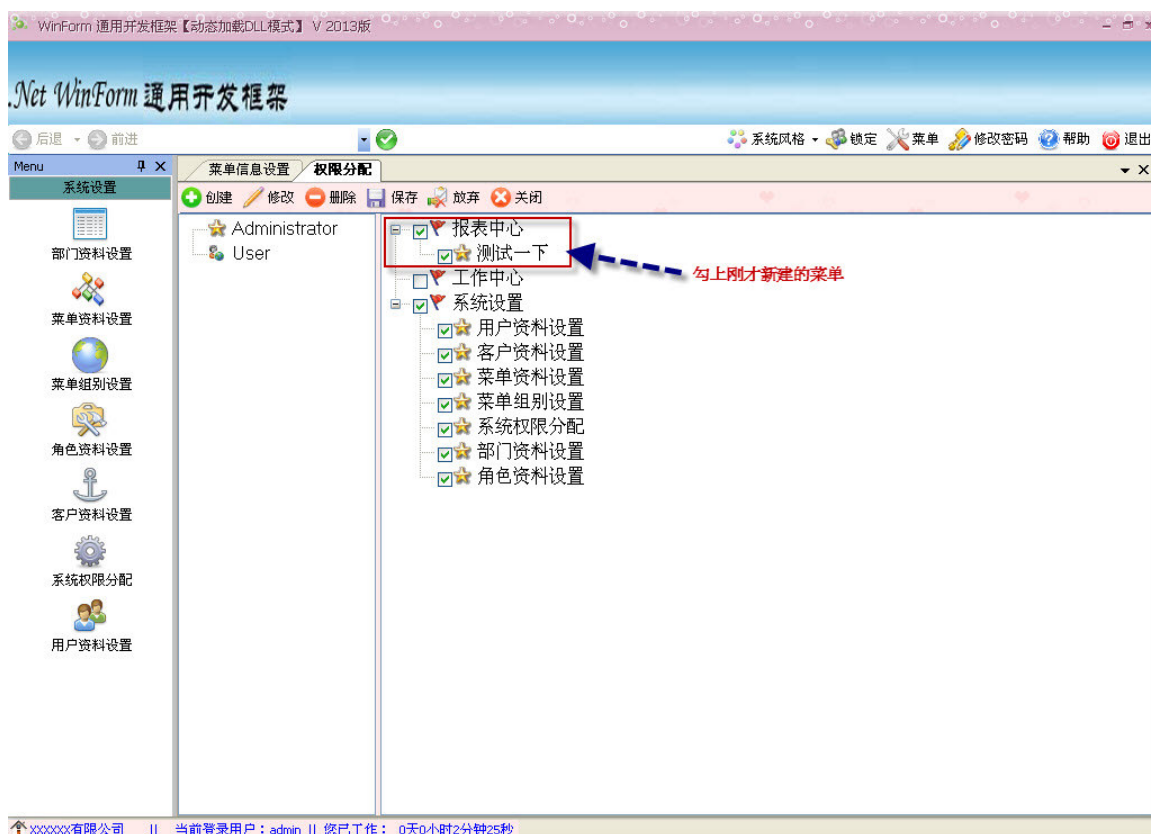
(图 18)

12. 主菜单名称表示当前的子菜单的上级，子菜单名称就是显示在工具箱中的名称，窗体名称比较重要，就是我们生成的 DLL 名称，菜单图标表示在工具箱中显示的图标，快捷代码表示在主界面通过这个快捷代码可以直接打开窗体。

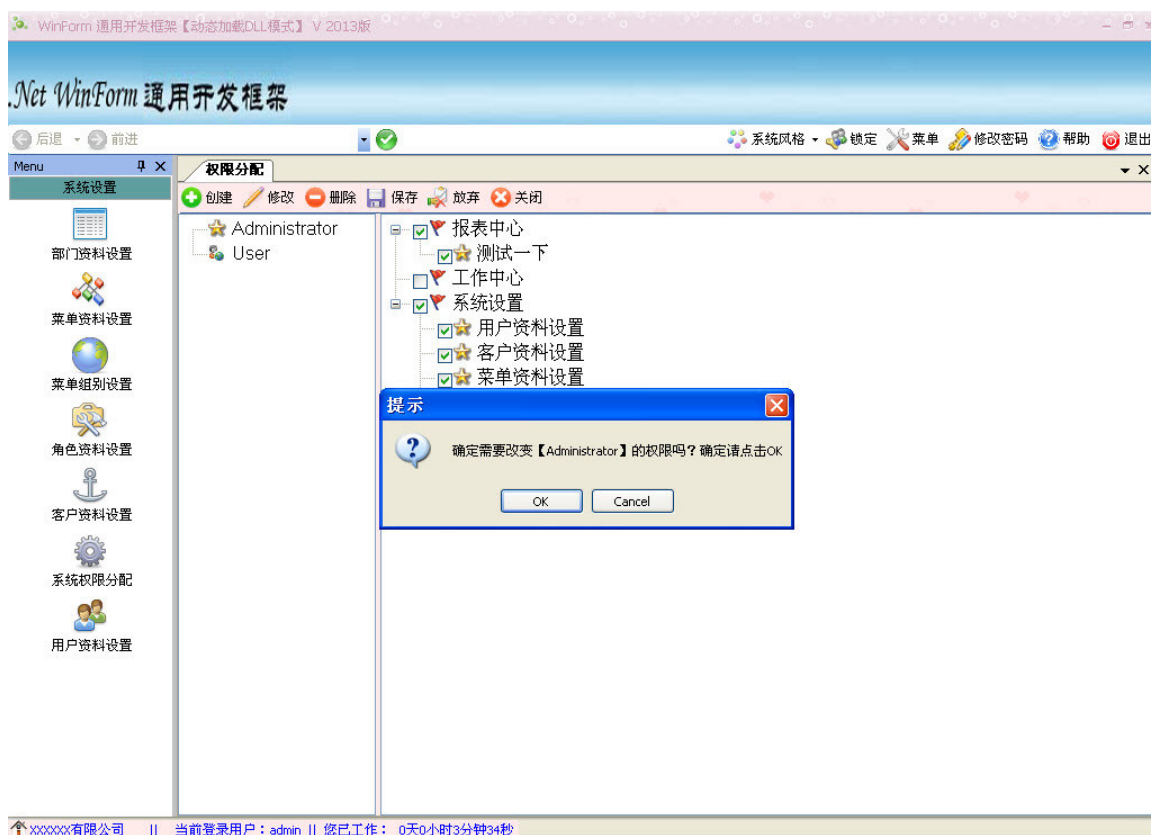


(图 19)

13. 现在需要对角色授权，让其可以访问刚才说创建的菜单，我们选择 Administrator 这个角色，点击这个角色名称之后会发现右边的菜单列表里面我们刚才创建的菜单并没有勾选上，这就说明无法访问这个菜单，现在我们将它勾选，点击保存。



(图 20)



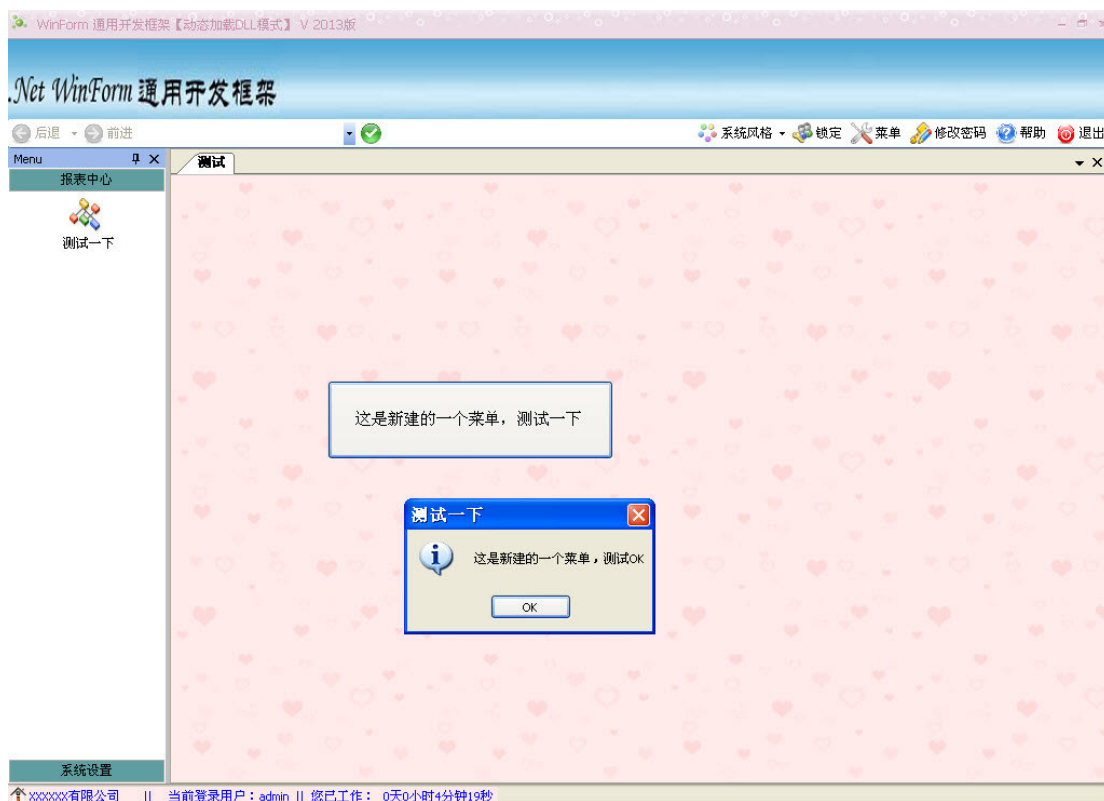
(图 21)

14. 权限设置完成之后我们刷新一下工具箱，可以看到最上面已经出现了“报表中心”以及其

子菜单“测试一下”，这就说明我的权限配置成功了，现在我们可以使用刚才创建的这个菜单了，点击“测试一下”这个图标，发现右中间已经正常加载出这个窗体。



(图 22)

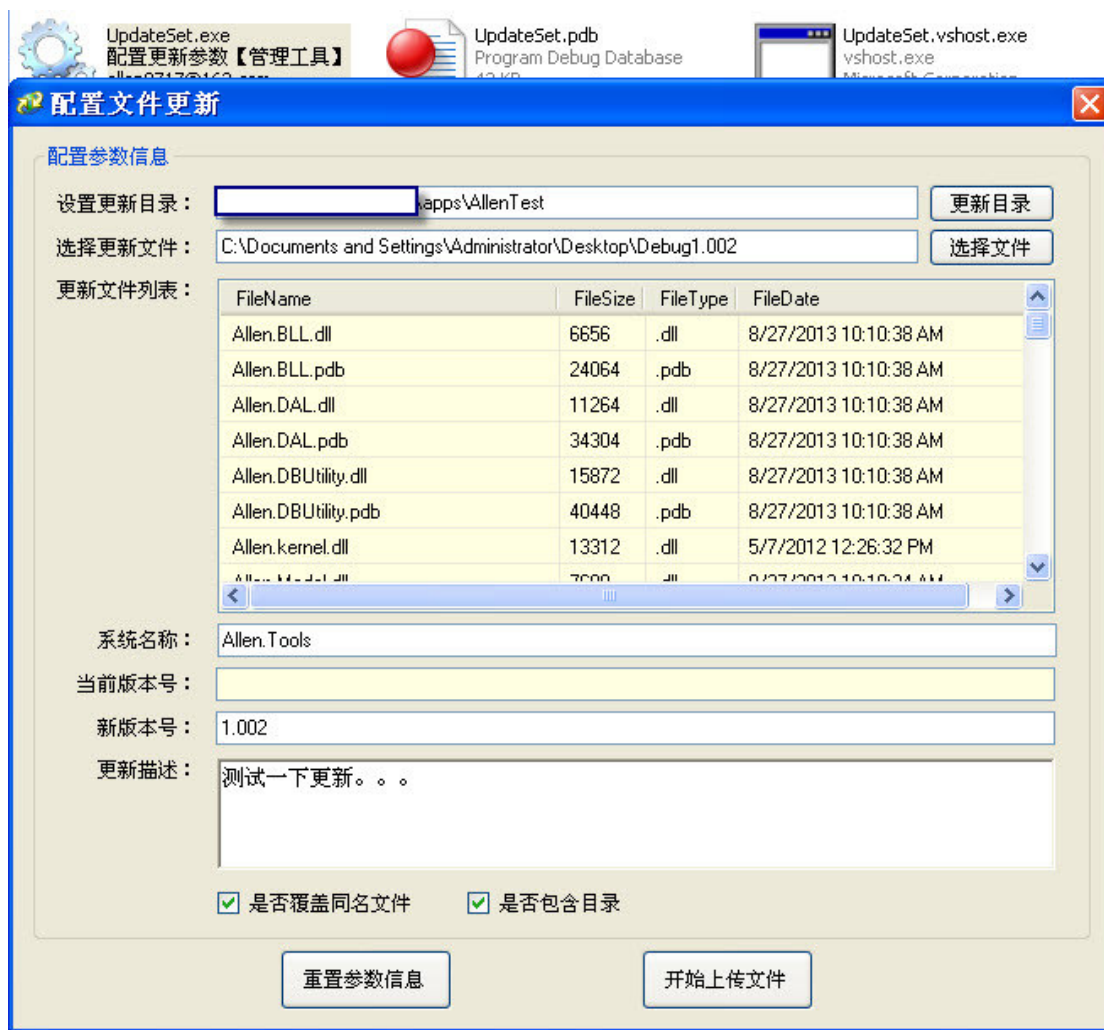


(图 23)

介绍如何实现自动更新:

当用户数量很多的时候,程序的自动更新就显得格外的重要,该框架的自动更新的实现原理如下:

1. 服务器上记录最新版本的情况,包含程序名称、当前的版本号、更新内容描述、更新日期等。
 2. 程序里面记录着每一次的版本号,每次修改程序之后便将该版本号适当的增加,同时将最新的文件放置文件服务器上,并且将服务器上面的版本也适当的增加,增加的幅度要和程序里面增加的幅度完全一致。
 3. 客户端打开程序的时候进行版本验证,这一步必须是连接到服务器上面验证,获取到服务器上面目前最新的版本号,再结合本地程序里面的版本号进行比较,如果本地版本号小于服务器上面的版本号则执行更新动作。
 - 4.** 更新的动作是独立的一个更新程序,这个程序里面实现了下载文件相关的一些事情,待最新文件被下载下来以后便立即重新启动主程序,弹出登录界面。
-
5. 解决方案中有一个 AutoUpdate 的项目,这个项目负责自动从文件服务器上下载最新的文件,
 6. 解决方案中有一个 UpdateSet 的项目,这个项目负责对文件更新进行配置。
 7. 达到程序自动更新的条件:
 - (A) 主程序根目录有这个 AutoUpdate.exe 文件,也就是 AutoUpdate 生成的执行文件
 - (B) 服务器上面的程序版本比客户端的版本新
 - (C) 客户端有权限去访问文件服务器
 8. 具体的实现方法:
 - (A) 将负责自动更新业务的 AutoUpdate.exe 文件放在主程序的根目录。
 - (B) 通过 UpdateSet 程序配置文件更新参数,如下图所示:



