2 部署开发环境

2.1 前提条件

- Phenix丶搭建在 Microsoft Visual Studio 2010-2017 IDE 之上，并以 Addin、组件、控件等方式整合到 IDE 中，辅助代码编写、界面设计，以达到快速开发的目的。
- Phenix丶的界面开发框架，整合的是 Developer Express v13.2.X 版本。如希望在你的应用系统中使用其他版本的 DEV 控件，需改写 Phenix.Windows.DevExpress.v13.2.X.csproj 并重新编译即可。
- Phenix丶仅限于支持 Oracle9i 以上、SQLServer 7.0 以上的数据库。

2.2 构建配置库表、注册 Addin 工具

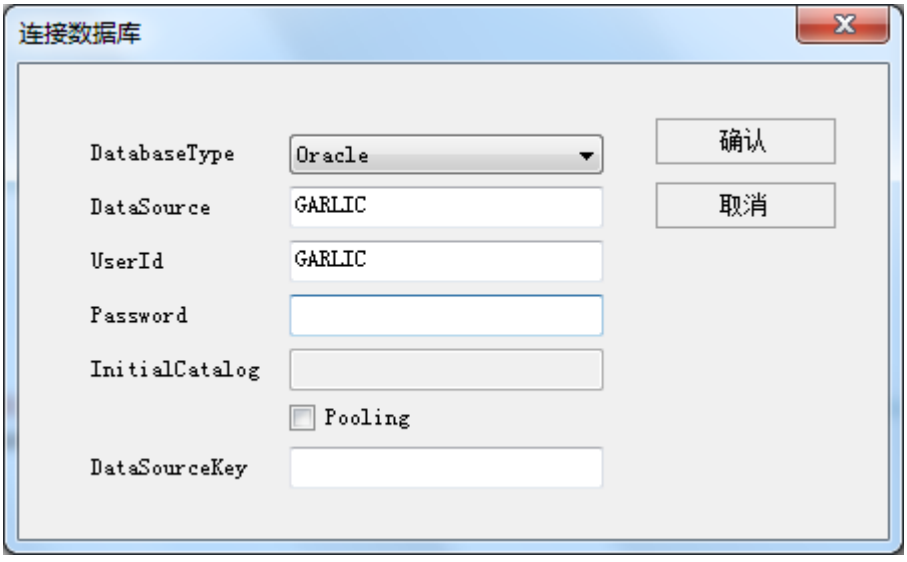
Phenix丶需要：

- 在业务数据库（Oracle9i 及以上版本、SQLServer 7.0 及以上版本）中构建一套配置库表
- 在 IDE（Microsoft Visual Studio 2010-2017 版本）中注册上 Addin 工具

才能在此基础上开发、运行应用系统。

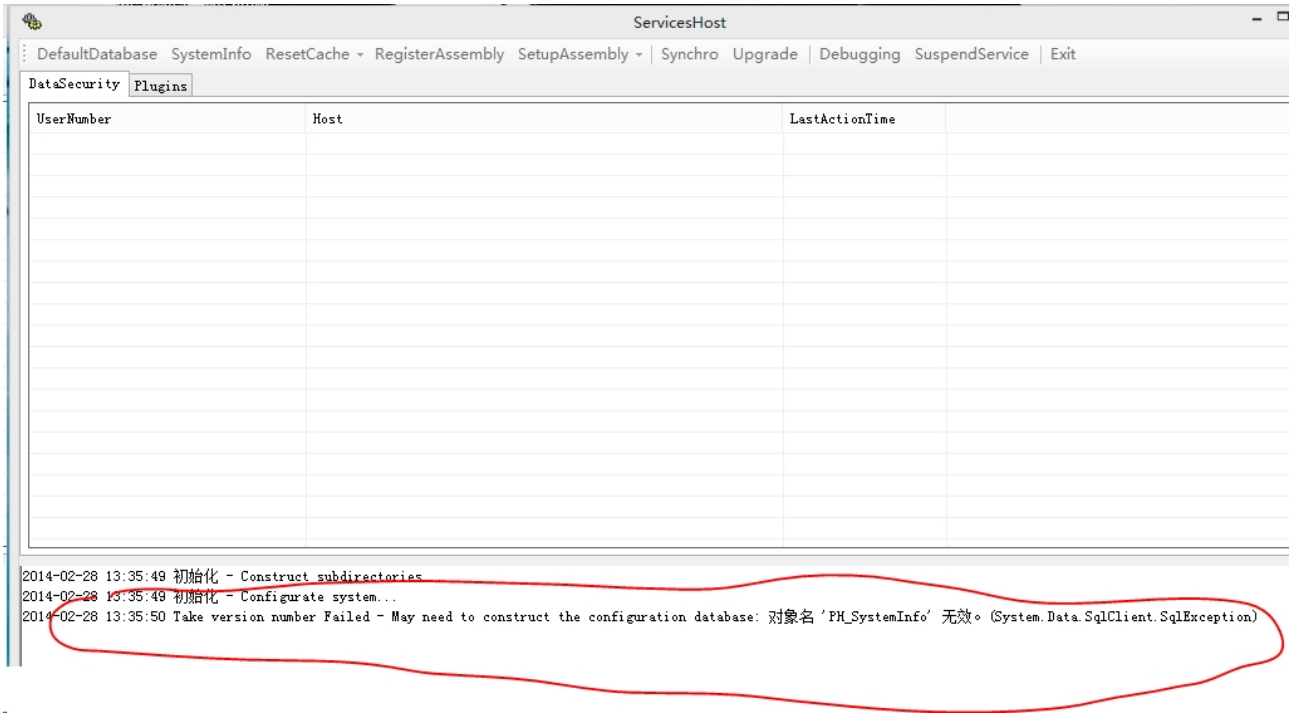
这仅需（以操作系统超级管理员身份）运行 Bin/Bin.Top 目录下的应用服务程序 Phenix.Services.Host.x86/x64.exe（请与数据库引擎版本一致），根据提示引导来完成初始化工作。

首先会弹出“连接数据库”对话框：



建议用 DBA 超级管理员权限用户登录数据库，填充好参数后点击“确认”按钮登录。

首次登录数据库时，如果未曾被自动构建过配置库表，则会在主界面的提示栏中显示“取版本号出错，可能需要构建配置库表”：



此时程序会再次弹出上述“连接数据库”对话框，需要你重新填写一遍数据库登录参数，点击“确认”后，Host 程序会在这个数据库中构建一套配置库表（Phenix丶需要用到的表结构和存储过程）。如果已构建过配置库表，程序会忽略这个过程。

登录数据库的过程如无异常，则在主界面的提示栏中显示如下类似的信息：

```
2012-01-19 13:58:12 初始化 - 构建子目录...
2012-01-19 13:58:13 初始化 - 配置系统...
2012-01-19 13:58:13 设置版本号 - 成功
2012-01-19 13:58:13 检查版本 - 完毕
2012-01-19 13:58:13 初始化主机信息 - 完毕
2012-01-19 13:58:12 同步时钟 - 完毕
2012-01-19 13:58:12 初始化 - 配置环境...
2012-01-19 13:58:18 初始化 - 程序集资料...
2012-01-19 13:59:00 初始化 - 表过滤器资料...
2012-01-19 13:59:00 初始化 - 注册服务...
```

接下来的提示是服务环境的配置信息：

```
2012-01-19 13:59:00 remoting Port - HTTP = 8084
2012-01-19 13:59:00 remoting Port - TCP = 8086
2012-01-19 13:59:06 WCF Port - HTTP = 9084
2012-01-19 13:59:06 WCF Port - TCP = 9086
2012-01-19 13:59:06 初始化 - 服务已开启
```

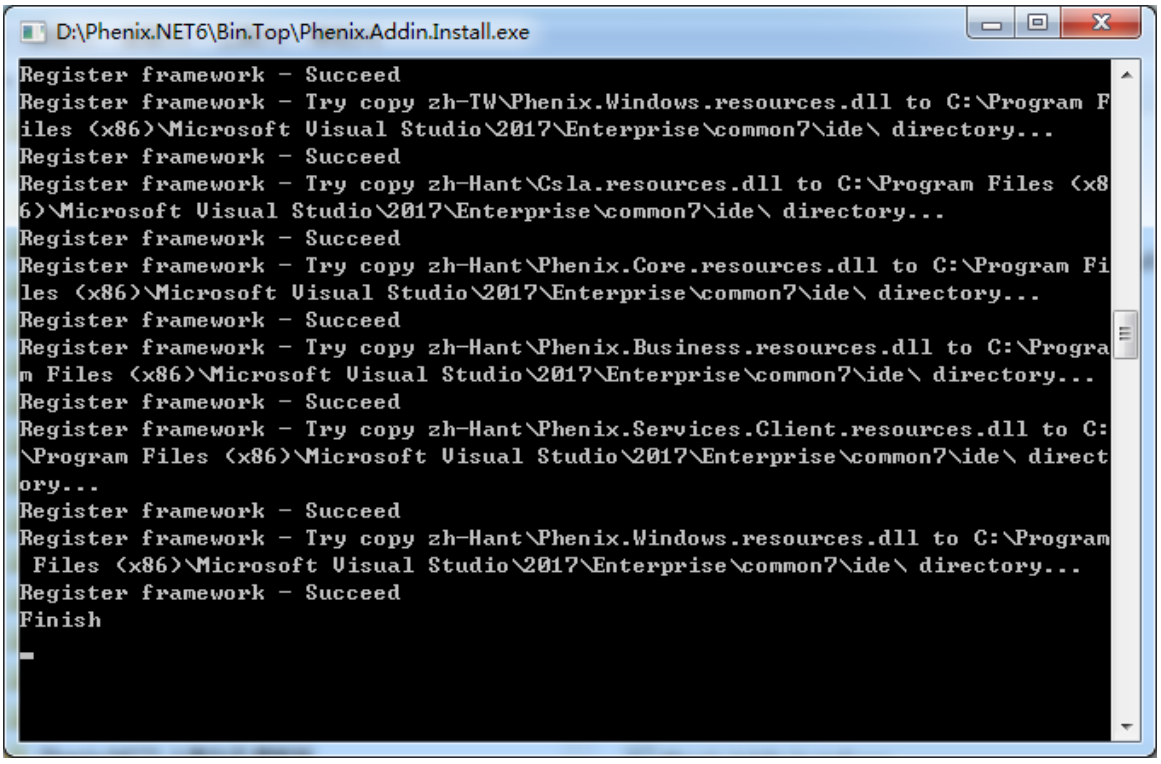
2012-01-19 13:59:06 初始化 - 完毕

2012-01-19 13:59:06 服务正常运行 - 响应中...

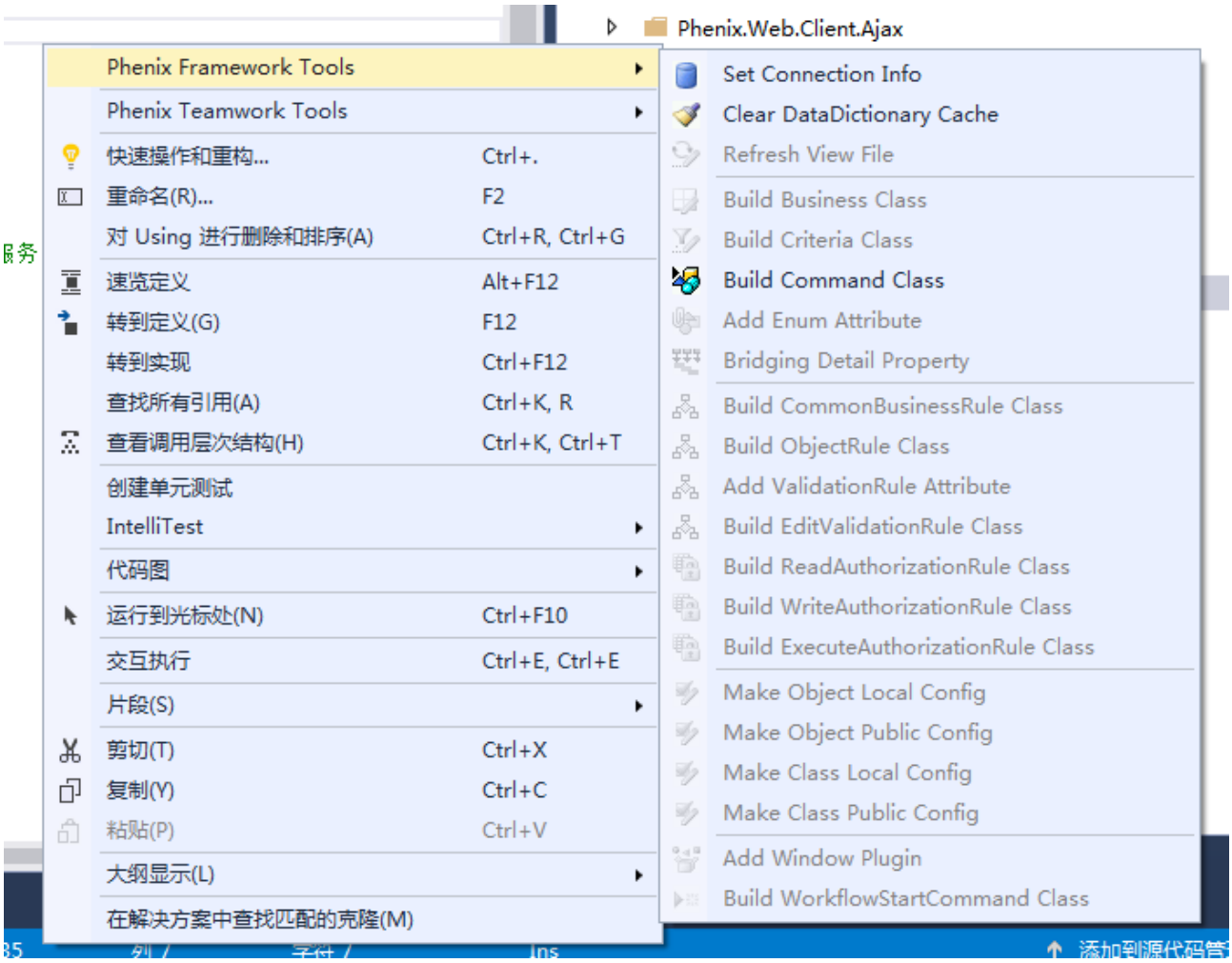
服务端和客户端之间默认协议是 Remoting TCP 8086/8087 端口，客户端无需编写额外代码，就可顺利登录到服务端。如需更改客户端连接方法（此时应该在服务端上做相应的配置以能匹配上），可参考 Phenix.Windows.Client 里被注释掉的示例代码：

```
////设置服务协议
//Phenix.Core.Net.NetConfig.ProxyType = ProxyType.Wcf;
//Phenix.Core.Net.WcfConfig.ServicesProtocolType = WcfProtocolType.BasicHttp;
```