(图 24)

更新目录：文件服务器路径

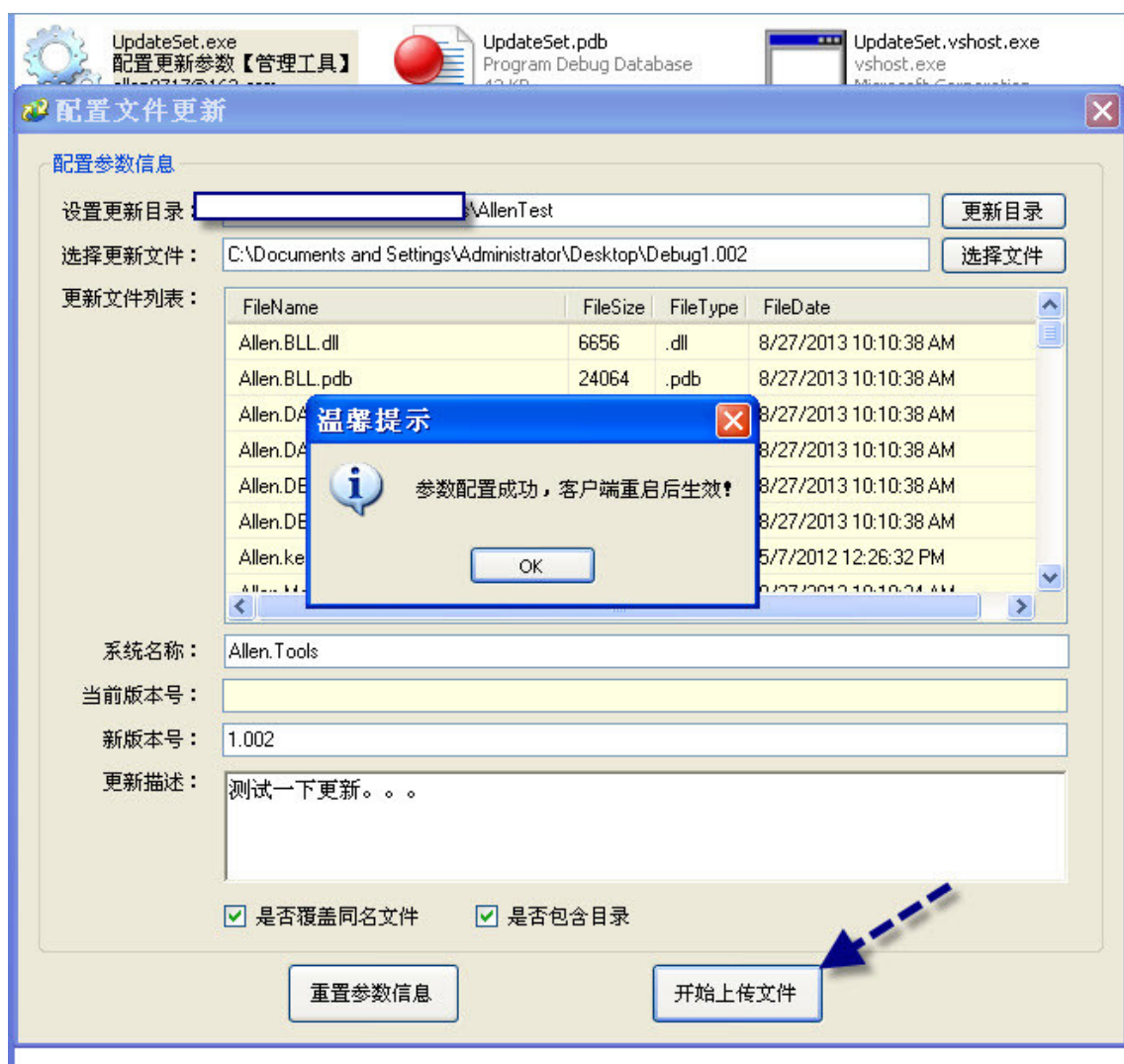
更新文件：最新版本的文件

系统名称：这个主程序的名称

当前版本：根据名称自动显示

新版本号：执行填写

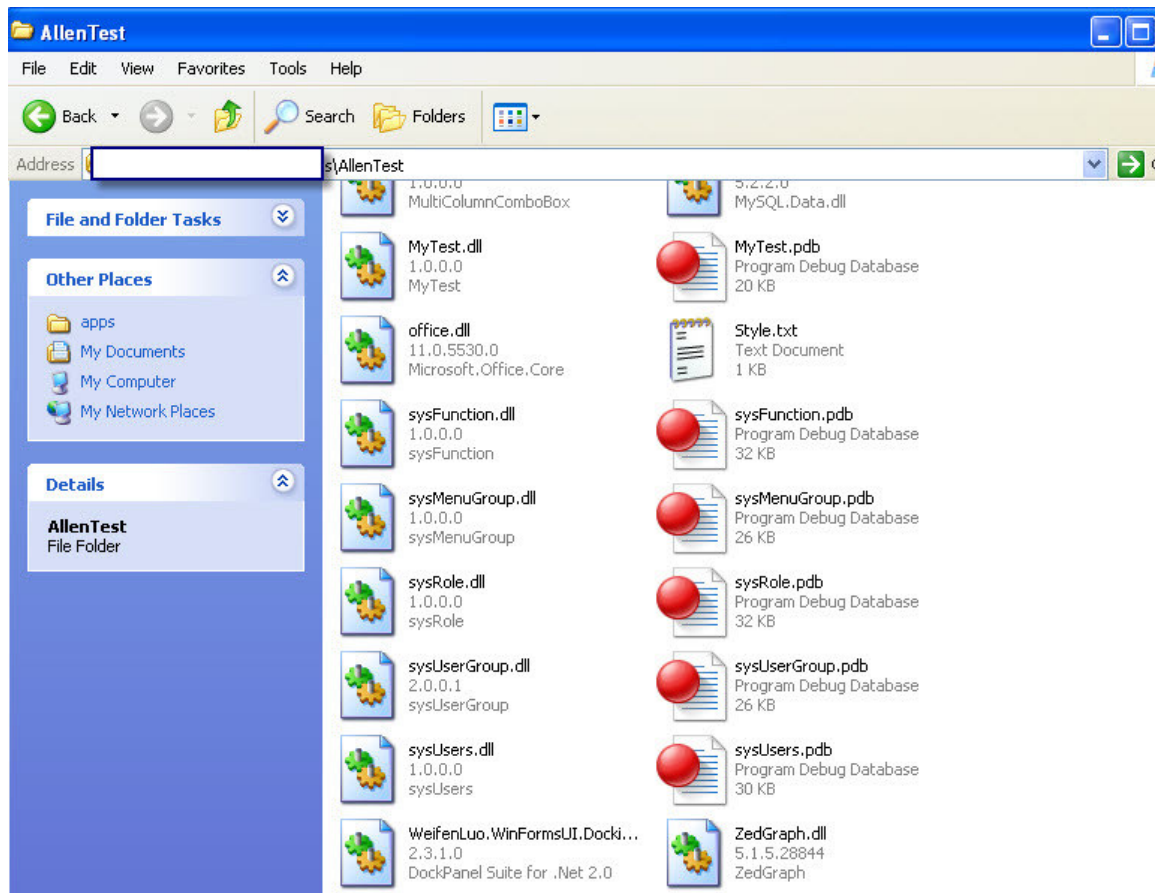
更新描述：显示给客户端用户查看更新的内容



(图 25)

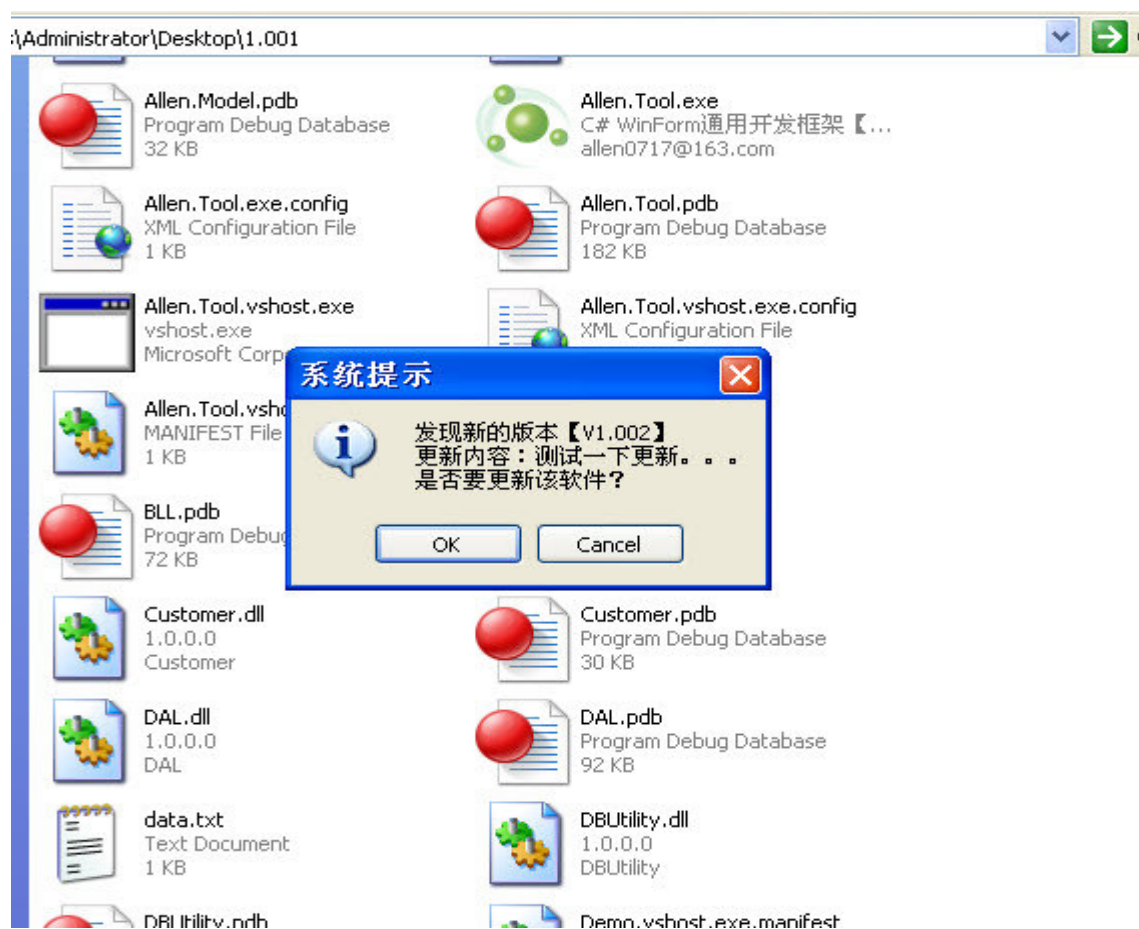
这个配置界面是管理员在后台操作的，前台用户根本看不到这个界面

我们发现文件服务器上面已经有最新的文件了，在上传文件之前，作者已经清空了该目录下所有的文件。



(图 26)

为了达到演示效果，作者编译了一份版本为 1.001 的程序放在本地，然后另外编译了一份版本为 1.002 的程序放在文件服务器上面，现在我们在客户端运行版本为 1.001 的程序，看看自动更新的效果：



(图 27)

程序启动的时候检测到有新版本程序，提示用户是否更新，如果用户选择不更新，那么可以继续使用旧版本的程序，我们选择“是”



(图 28)



(图 29)

更新完成之后我们看看客户端的是程序是不是最新版本 1.002 的，如下图所示，说明我们的自动更新成功了。



(图 30)

如果是从文件服务器上面来进行文件更新的话，需要注意服务器上面是不是允许客户端从服务器上面 Copy 文件，如果需要指定用户名和密码来获取，那么请通过以下方式：

首先声明如下变量：

```
[DllImport("mpr.dll", EntryPoint = "WNetAddConnection2")]
public static extern uint WNetAddConnection2(
    [In] NETRESOURCE lpNetResource,
    string lpPassword,
    string lpUsername,

    uint dwFlags);
[DllImport("Mpr.dll")]
public static extern uint WNetCancelConnection2(string lpName, uint
dwFlags, bool fForce);
[StructLayout(LayoutKind.Sequential)]
public class NETRESOURCE
{
    public int dwScope;
    public int dwType;
    public int dwDisplayType;
    public int dwUsage;
    public string LocalName;
    public string RemoteName;
    public string Comment;
    public string Provider;
}
```

接着用下面的代码来通过密码访问服务器：

```
NETRESOURCE myNetResource = new NETRESOURCE();
myNetResource.dwScope = 2;
myNetResource.dwType = 1;
myNetResource.dwDisplayType = 3;
myNetResource.dwUsage = 1;
myNetResource.LocalName = "T:";
//myNetResource.RemoteName = @"\\ServerName\\apps\\Update.exe";
myNetResource.Provider = null;
uint nret = WNetAddConnection2(myNetResource, "LoginID", "Password", 0);
System.Threading.Thread.Sleep(1000);
//这里就可以直接调用文件Copy的方法了。
//System.IO.File.Copy("", "", true)
```