最后，是注册 Addin 工具。注册前请关闭机器上所有 Visual Studio 程序，请在 Bin.Top 目录里找到 Phenix.Addin.Install.exe，（以操作系统超级管理员身份）运行之：



注册后如无意外，当你启动 Microsoft Visual Studio 2010-2017 IDE 后，在编辑页中可以点击右键见到“Phenix Framework Tools”菜单：



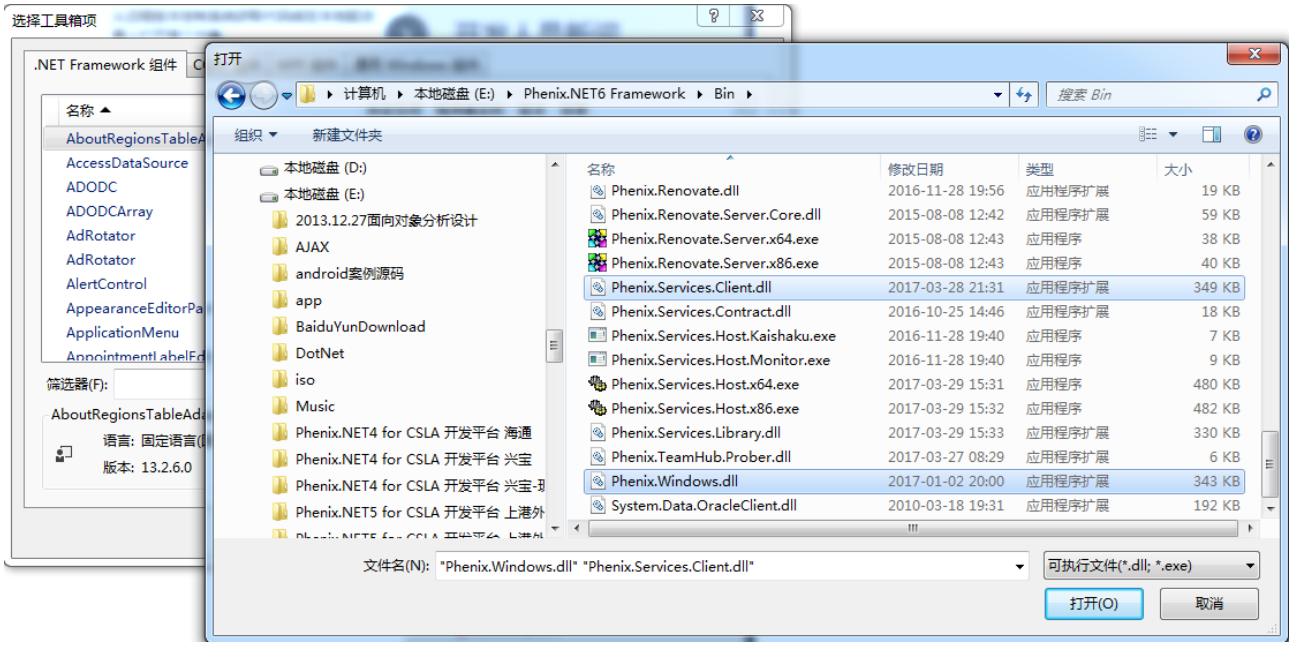
2.3 注册控件/组件

Phenix 有如下控件和组件：

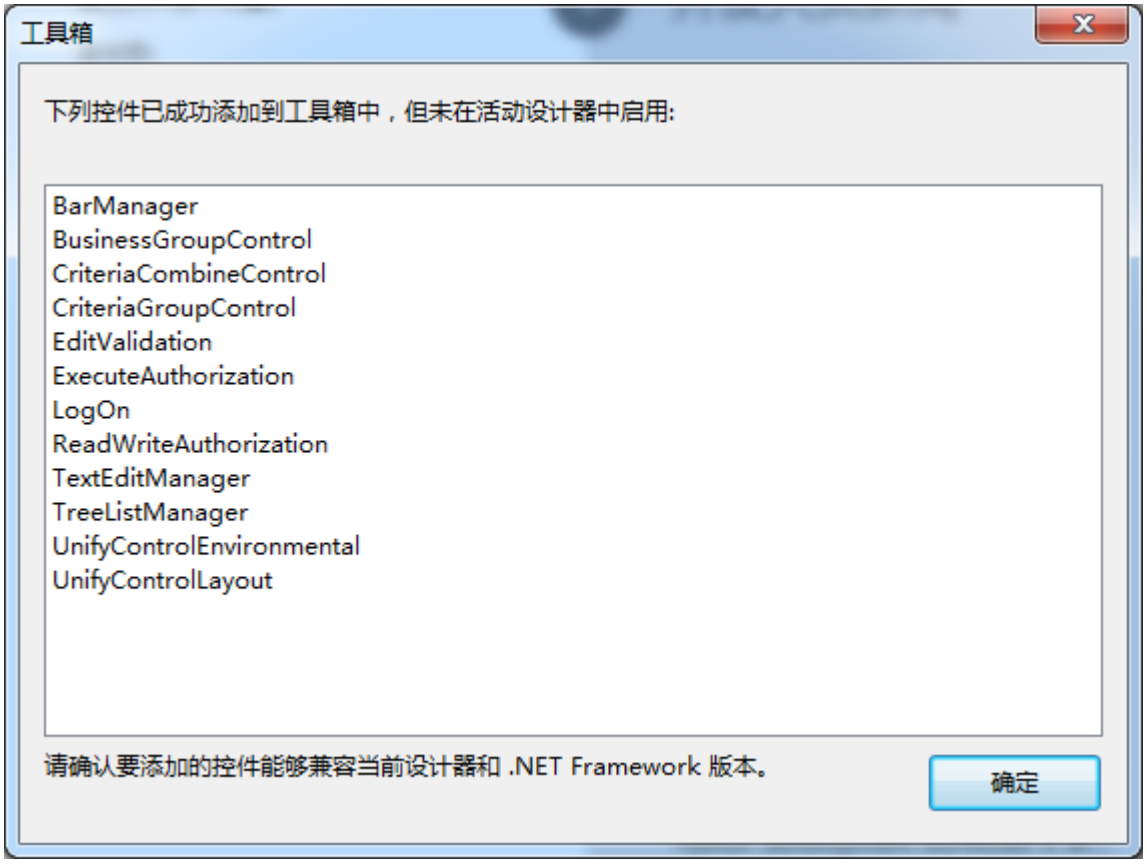
控件/组件名	所在程序集	说明
LogOn	Phenix.Services.Client	登录组件：提供缺省登陆界面，校验用户及升级服务。平台搭建配置库表时，会在配置 Ph_User 表中添加 ADMIN 用户（初始登录密码 ADMIN）供调试
ExecuteAuthorization	Phenix.Services.Client	执行授权组件：控制 WinForm 界面上组件所绑定业务对象过程的执行授权，涉及其属性：Enabled、Visible(Visibility)
ReadWriteAuthorization	Phenix.Services.Client	读写授权组件：控制 WinForm 界面上控件所绑定数据的读写授权，涉及其属性：ReadOnly（DEV 控件：Properties.ReadOnly）