小技巧：如何修改工具箱的背景颜色以及文字颜色

//工具箱的背景颜色

```
_outlookBar.ViewStyle.BarClientAreaColor = Color.White;
```

//鼠标悬浮在菜单上面的时候菜单的文字前景色

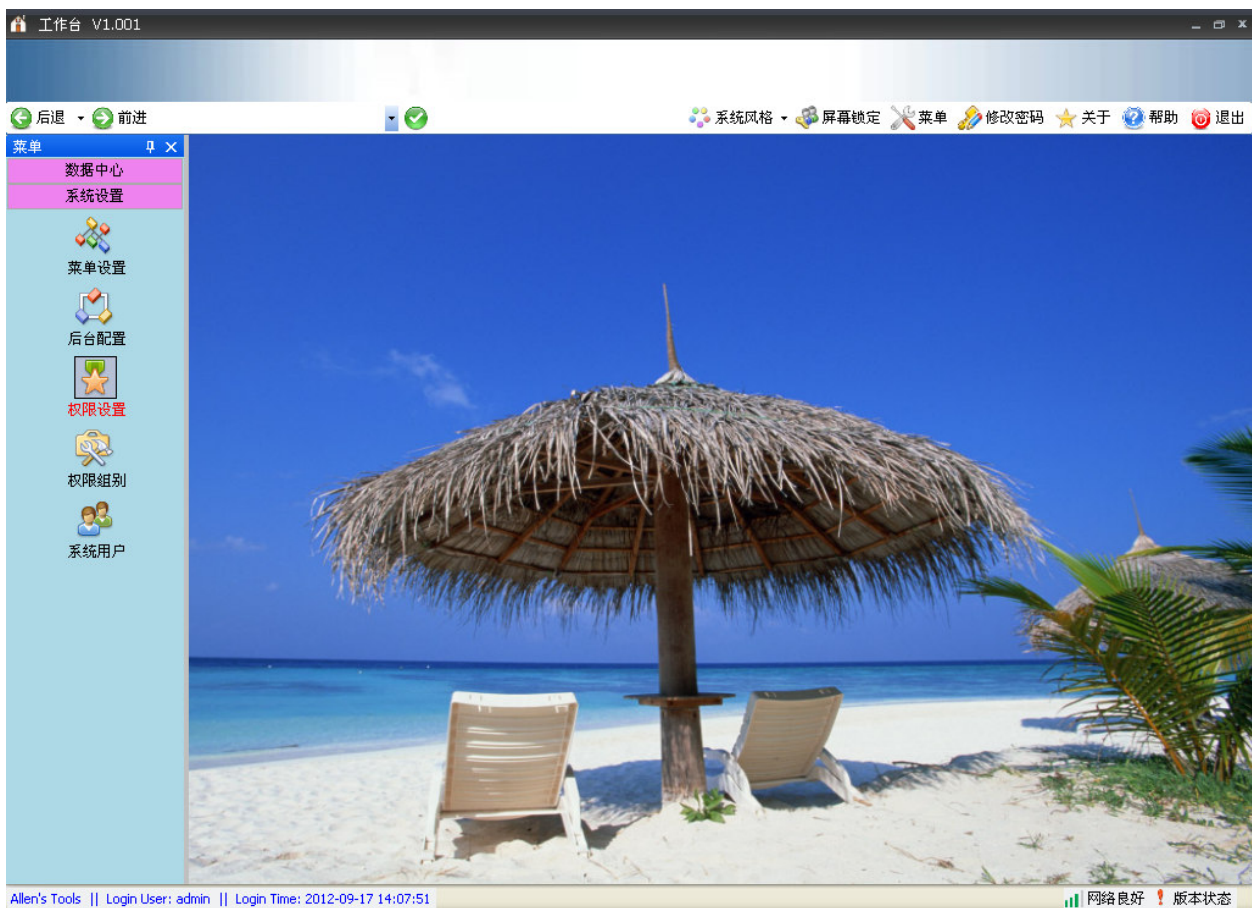
```
_outlookBar.ViewStyle.BarItemHotTextColor = Color.Blue;
```

//菜单的文字前景色

```
_outlookBar.ViewStyle.BarItemTextColor = Color.Black;
```

//主菜单名称的背景颜色

```
_outlookBar.ViewStyle.BarColor = Color.BurlyWood;
```



(图 31)

图 (18) 左侧的工具箱跟其它的图片略有不同，这里涉及到 4 种颜色，

- A. 工具箱文字的颜色：BarItemTextColor
- B. 工具箱主菜单的背景颜色，即每一个菜单组的背景颜色：BarColor
- C. 工具箱的背景颜色：BarClientAreaColor
- D. 鼠标悬浮在工具箱菜单上面的时候文字的颜色：BarItemHotTextColor

下面介绍一下该框架里面的注册功能

根据客户端 PC 的 CPU 编号和硬盘编号来生成一个机器码，然后通过这个机器码来获得一个对应的注册码，只要使用这个注册码去注册即可，注册的信息和试用的次数都是写在注册表信息里面；如果是试用版，则每次打开一次程序便将使用次数加 1，一直加到最上限 (这个可以随意去设置)，当使用次数达到最上限的时候就不能继续使用，只能注册后方可使用，

以下便是生成注册码的算法：

```
public string GetRegNum(string MachineCode)
{
    SetIntCode();
    string strMNum = MachineCode;
    for (int i = 1; i < charCode.Length; i++)
    {
        charCode[i] = Convert.ToChar(strMNum.Substring(i - 1, 1));
    }
    for (int j = 1; j < intNumber.Length; j++)
    {
        intNumber[j] = Convert.ToInt32(charCode[j]) +
intCode[Convert.ToInt32(charCode[j])];
    }
    string strAsciiName = "";
    for (int k = 1; k < intNumber.Length; k++)
    {
        if ((intNumber[k] >= 48 && intNumber[k] <= 57) || (intNumber[k]
>= 65 && intNumber[k]
<= 90) || (intNumber[k] >= 97 && intNumber[k] <= 122))
        {
            strAsciiName += Convert.ToChar(intNumber[k]).ToString();
        }
        else if (intNumber[k] > 122)
        {
            strAsciiName += Convert.ToChar(intNumber[k] -
10).ToString();
        }
        else
        {
            strAsciiName += Convert.ToChar(intNumber[k] -
9).ToString();
        }
    }
}
```



```
return strAsciiName;  
}
```

下面来看看图片效果，有图有真相：



图 (32)

程序检测到还未注册的时候会将“注册软件”的按钮显示出来，



图 (33)



图 (34)

图(34)显示的机器码为程序根据本地硬件自动生成的机器码。



图(35)

“生成注册码”这个界面是开发人员(或者说是软件出售者)自己使用的,通过客户端的机器码生成出对应的注册码,将生成的注册码发送给客户端,客户端直接注册便可。

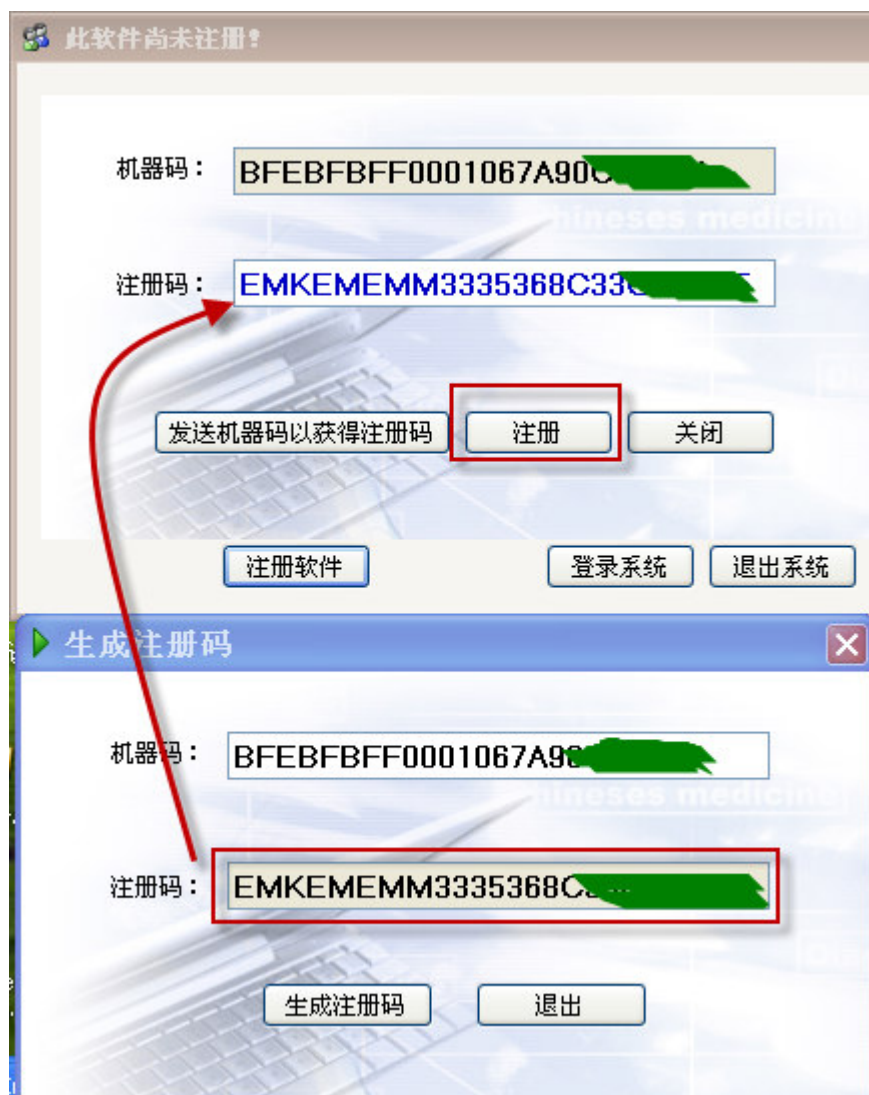


图 (36)



图 (37)

客户端通过得到的注册码直接注册便可成功注册，注册后每次登录便不会弹出提示信息。

总结：这个注册的功能对于开发人员来说并不是十分严谨，因为你有源代码，可以看得到这个注册的动作到底做了一些什么事情，就算是修改了注册表，你也可以通过源代码找到修改的地方，将其彻底处理掉，那么就相当是一个破解了；但是对于用户来说，用户并没有源代码也不知道到底是哪一些注册表文件在起作用，所以还是有点效果的。

下面介绍一下 SQL 数据库相关的内容

整个框架中所有与后台数据库相关的操作都是使用的存储过程，因为存储过程可以处理一些比较复杂的逻辑关系，另外存储过程直接在服务器上面执行，效率上面会有很大的优势，另外一个主要的原因是日后方便维护，如果全部将 SQL 语句写在程序里面，后台数据库有一点点更改的时候岂不是要重新修改程序里面 SQL 语句并重新打包才可以？这样的话是一个很庞大的工作量，并且还不一定能够很好的去维护；如果是用存储过程，那么只需要在服务器上面将存储过程稍微修改一下，然后执行一下并保存，客户端立刻就生效了，是不是很方便？

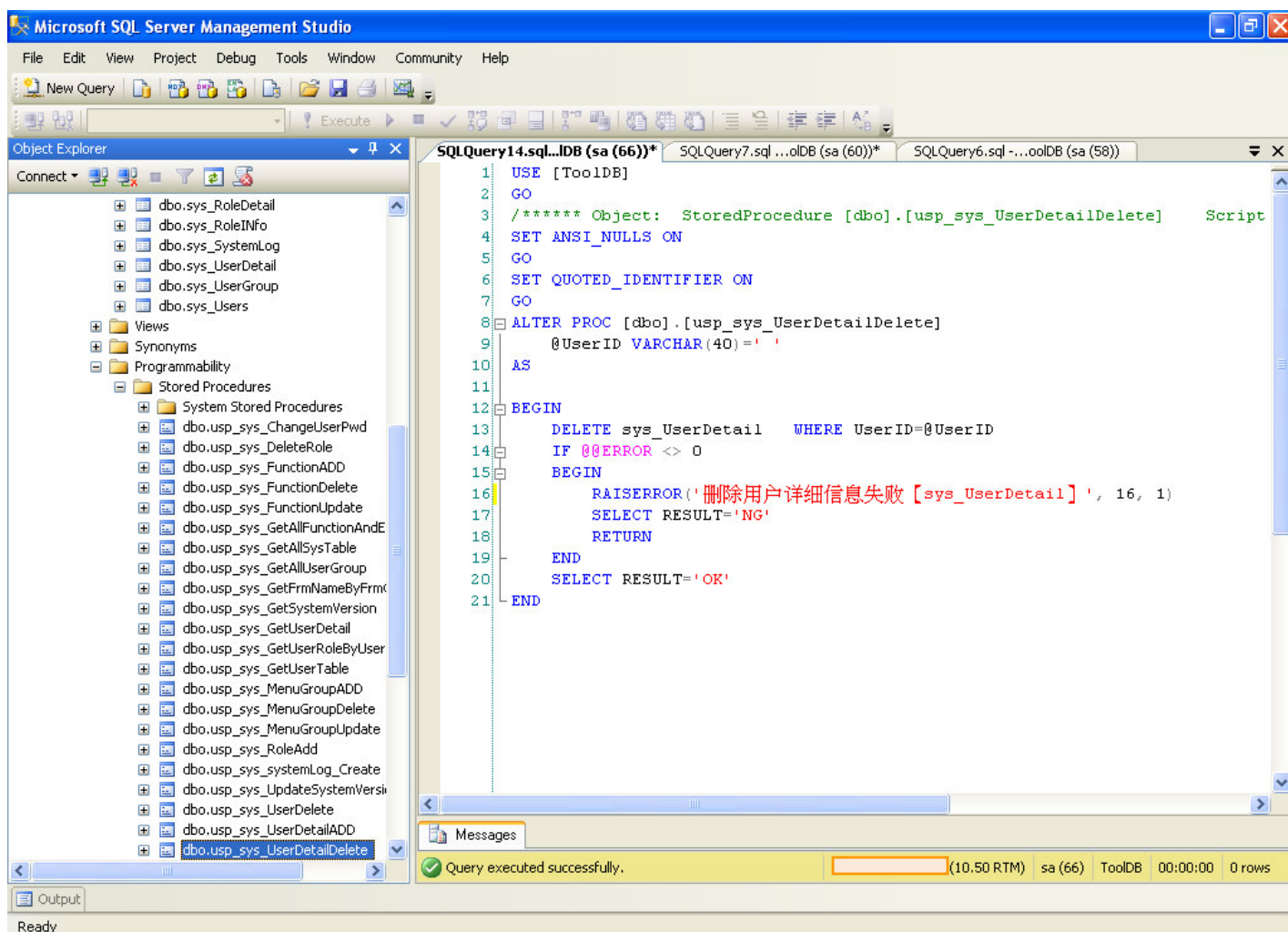
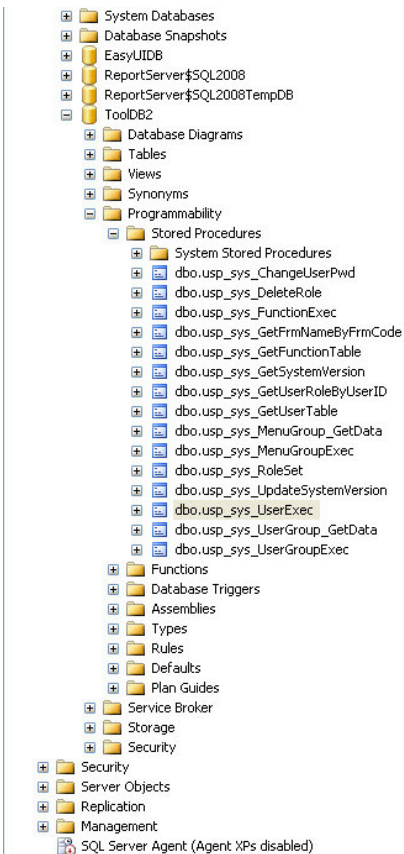


图 (38)



```

SET ANSI_NULLS ON
GO
SET QUOTED_IDENTIFIER ON
GO
ALTER PROC [dbo].[usp_sys_UserExec]
    @Type VARCHAR(1) = '0',
    @UserID varchar(20) = ' ',
    @UserName varchar(20) = ' ',
    @GroupID varchar(50) = ' ',
    @UserPassword varchar(20) = ' ',
    @UserPhone varchar(30) = ' ',
    @IsEnabled varchar(2) = ' ',
    @Remark varchar(30) = ' '
AS
BEGIN TRAN
IF @Type='0' --增加
BEGIN
    IF EXISTS(SELECT * FROM sys_Users WHERE UserID=@UserID)
    BEGIN
        SELECT FERRDESC='NG,用户(' + @UserID + ') 已经存在，不可重复[sys_UserInfo]'
        ROLLBACK TRAN
        RETURN
    END
    INSERT INTO sys_Users (UserID,UserName,GroupID,UserPassword,UserPhone,IsEnabled,Remark,C
VALUES (@UserID,@UserName,@GroupID,@UserPassword,@UserPhone,@IsEnabled,@Remark,GETDATE()
    IF @@ERROR <> 0
    BEGIN
        SELECT FERRDESC='NG,用户(' + @UserID + ') 创建失败[sys_UserInfo]'
        ROLLBACK TRAN
        RETURN
    END
    SELECT FERRDESC='OK,(' + @UserID + ') 创建成功.'
END
IF @Type='1' --删除
BEGIN
    IF NOT EXISTS(SELECT * FROM sys_Users WHERE UserID=@UserID)
    BEGIN
        SELECT FERRDESC='NG,用户(' + @UserID + ') 不存在[sys_UserInfo]'
        ROLLBACK TRAN
        RETURN
    END

```

(图 39)

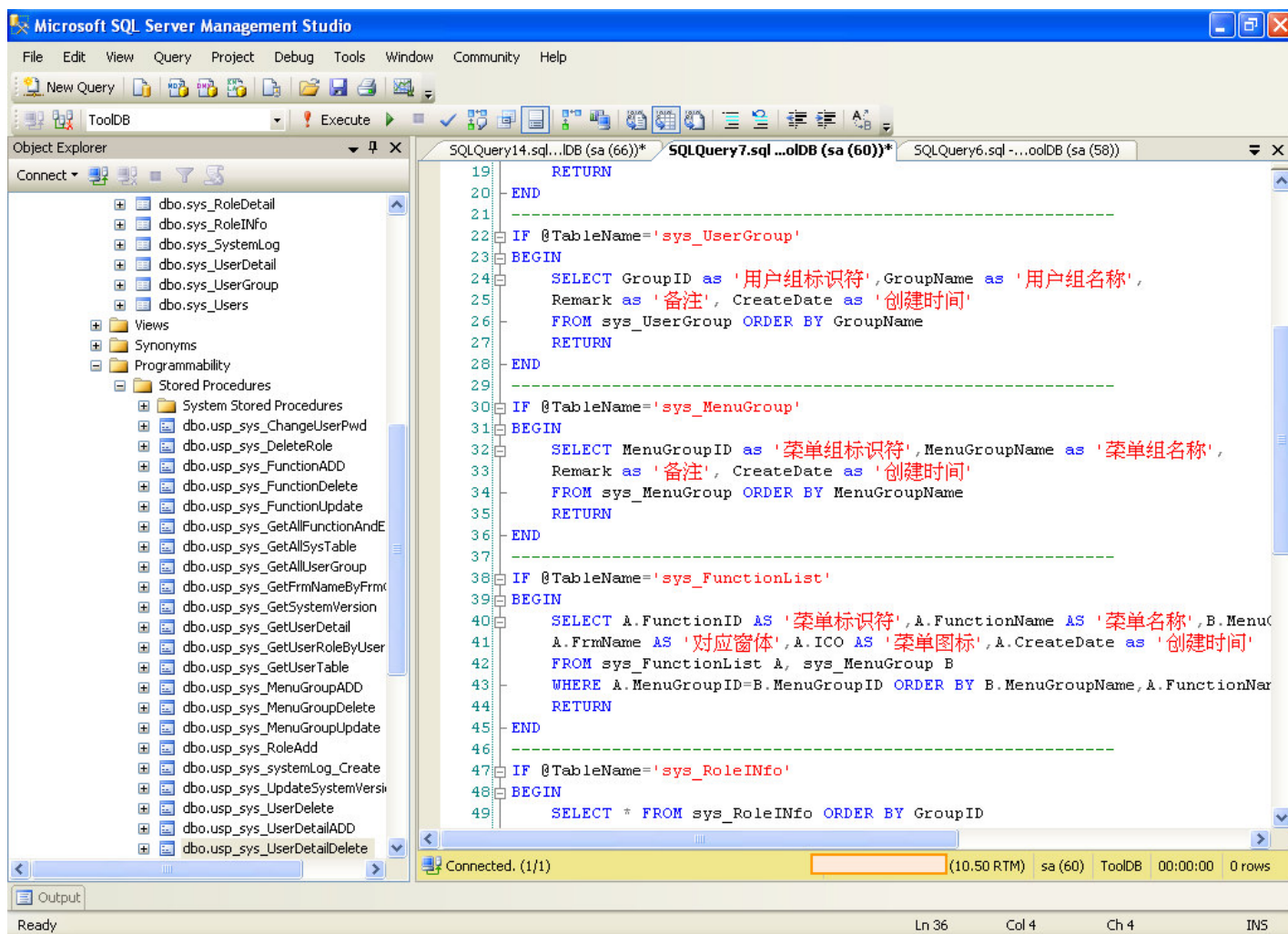


图 (40)

最后附上一张作者使用该框架开发的系统的截图，截至作者编写这份开发文档的时候，作者利用这套框架开发的这个程序以及运行 1 年时间了，用户数最多的时候有 200 个左右，也就是说服务器上面的连接数有 200，因为数据库安装在服务器上面，所有客户端都是访问同一个数据库。

下图中公司 Log 作者已经去掉了，图片表示的是生产线上锡炉的动态参数看板，实时刷新数据，当时国外的客人看了这个电子系统之后给予的很好的评价。

CMS.MES V1.062

后退 前进


系统风格 屏幕锁定 菜单 修改密码 关于 帮助 退出

锡炉看板

Wave Solder Process Visual Aid

Rev # 00 Rev Date 02/29/2012 0:00:00 启用定时刷新 Print

WS Palette Information



WS Palette # 50-9481-30T01

TO WAVE

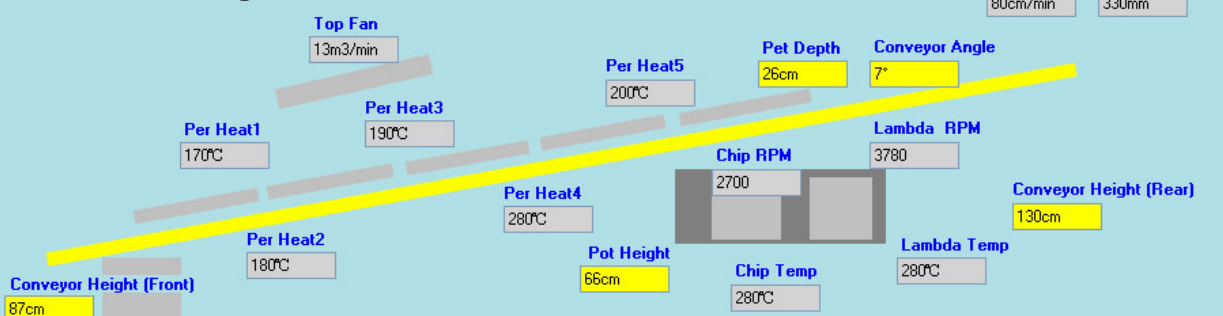
PCBA Model WS Line

PCBA P/N Rev 50-9481-03T01

WS Line # 3F1L

WS Program Filename 50-9481-30T01

WS Machine Setting



Conveyor Speed 80cm/min Conveyor Width 330mm

Top Fan 13m3/min

Per Heat1 170°C Per Heat2 180°C Per Heat3 190°C Per Heat4 280°C Per Heat5 200°C

Pot Height 66cm

Pet Depth 26cm

Conveyor Angle 7°

Chip RPM 2700

Lambda RPM 3780

Conveyor Height (Front) 87cm

Conveyor Height (Rear) 130cm

Chip Temp 280°C

Lambda Temp 280°C

Flux Spray Setting

Flux Pressure 5mpa	Flow Rate 47%	Nozzle Height 5cm	Flux Delay 100mm/sec
Flux Speed 300mm/sec	Spray Width 310mm	Flux Part Number 50-8051-00N00	Flux Pull-In 100mm/sec

Weight Fixture

1. N/A	5. N/A
2. N/A	6. N/A
3. N/A	7. N/A
4. N/A	8. N/A

登录ID: Allen 【allenchen】 登录时间: 2012-09-17 09:23:05

网络良好 版本状态

(图 41)

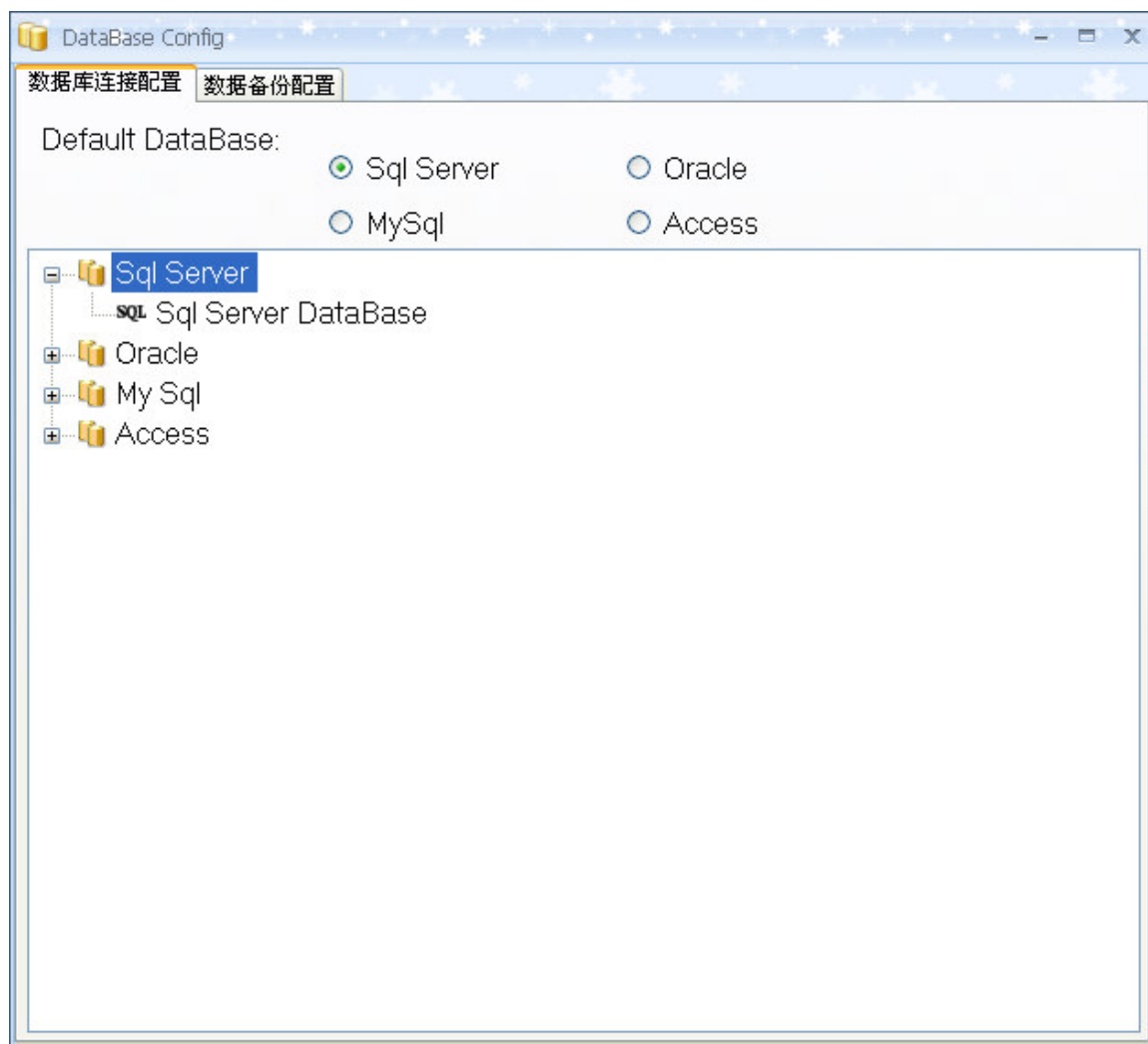
下面为部分界面截图



(图 42)



(图 43)



(图 43)