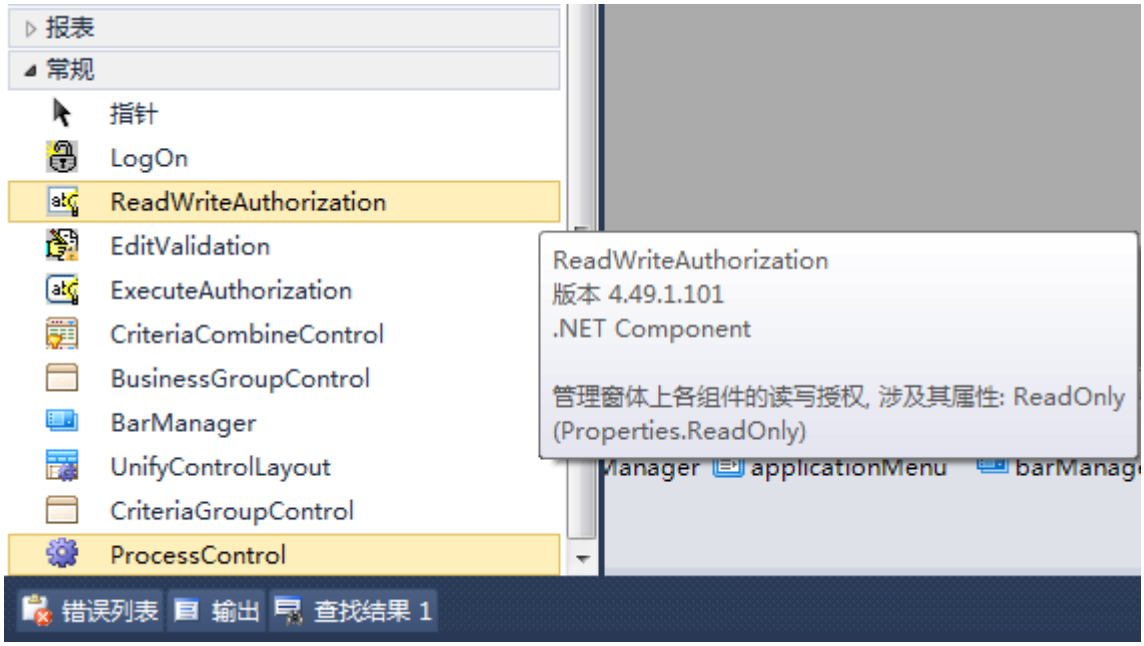
EditValidation	Phenix.Services.Client	有效性校验组件：辅助设置 WinForm 界面上绑定数据控件的一些有效性验证属性值，自动管理关联标签控件的友好性显示内容
UnifyControlLayout	Phenix.Windows	统一界面风格组件：统一界面风格
CriteriaGroupControl	Phenix.Windows	标准检索栏控件：标准检索栏
BusinessGroupControl	Phenix.Windows	标准业务栏控件：标准业务栏
BarManager	Phenix.Windows	继承自 DevExpress.XtraBars.BarManager，缺省构建了数据集的增删改查、打印、导入、导出等功能按钮；集成 EditValidation、ReadWriteAuthorization 组件，通过当前 BindingSource 以及当前用户的权限，控制 Tools Bar、Status Bar 的灰亮、显示内容及绑定数据控件的读写规则、验证规则等；Tools Bar 上的按钮除了 Help 外，都有缺省的处理过程；

为方便使用，需要将这些控件和组件放置到 IDE 的工具箱中。请：



请注意 Bin 目录是 for .net 4.0，Bin.Top 目录是 for .net 4.5，根据需要选择具体的目录文件添加到 IDE 工具栏里。

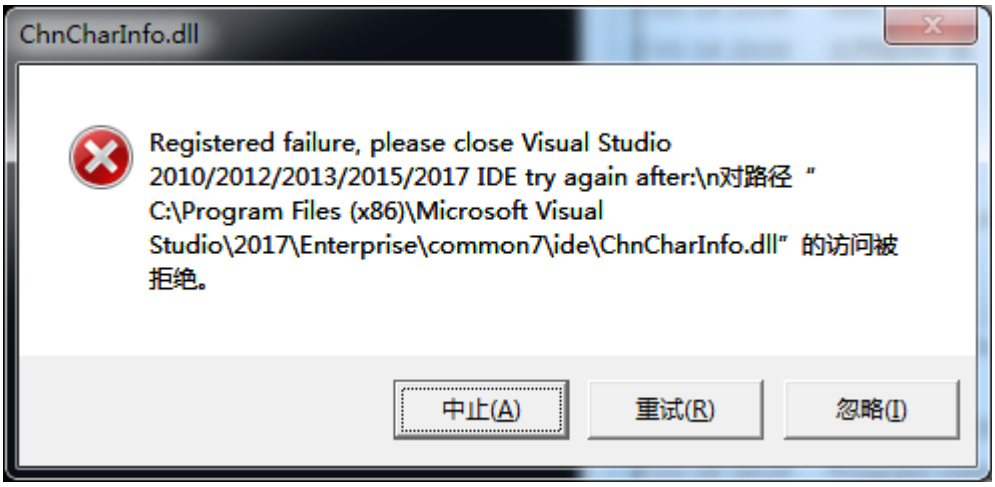




2.4 框架升级的注意事项

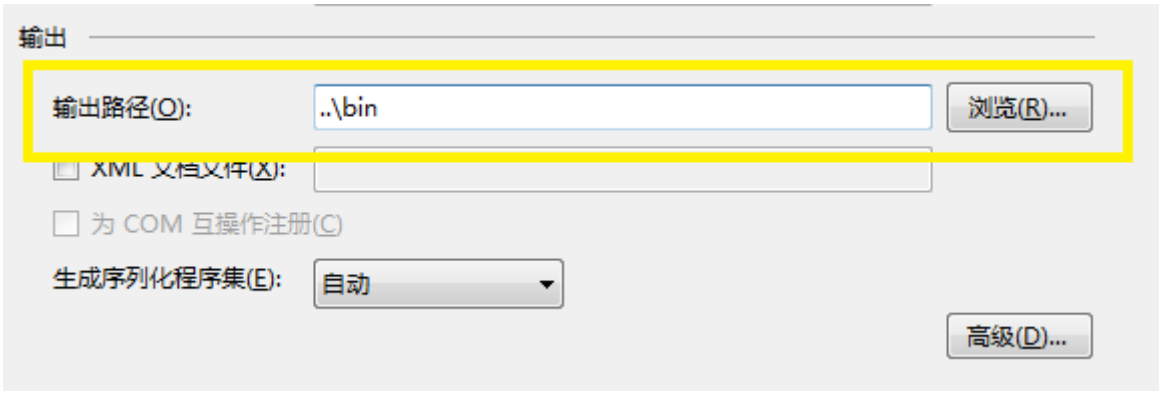
Phenix丶的历次升级，都有可能在版本号、配置库表、Addin 工具上有所改动，所以请务必运行一遍 Bin.Top 目录下的 Phenix.Services.Host.x86/x64.exe、Phenix.Addin.Install.exe 程序，并重新整理 IDE 工具栏。

如果运行当中有弹出类似以下图示的提示框，说明应该以操作系统超级管理员身份来运行它们，或者进程池中还有未杀尽的 devenv.exe 进程：

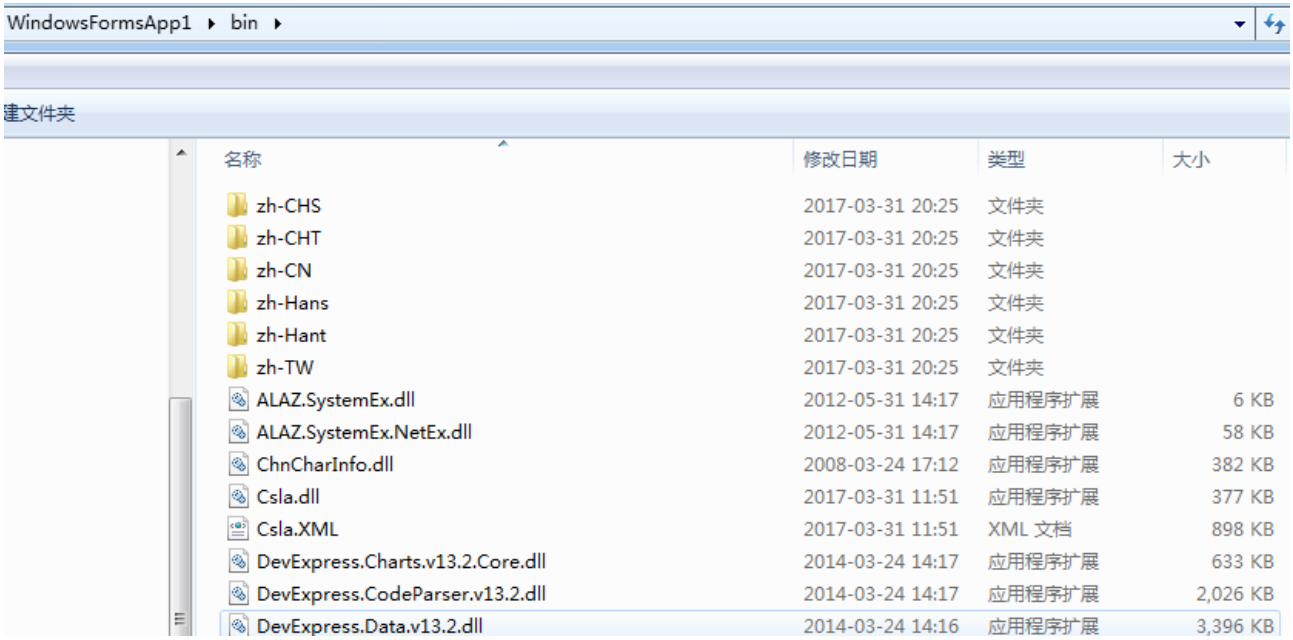


另外，升级前后框架 DLL 的版本号会不一样，为了你的业务系统的工程不受版本升级影响，请做好三件事情：

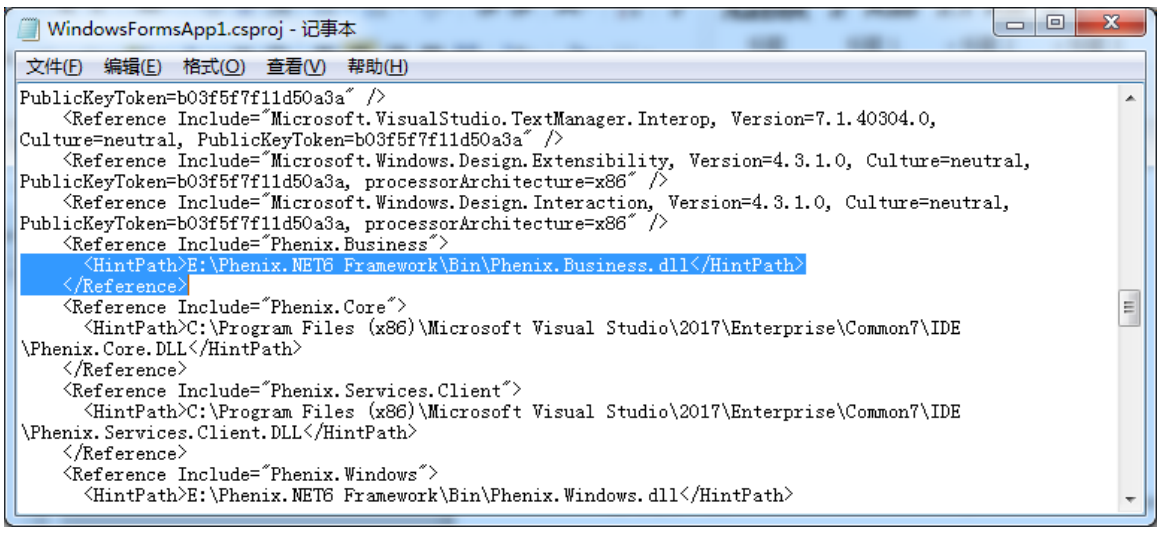
1）将全部工程的输出路径统一到一个目录上：



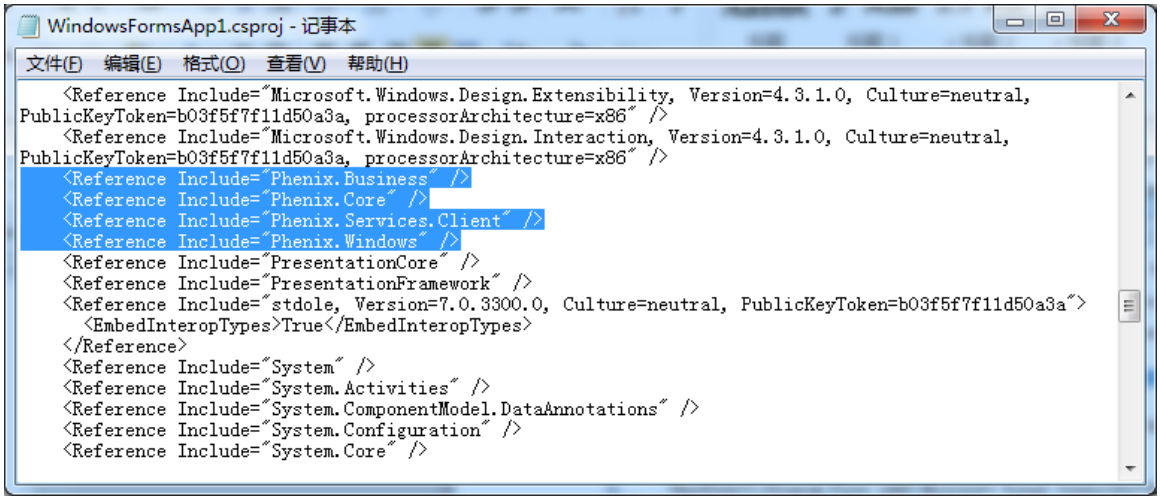
2）将引用到的框架 DLL 文件拷贝到这个目录上（.net4.0 的工程请拷贝 Bin 目录下的程序集文件，.net4.5 以上的工程请拷贝 Bin.Top 目录下的程序集文件）：