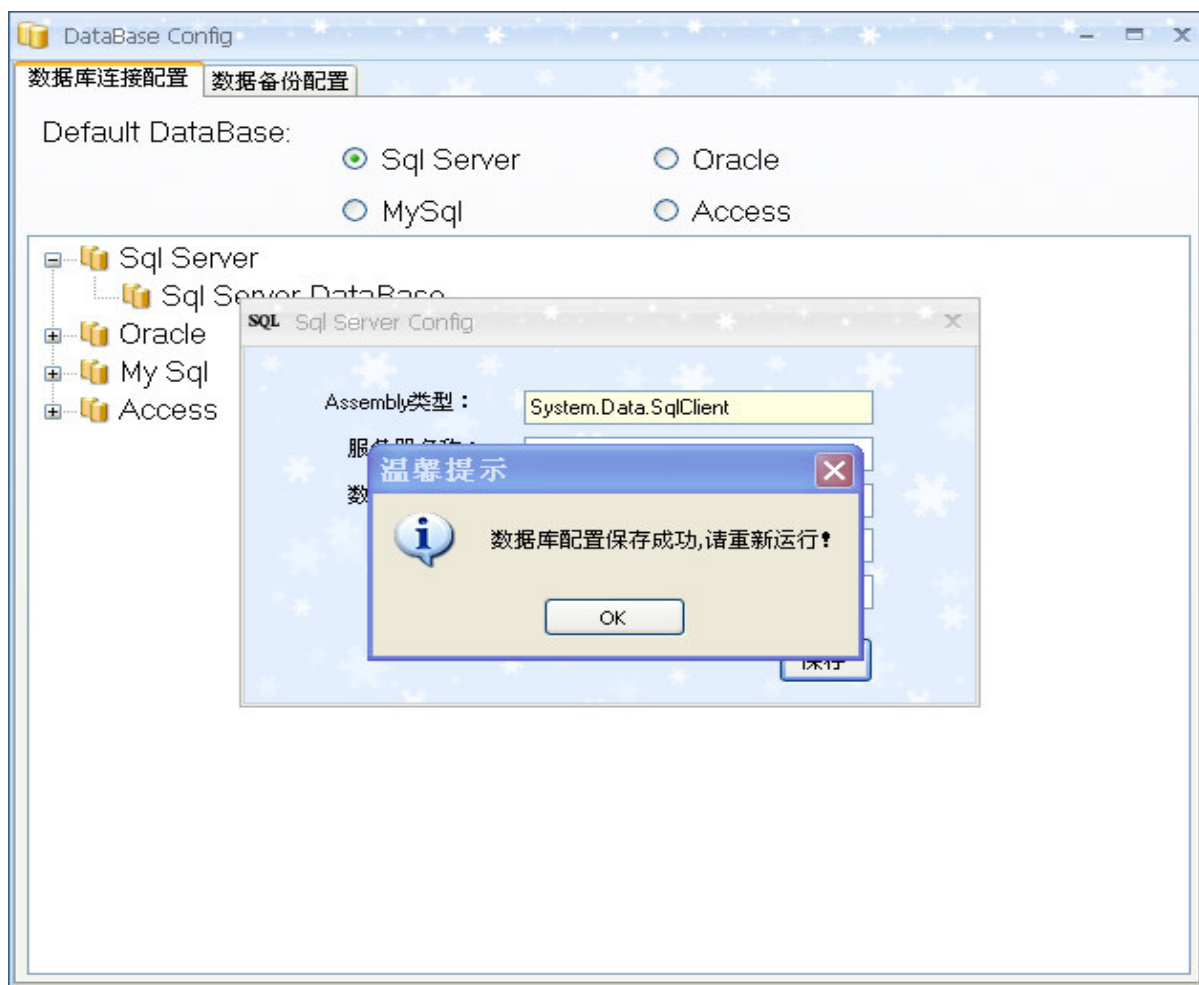


图 (44)



图 (45)

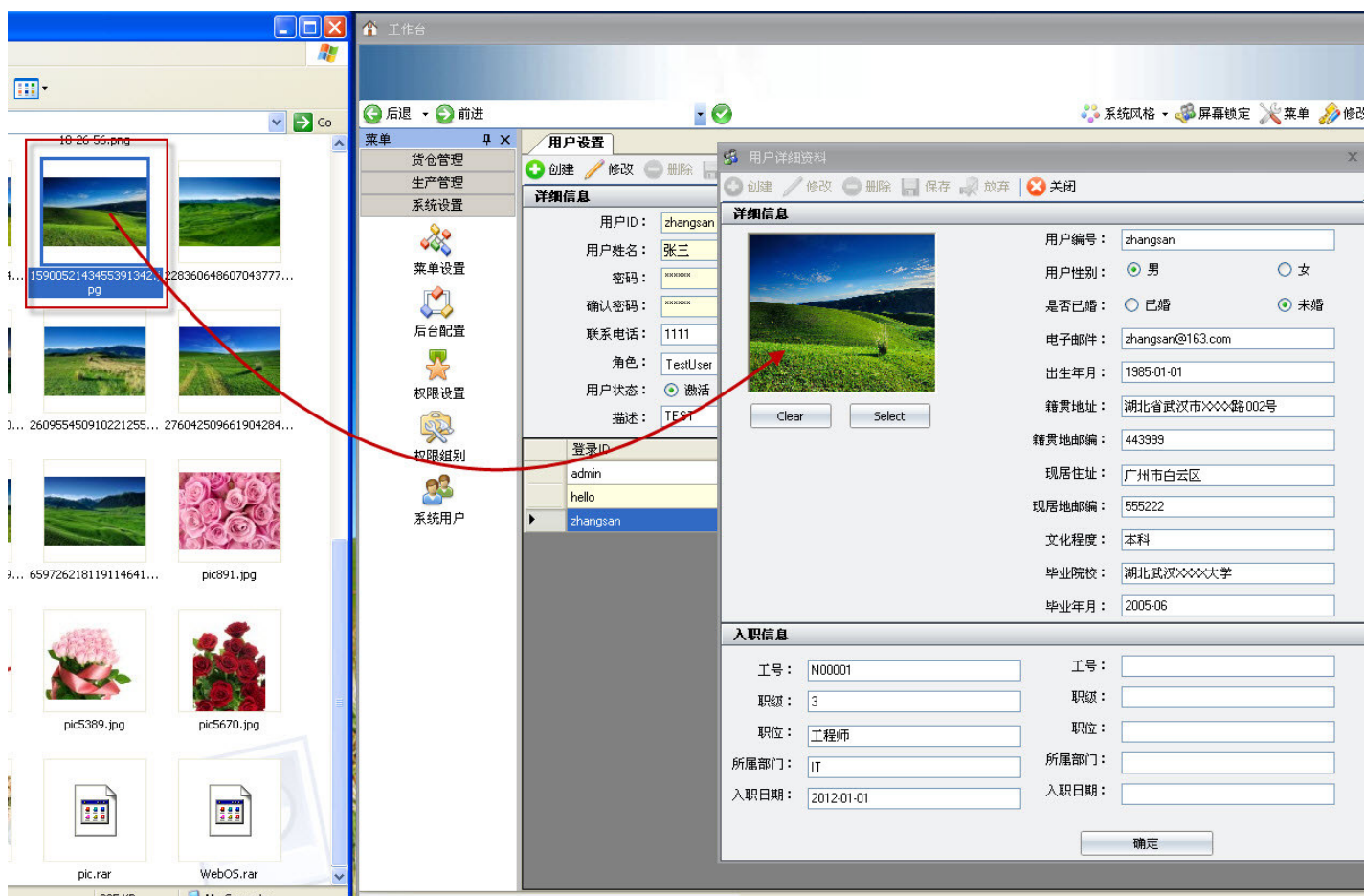


图 (46)

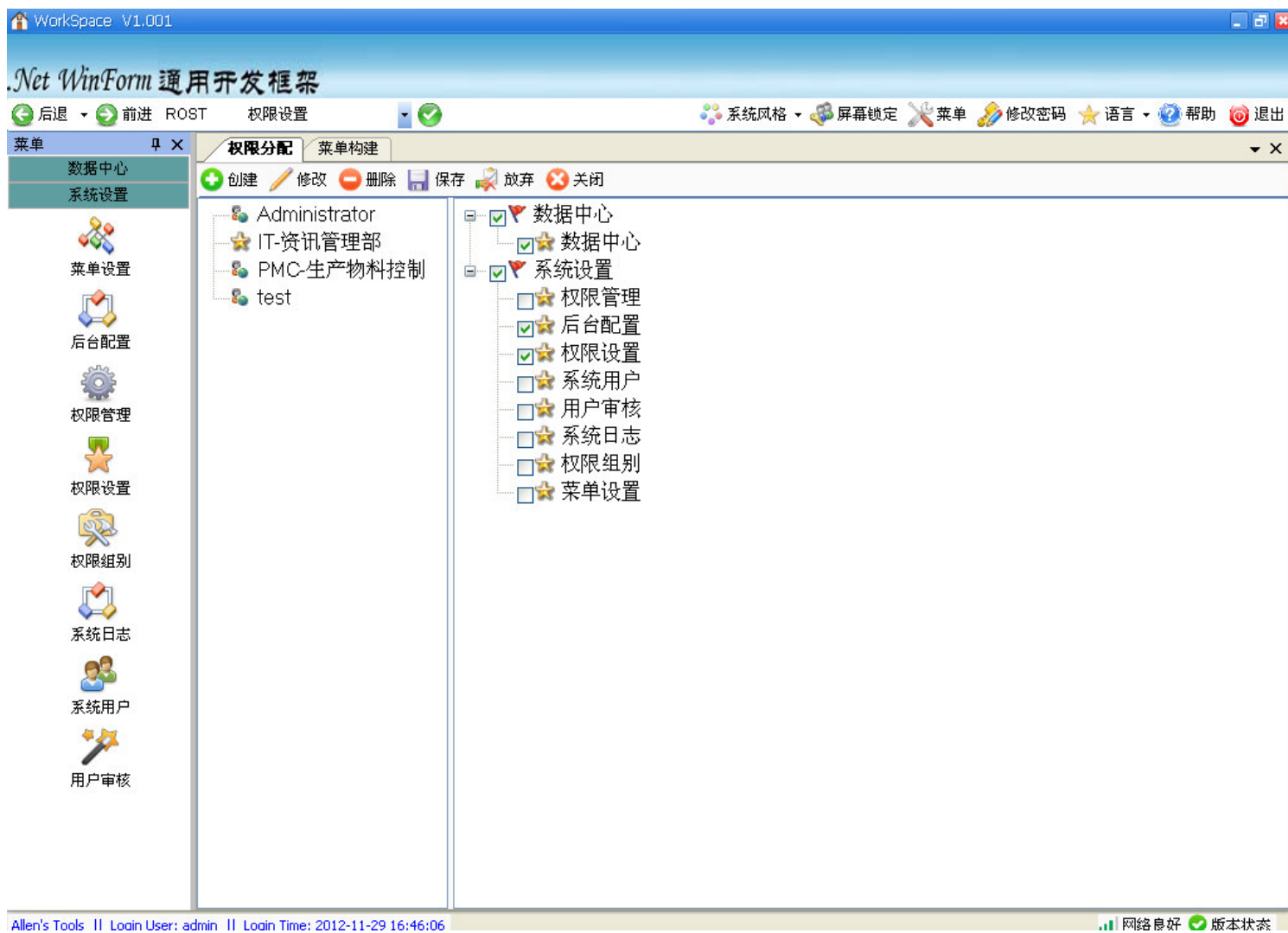


图 (47)

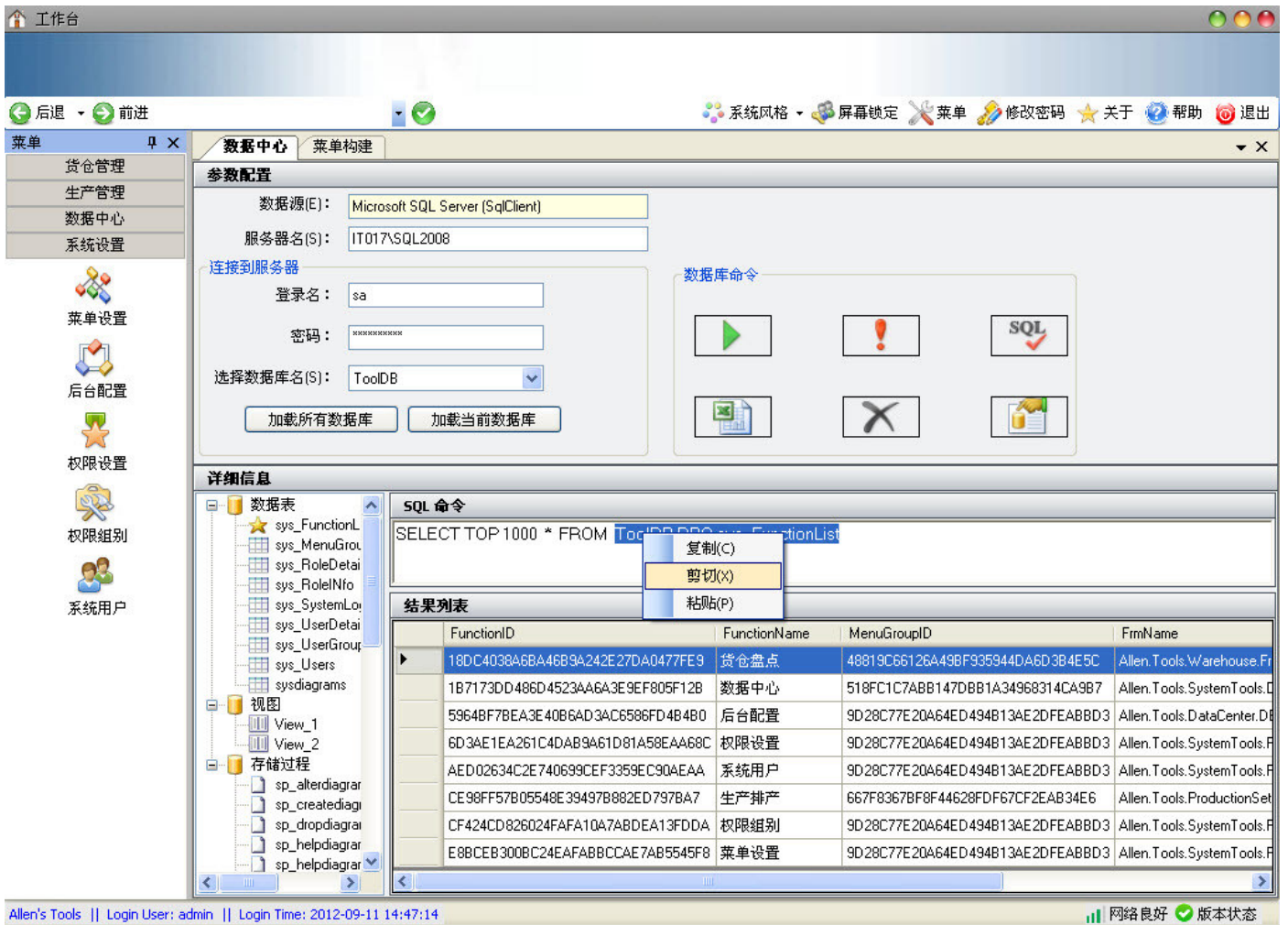


图 (48)