3）找到工程文件里被引用到的框架配置项，请删除它们的引用路径、版本信息：

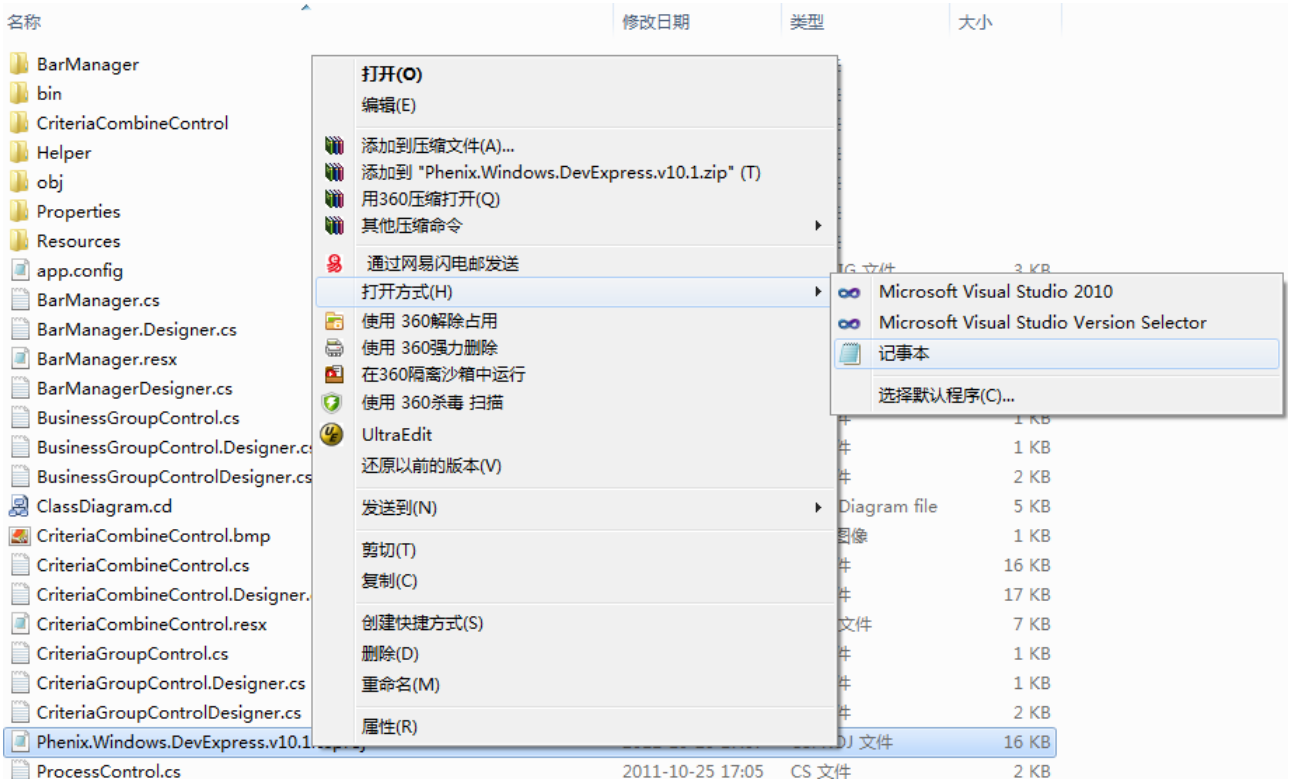


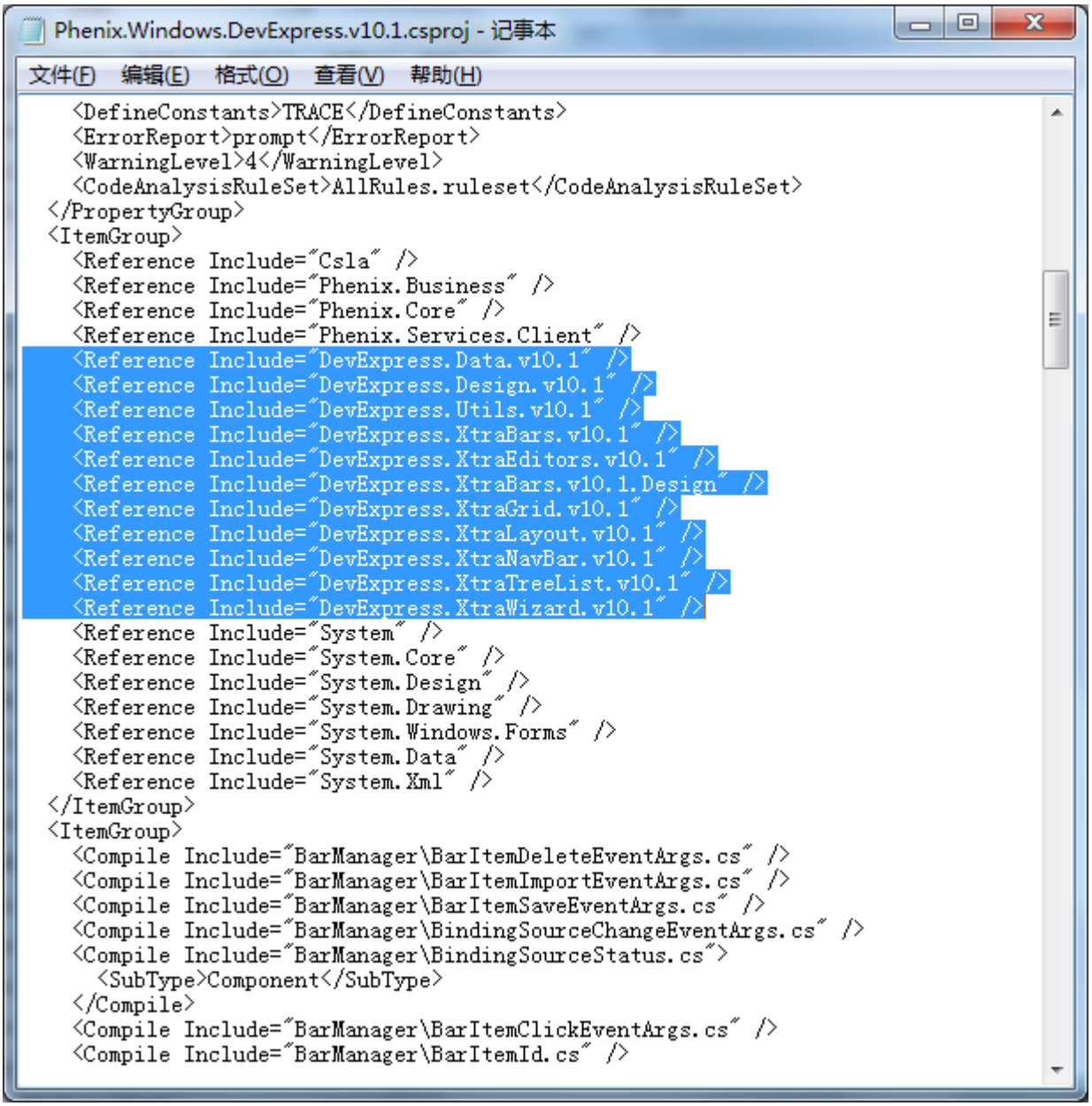
改成：



2.5 如何支持不同版本 Developer Express 控件包？

和处理框架升级一样，请用记事本打开 Phenix.Windows.DevExpress.v13.2.X.csproj 文件：





把版本号改成你自己的就可以了（高版本可能需要添加额外的 DEV 组件引用）。

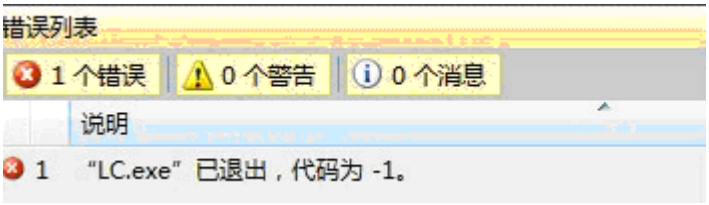
另外，在 Phenix.Windows.Helper.DesignerHelper 类中也要改掉版本号，否则 IDE 设计时会出错：

```
using System.ComponentModel.Design;
using System.Reflection;

namespace Phenix.Windows.Helper
{
  /// <summary>
  /// 设计器助手
  /// </summary>
  public static class DesignerHelper
  {
```

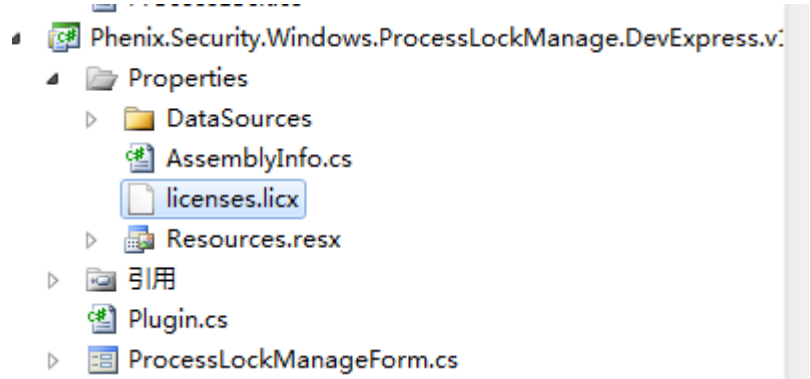
```
    internal static void ReferenceAssembly(ITypeResolutionService service)
    {
      if (service == null)
        return;
      service.ReferenceAssembly(new AssemblyName("Csla"));
      service.ReferenceAssembly(new AssemblyName("Phenix.Core"));
      service.ReferenceAssembly(new AssemblyName("Phenix.Business"));
      service.ReferenceAssembly(new AssemblyName("Phenix.Services.Client"));
      service.ReferenceAssembly(new AssemblyName("DevExpress.Data.v10.1"));
      service.ReferenceAssembly(new AssemblyName("DevExpress.Design.v10.1"));
      service.ReferenceAssembly(new AssemblyName("DevExpress.Utils.v10.1"));
      service.ReferenceAssembly(new AssemblyName("DevExpress.XtraBars.v10.1"));
      service.ReferenceAssembly(new AssemblyName("DevExpress.XtraEditors.v10.1"));
      service.ReferenceAssembly(new AssemblyName("DevExpress.XtraBars.v10.1.Design"));
      service.ReferenceAssembly(new AssemblyName("DevExpress.XtraGrid.v10.1"));
      service.ReferenceAssembly(new AssemblyName("DevExpress.XtraLayout.v10.1"));
      service.ReferenceAssembly(new AssemblyName("DevExpress.XtraNavBar.v10.1"));
      service.ReferenceAssembly(new AssemblyName("DevExpress.XtraTreeList.v10.1"));
      service.ReferenceAssembly(new AssemblyName("DevExpress.XtraWizard.v10.1"));
    }
  }
}
```

然后，保存后关闭文档，编译一遍 Phenix.Windows.DevExpress.v13.2.X.csproj 就可以了。

更改版本后，工程可能无法编译，IDE 会提示如下信息：

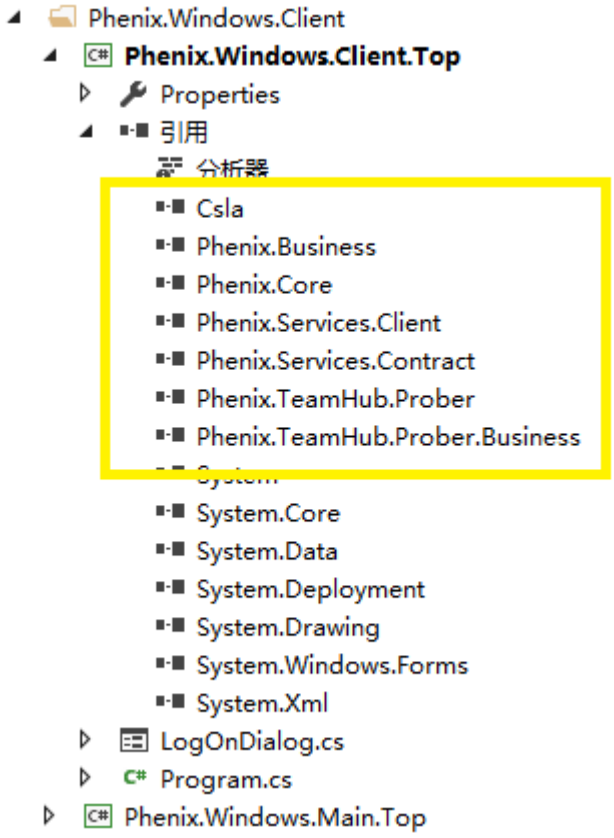


请按照下图找到 DEV 的注册文件，删掉，重新编译工程就可以了（可使用 Addin 工具的批量删除 licx 功能）。



2.6 如何调试 Winform 插件？

WinForm 插件是 Phenix丫的 Plugin 框架在业务系统 Windows 展现层的应用，可规范窗体的开发，也使得业务系统的升级成为非常便利的一件事情：



主程序仅需引用上图中的程序集，通过 ClickOnce 等工具一起打包发布到客户端，千万不要再打包进多余的程序集。因为，包括 Main 主窗体在内的所有客户端程序集，都可以 LogOnDialog 界面登录系统时，由其自动从服务端的 ClientLibrary 子目录上下载到本地来升级（当然，上图中的程序集，如有升级的话，还是要通过第三方手段（比如 ClickOnce 工具）来重新发布的）。