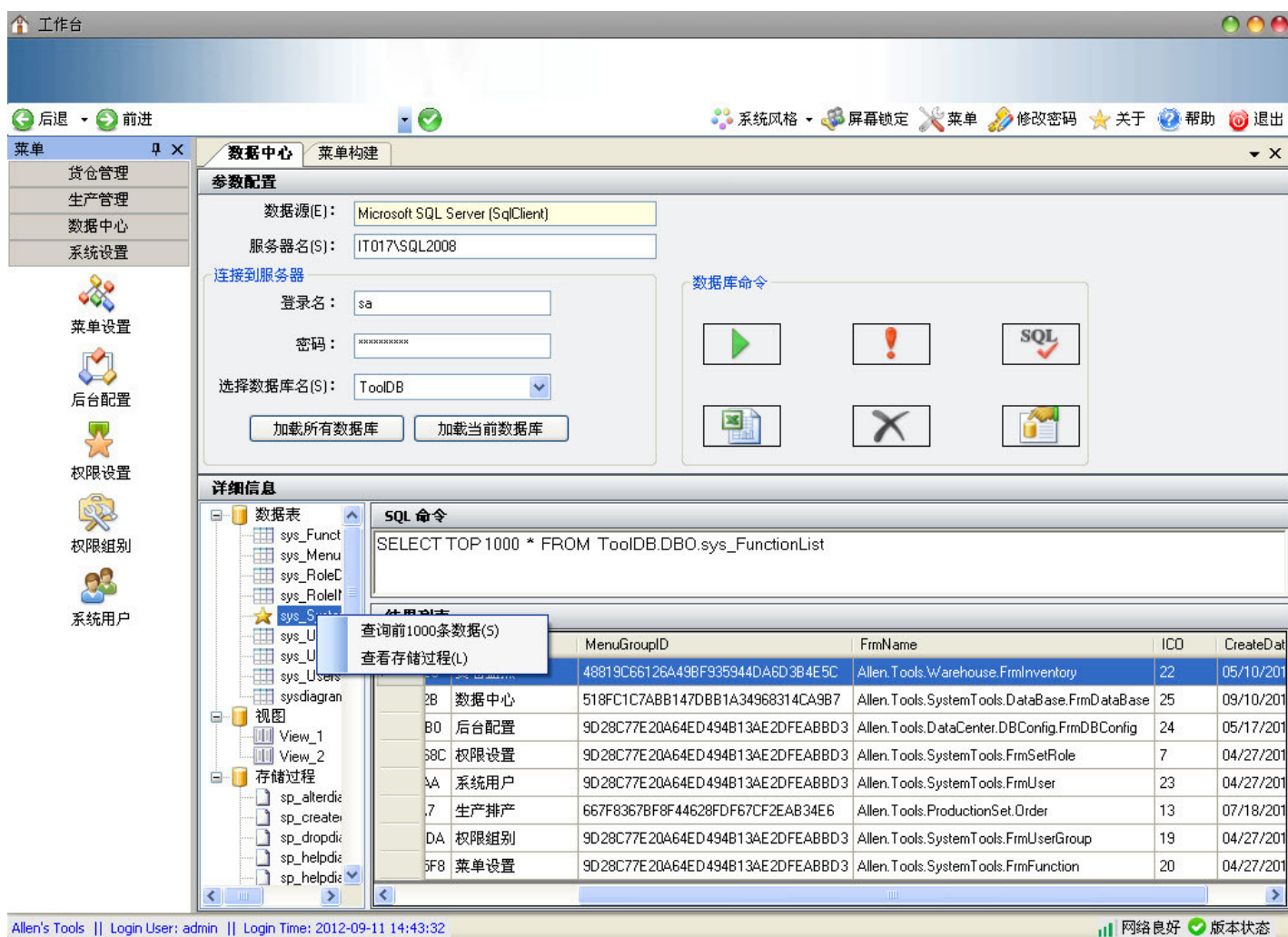


图 (49)

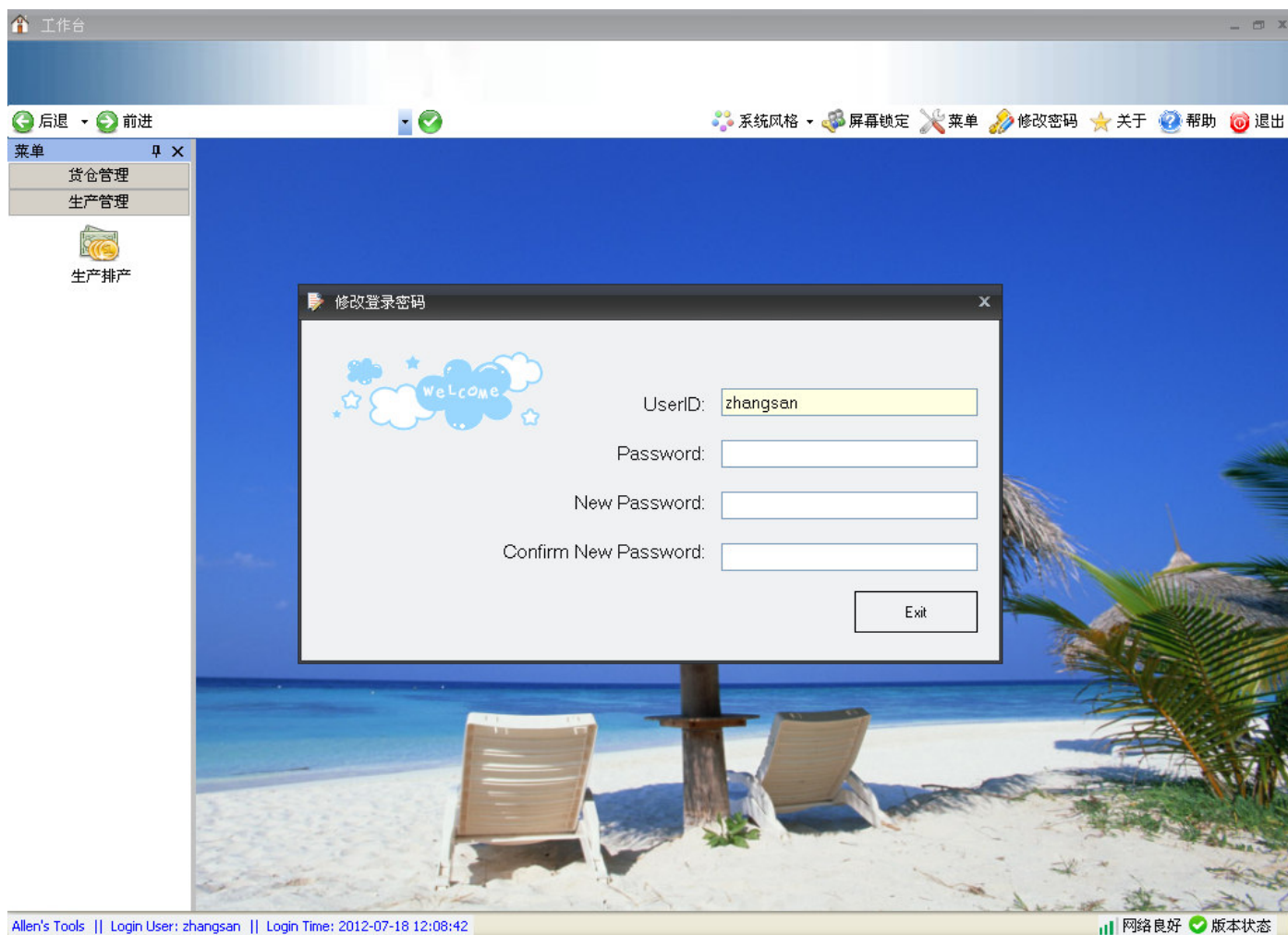


图 (50)

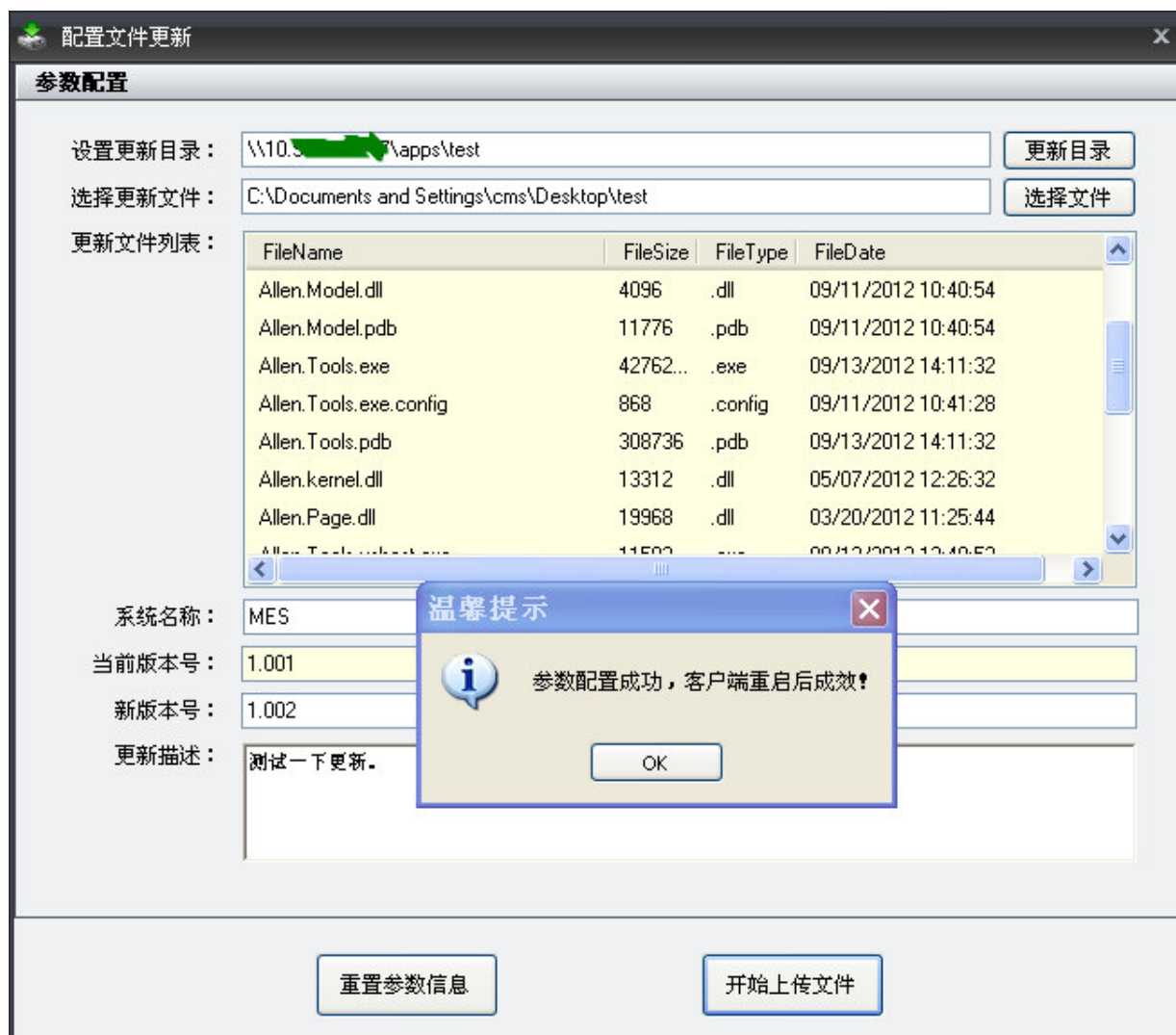


图 (51)

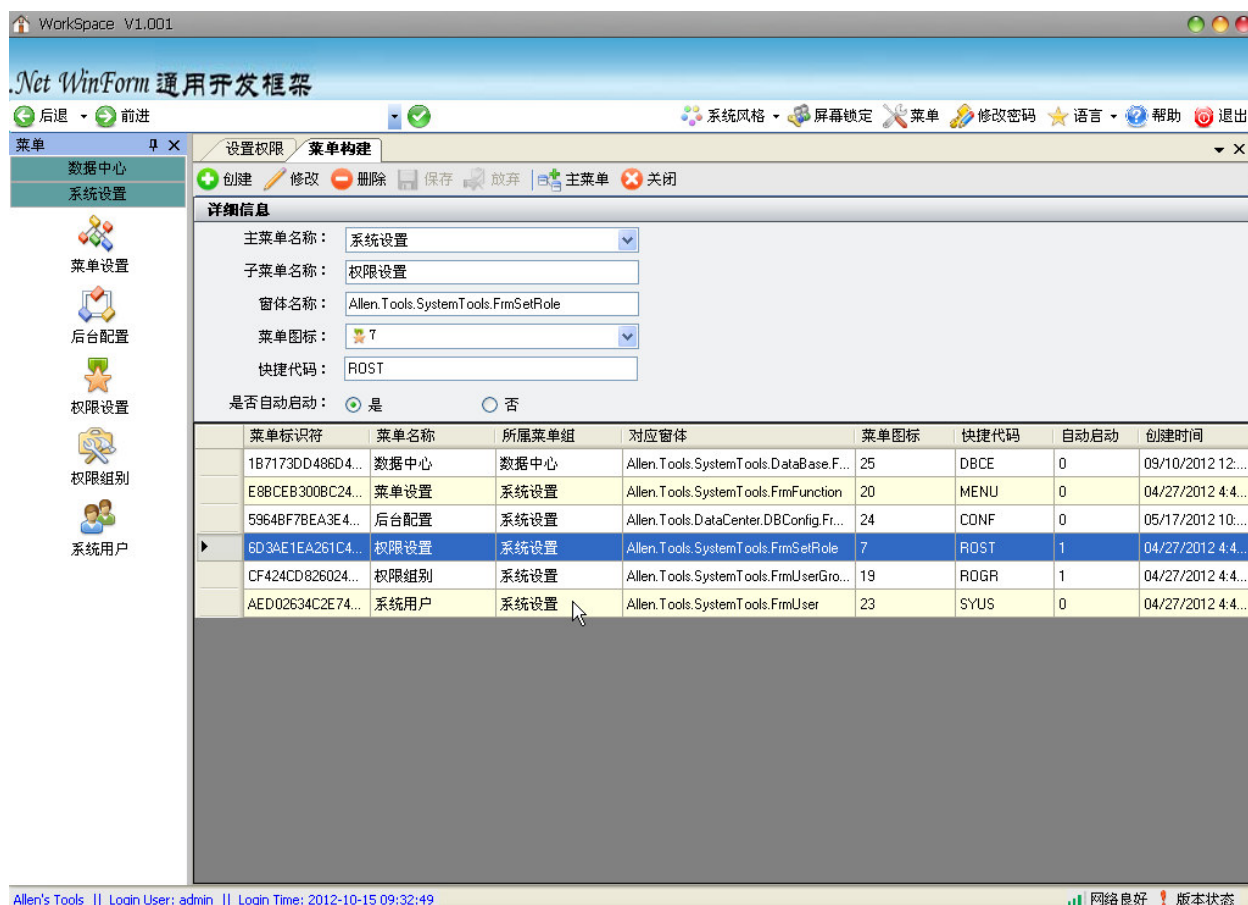


图 (52)