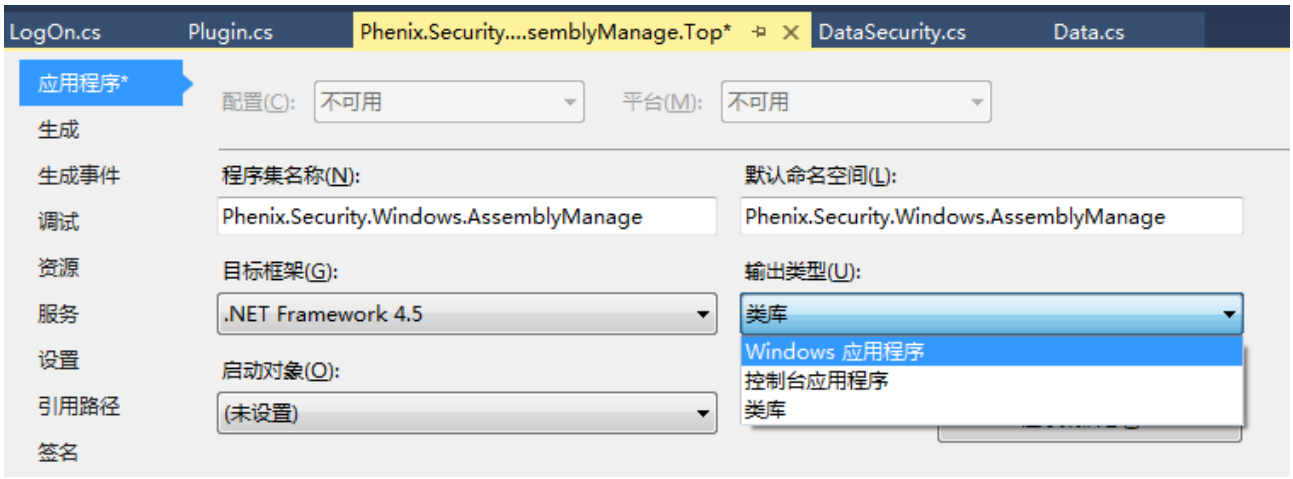
与之配套的，是 Phenix丫提供了各种手段来尽可能地便利于这些组件的调试工作。比如其 Addin 工具在构建 WinForm 插件工程的框架性代码时，已将 Program 入口代码自动写在了 Plugin 插件类的同一个文件里（至于如何构建一个 WinForm 插件工程，可参考“Addin 工具使用方法—窗体插件类”章节）：

```
namespace Phenix.Security.Windows.AssemblyManage
{
  public class Plugin : PluginBase<Plugin>
  {
    private BaseForm _mainForm;
```

```
    /// <summary>
    /// 主窗体
    /// </summary>
    public BaseForm MainForm
    {
      get { return _mainForm; }
    }

    /// <summary>
    /// 分析消息
    /// 由 PluginHost 调用
    /// </summary>
    /// <param name="message">消息</param>
    /// <returns>按需返回</returns>
    public override object AnalyseMessage(object message)
    {
      BaseForm ownerForm = message as BaseForm;
      if (ownerForm != null && ownerForm.IsMdiContainer)
      {
        _mainForm = BaseForm.ExecuteMdi<AssemblyManageForm>(ownerForm);
        return MainForm;
      }
      return BaseForm.ExecuteDialog<AssemblyManageForm>(message);
    }
  }

  /// <summary>
  /// 可变更程序集输出类型以用于调试
  /// </summary>
  static class Program
  {
    [STAThread]
    static void Main()
    {
      Application.EnableVisualStyles();
      Application.SetCompatibleTextRenderingDefault(false);

      //设为调试状态
      Phenix.Core.AppConfig.Debugging = true;
      //模拟登陆
      Phenix.Business.Security.UserPrincipal.User =
Phenix.Business.Security.UserPrincipal.CreateTester();
      Phenix.Services.Client.Library.Registration.RegisterEmbeddedWorker(false);
      //模拟启动插件
      PluginHost.Default.SendSingletonMessage("Phenix.Security.Windows.AssemblyManage", null);
    }
```

```
    }
}
```

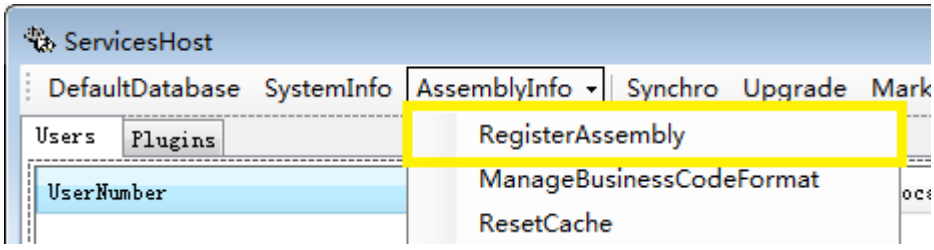
当我们需要调试这个 WinForm 插件时，可以将工程的输出类型改成“Windows 应用程序”：



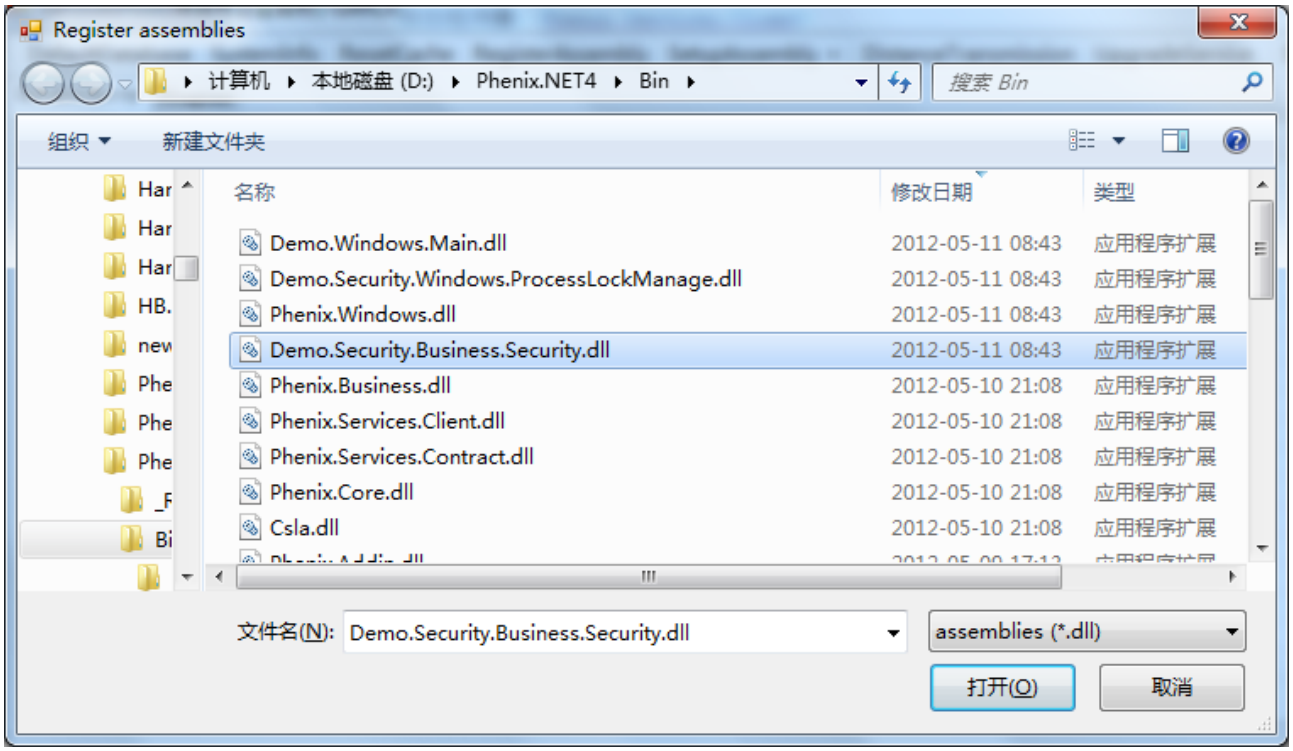
到了发布时，再改回“类库”重新编译。

2.7 注册程序集

Phenix丶有对窗体对象（界面组件等）、业务对象（属性、过程）的权限控制，有缓存机制来（比如有默认下是不带查询条件的全景 Fetch）读取业务对象，需要注册这些程序集，具体方法是在 Phenix.Services.Host.exe 上有专门的功能菜单：



点击“RegisterAssembly”功能菜单后，弹出选择待注册程序集文件的对话框：



找到后选择打开，注册时提示信息类似于：

2012-05-11 08:47:47 Initialize failure - If without registration procedures set components can be omitted the tip.：Demo.Security.Business.Security, Version=1.0.0.0, Culture=neutral, PublicKeyToken=null未找到 IPlugin 接口

2012-05-11 08:47:49 Initialize - Register class: Demo.Security.Business.Security.ProcessLock

2012-05-11 08:47:49 Initialize - Register class: Demo.Security.Business.Security.ProcessLockList

2012-05-11 08:47:49 Initialize - Registered assembly over.

如果未注册，有可能 Fetch 不到数据，或者新增数据 Save 后再 Fetch 时发现刷不出数据（其实数据库表里已保存了），还有比如权限控制不起作用，等等的奇怪问题。

如果系统开发中有升级了业务组件的程序集，需再次注册该程序集的最新版本。