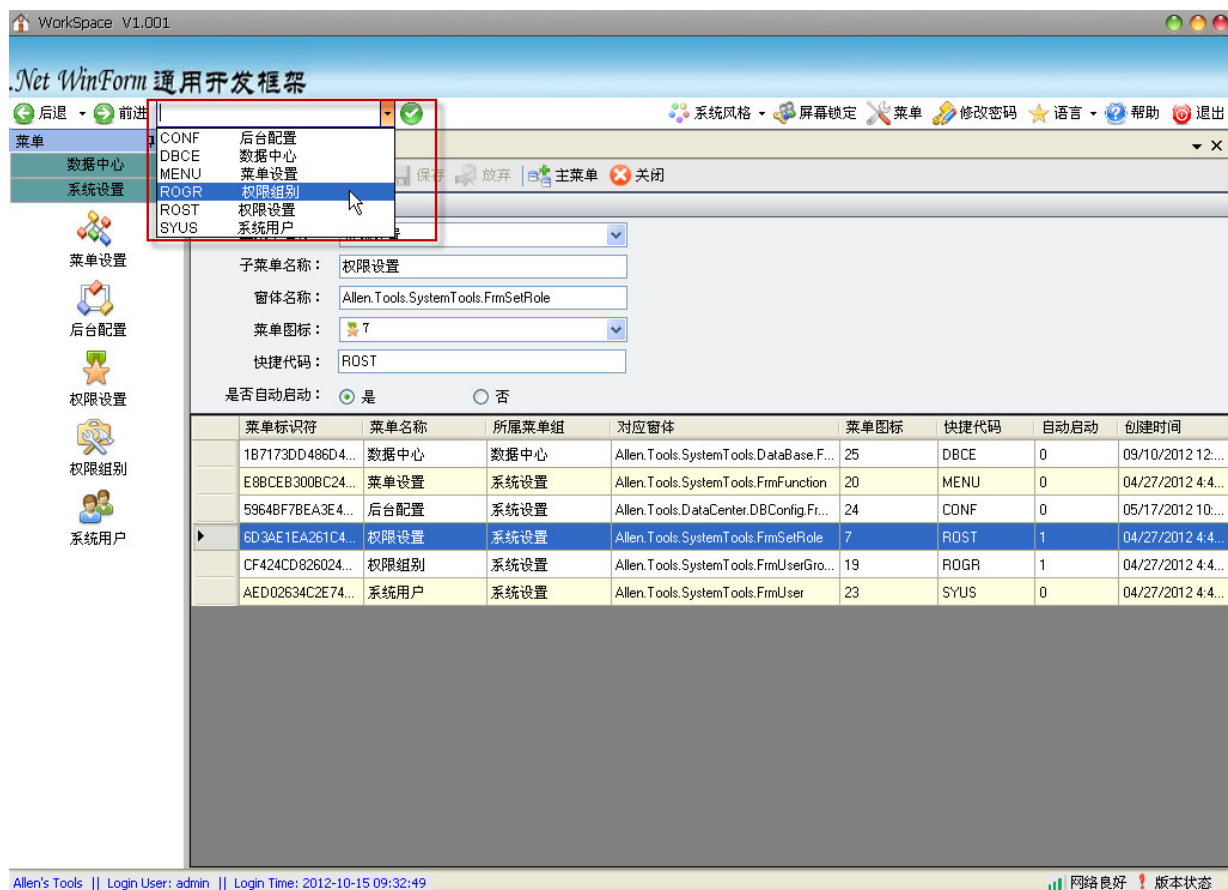


图 (53)

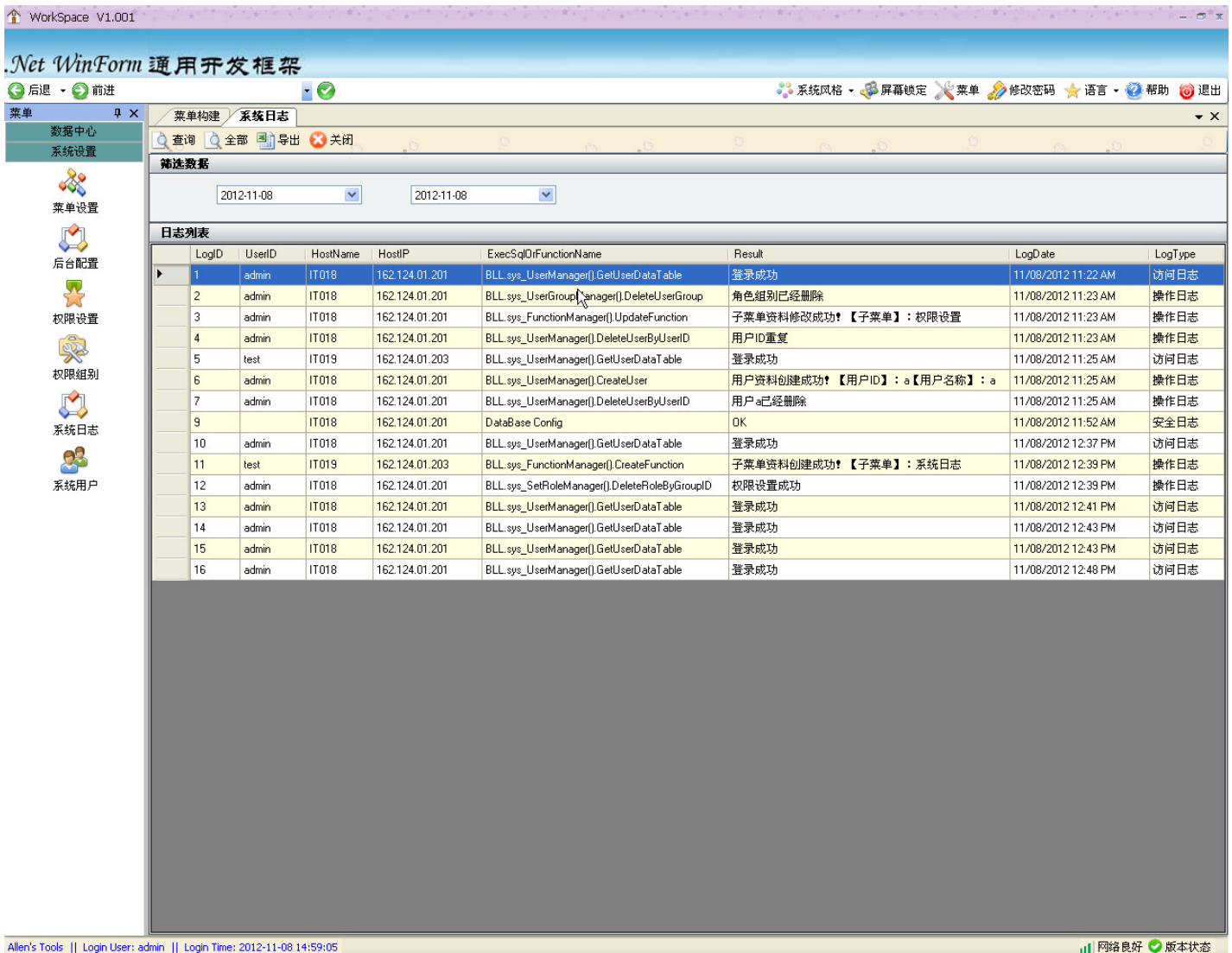
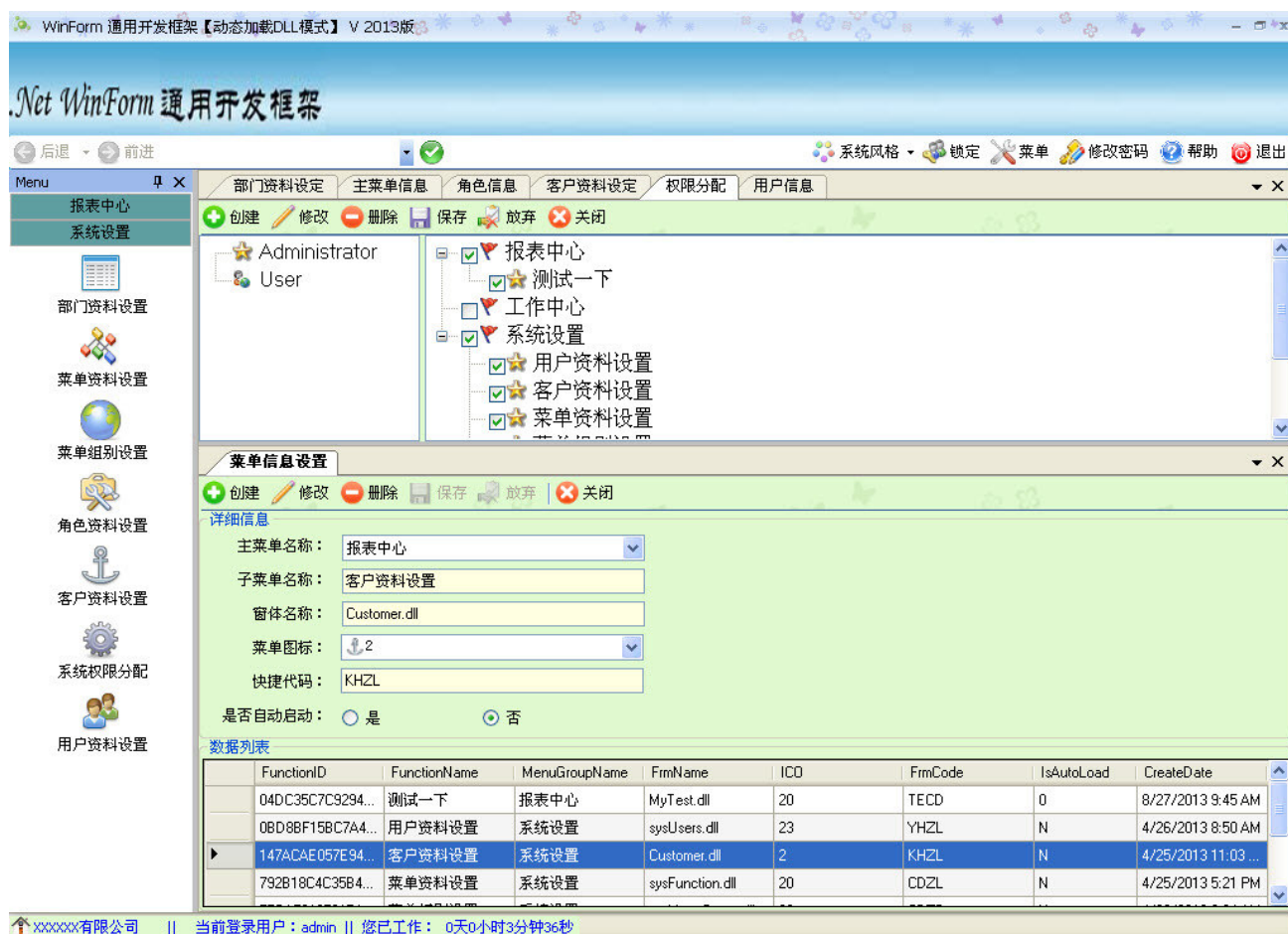


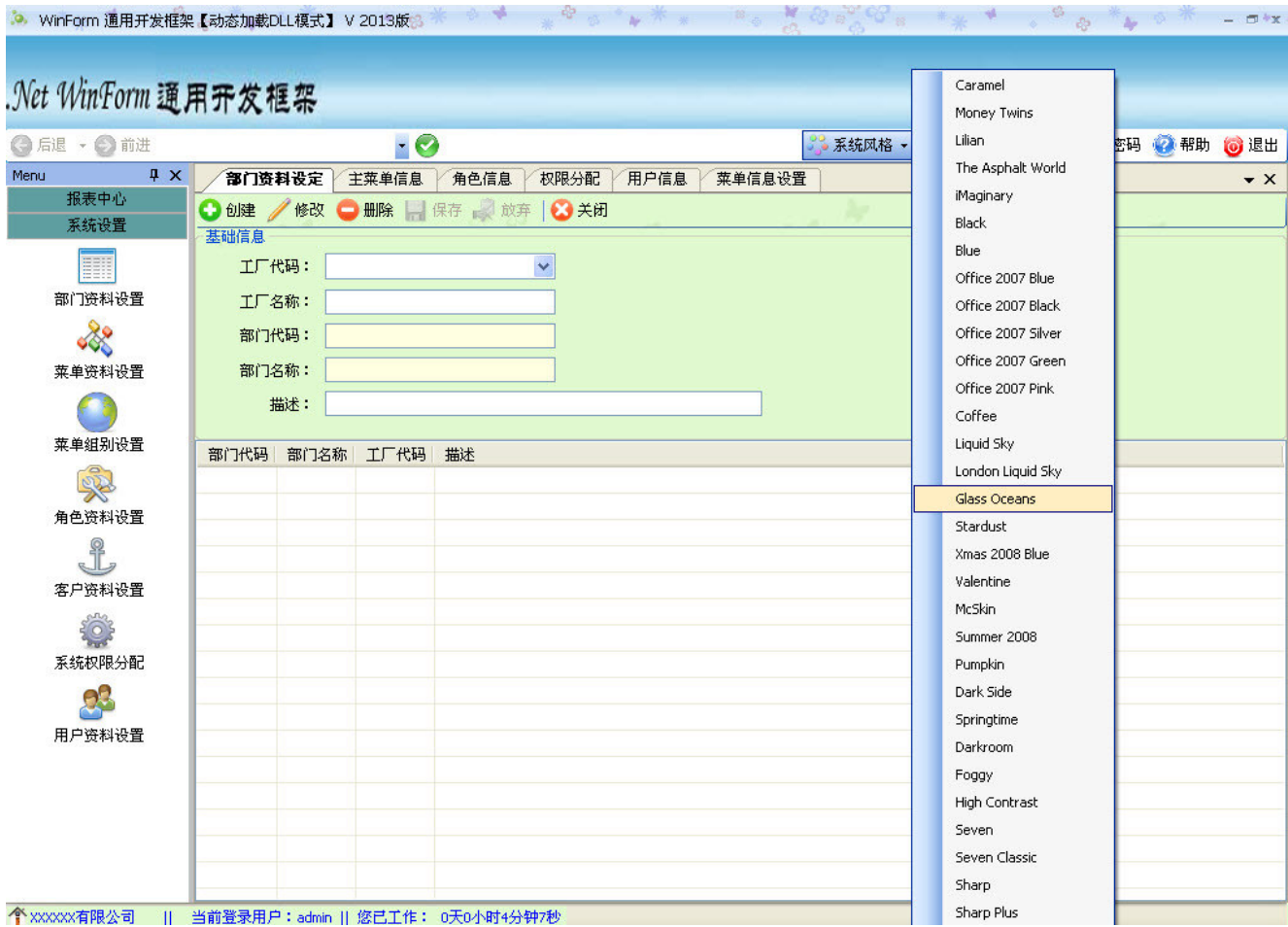
图 (54)



(图 55)



(图 56)



(图 57)



(图 58)

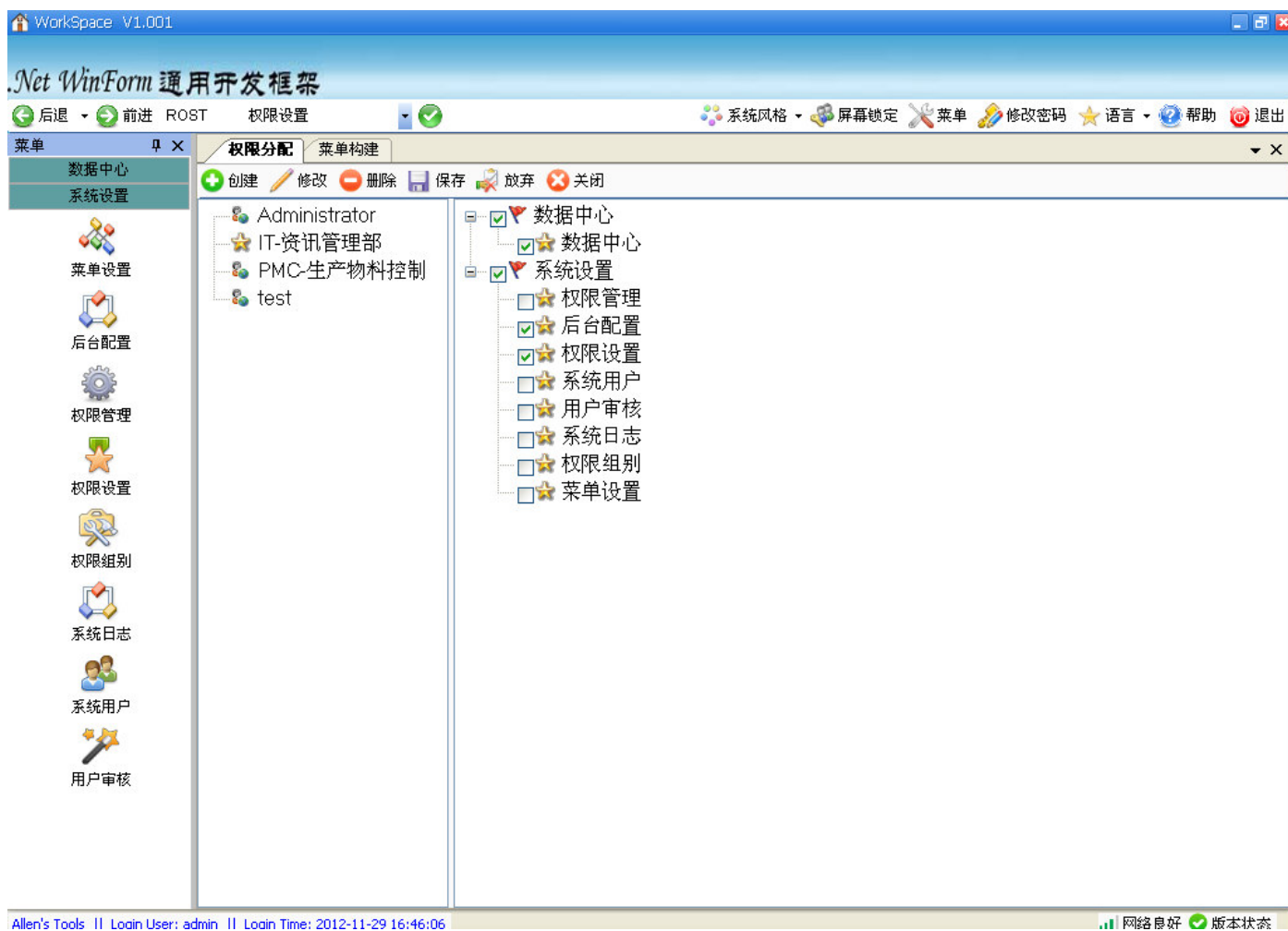


图 (59)

开发环境: Microsoft Windows XP Professional Service Pack 3

Visual Studio 2010 专业版

Microsoft .NET Framework 4.0 SP1

SQL Server 2008 R2 英文版

用户在使用过程中的一些问题点

Q1: 在 Release 环境下编译失败, 提示找不到 `Allen.DAL;Allen.BLL;Allen.Model` 等等?

A1: 作者为了调试方便, 都是在 DeBug 环境下开发的, 如果需要切换到 Release 环境, 请将一下文件复制到 Release 环境下的 Bin 目录里面:

```
Allen.BLL.dll
Allen.DAL.dll
Allen.kernel.dll
Allen.Model.dll
Allen.Page.dll (分页组件)
DevExpress.BonusSkins.v9.2.dll
DevExpress.Data.v9.2.dll
DevExpress.OfficeSkins.v9.2.dll
DevExpress.Utils.v9.2.dll
DevExpress.XtraEditors.v9.2.dll
ImageComboBox.dll
LumiSoft.UI.dll
Microsoft.Office.Interop.Excel.dll
Microsoft.Practices.EnterpriseLibrary.Common.dll (企业库组, 目前未使用)
Microsoft.Practices.EnterpriseLibrary.Data.dll (企业库组, 目前未使用)
Microsoft.Vbe.Interop.dll
MultiColumnComboBox.dll
office.dll
WeifenLuo.WinFormsUI.Docking.dll
```

Q2: 为什么每次“全部重新生成解决方案”之后需要重新配置后台连接?

A2: 因为连接是保存在 Setting 里面的, 每次“全部重新生成解决方案”的会将内存的数据清空掉, 所以要重新配置。

Q3: 每次运行程序总是提示要注册, 能不能去掉注册的检查?

A3: 检查是否注册的代码在 `FrmLogin.cs` 里面, 只需要将这段代码注释掉即可。

Q4: 数据库文件附加的时候出错?

A4: 系统建立的时候使用的数据库是 SQL Server 2008 R2 英文版, 如果你的 SQL 不是这个版本则不可以直接附加, 出现这种情况请将项目里面的 `DataBaseScript.SQL` 文件按照说明在你本机的 SQL 查询分析器里面执行一次, 这个文件是所有数据库的脚本, 目的就是防止数据库版本不一致导致无法附加数据库。

Q5: 开发的时候没有 DevExpress 9.2 的安装文件和破解文件?

A5: 作者会直接以邮件的方式将安装文件和破解文件一并发送, 客户端不需要安装这个软件, 只是为了开发的时候方便一些。

Q6: 修改了数据库名称之后, 后台配置成功了, 但是无法登录到系统?

A6: 开发的时候每一个存储过程都是按照“数据库名称.DBO.存储过程名称”的格式编写的, 所以如果数据库名称被改变了, 请将 Allen.DAL 里面的所有类文件查找一次, 将 ToolDB 替换为新的数据库名称即可。

Q7: 客户端只安装了 Framework 2.0, 应该怎么样在 Framework 2.0 上面运行该系统?

A7: Framework 2.0 和 Framework 3.0/3.5 有一些不同的地方, 比如 Framework 3.0/3.5 里面有 Linq, 但是 Framework 2.0 里面就没有, 所以在整个项目里面都没有用到 Linq, 目的就是防止客户端使用的 Framework 是 2.0 的时候, 那些用到 Linq 的地方将出现错误; 如果想要在 Framework 2.0 上面运行 (该框架开发的平台为 Framework 3.5), 请在框架的属性里面设置目标 Framework 版本为 2.0, 然后重新编译, 具体方法如下:

打开项目-解决方案资源管理器-右键点击项目名称-属性-应用程序-目标 Framework-选择 2.0 即可。

Q8: 无法从加载外接程序?

A8: 开发机器上面如果安装了 DevExpress 9.2 以及 DevExpress 9.2 的破解程序, 在 Visual Studio 中弹出警告: 无法加载外接程序 DevExpress Registration 2009, 这个问题是 VS 里面不允许加载外接程序组件, 解决办法:

打开 Visual Studio-菜单里面的工具-选项-环境-外接程序/宏的安全性-勾选“允许加载外接程序组件”, 然后重新编译即可。

。

Q9: 在 Win7 下锁定屏幕的功能会出现错误信息?

A9: 屏幕锁定的功能用到了“任务管理器”, 程序里面会将“任务管理器”禁止掉, 但是在 Win7 下需要用管理员权限才可以, 如果当前登录的账户是“超级管理员”则不会出现错误信息。

Q10: 后续有新的版本是否可以免费提供?

A10: 后续如果框架有较大的改动, 作者会免费提供一份源码, 具体的事宜请和作者沟通。

Q11: 自动更新的程序如何配置?

A11: 将编译好的 AutoUpdate.exe 文件复制到主程序的根目录, 也就是和主程序放在一起, 然后通过 UpdateSet 程序来设置文件更新的参数。

Q12: 自动更新的程序连接不上服务器?

A12: 在生成自动更新的程序之前请先将 Update->DBHelper 里面的 strCon 赋值, 因为作者都是用的自己的服务器信息, 这里考虑到安全方面, 在打包发送源代码之前将这些连接字符串全部清空了。

```

8 namespace Update
9 {
10     public class DBHelper
11     {
12         string strCon = ""; //这里指定连接
13         private SqlConnection connection;
14         public SqlConnection Connection
15         {
16             get
17             {
18                 string connectionString = strCon;
19                 if (connection == null)
20                 {
21                     connection = new SqlConnection(connectionString);
22                     connection.Open();
23                 }
24                 else if (connection.State == System.Data.ConnectionState.Closed)
25                 {
26                     connection.Open();
27                 }
28                 else if (connection.State == System.Data.ConnectionState.Broken)
29                 {
30                     connection.Close();
31                     connection.Open();
32                 }
33                 return connection;
34             }
35         }
36     }
37 }

```

Q13: 第一次打开源代码出现“LC.exe 已退出，代码为-1”。

A13: 这是因为 licenses.licx 文件的问题，首先找到解决方案里面的 licenses.licx，以本项目为例：找到 Allen.Tools.Properties.licenses.licx，右键这个文件，选择删除，为了确保万一，在保存该项目的硬盘里面也将 Allen.Tools.Properties.licenses.licx 直接删除，然后重新生成解决方案即可。

Q14: 如何快速获得 DevExpress 9.2 以及 DevExpress 9.2 的破解程序？

A14: 该框架在开发的时候使用的是 DevExpress 9.2 版本，DevExpress 9.2 版本的安装程序可以通过以下链接进行下载，之前作者都是通过 QQ 发送给用户，但是发现文件太大了，发送文件的时间比较长，所以将这些文件上传至 CSDN 上面以供下载，DevExpress 9.2 一起有 6 个分卷，请全部下载之后进行解压安装，全部都是免费下载，其中 DevExpress 9.2 的破解程序由于是很早以前上传的，需要积分才可以下载，如果您的 CSDN 积分不够，请联系作者以其它方式发送给您。

补充：新版本中已经将更新模块稍作修改，更新程序与后台设置程序分开为两个项目，操作更方便。

安装文件:

<http://download.csdn.net/detail/chenpeng0118/4776596>

<http://download.csdn.net/detail/chenpeng0118/4805967>

<http://download.csdn.net/detail/chenpeng0118/4804225>

<http://download.csdn.net/detail/chenpeng0118/4805894>

<http://download.csdn.net/detail/chenpeng0118/4805904>

<http://download.csdn.net/detail/chenpeng0118/4805930>

破解文件:

<http://download.csdn.net/detail/chenpeng0118/3920563>

特别说明 1: 该框架所用到的风格样式是继承于 DevExpress 9.2, 如果您用在大型的项目里面, 请购买正版的 DevExpress 9.2 软件。

开发框架软件授权许可

您如果使用产品，则表示您接受本协议的所有条款。

该框架您购买之后可以放心应用于您的项目，不会存在版权纠纷，但是您不得基于本产品的基础，修改包装成一个与本产品功能类似的程序，进行销售和发布，参与同类软件产品市场的竞争，近期有在淘宝上面发现有人盗取此开发框架的图片，直接在淘宝里面进行出售，其产品跟原版作品毫无区别，请尊重作者的劳动成果。

免责声明

- 本软件只可用于商业营销、产品推广、资料发布等国家法律允许的用途，不得用于制造垃圾邮件、散布反动、色情等资料。任何利用本软件从事违反国家法律的活动由用户自行承担全部法律责任，与卖家无关。

交易步骤:

1. 在购买之前请首先通过 QQ 或者 Email 联系作者确定该框架是否和您的需求一致。
2. 通过与作者的沟通之后如果您确定购买, 请联系作者给您发送商品的链接地址。
3. 交易方式通过淘宝/支付宝交易, 您拍下商品之后进行付款。
4. 如果价格有打折之类的请拍下商品后及时联系作者进行修改价格。
5. 您付款之后作者会直接点击“发货”, 待作者发货之后您需要点击“确认收货”, 这里的细节后面会写到。
6. 待您点击“确认收货”之后作者会立即将源代码文件打包发送给您。

为什么需要买家先确认收货之后才能发送源代码给买家?

因为宝贝是虚拟的东西, 如果作者先将源代码发送给您, 您付的款存放在支付宝里面, 您如果觉得宝贝不符合您的要求, 以任何一个理由要求作者退款, 那么作者只有给您退款, 但是作者的源代码您却可以复制一份甚至很多份在您自己的电脑上面, 由于宝贝性质的特殊, 不像您买部手机, 不满意的话您退给卖家, 卖家至少还能得到一部手机, 所以必须是买家先确认收货之后卖家才可以发送源代码; 这也是作者必须给您解释清楚的地方; 之前的几个买家起初也是因为这个问题有点不放心, 后来经过详细的解释之后按照以上的步骤进行付款的, 到今天为止, 作者仍然在远程协助给一些买家解决一些程序里面买家不明白的问题。

联系 QQ: 969661314

联系 Email: allen0717@163.com