

Contents

1	C#.Net COM Addin	2
1.1	创建项目	2
1.2	勾选 COM 可见、为 COM 互操作注册、签名、调试 [1]	3
1.2.1	COM 可见	3
1.2.2	为 COM 互操作注册	4
1.2.3	调试	4
1.2.4	签名	5
1.2.5	添加资源文件	5
1.3	添加引用	7
1.4	实现接口	9
1.4.1	实现 OnConnection 接口	10
1.4.2	实现 GetCustomUI 接口	10
2	编译安装	12
2.1	编译生成	12
2.2	注册安装	12
2.3	反注册 [2]	15
2.4	使用插件部署安装	16
3	VSTO 开发	17
3.1	Setup Project	17
3.2	inno setup 打包 vsto 项目	17
	Index	17

C#.Net COM Addin

一直以来总是在使用 VB6 写 comaddin，虽然用起来很顺手，但是 VB6 毕竟老了。于是乎有了向 .net 靠拢的思维。说实话，.net 我也不是很喜欢，不过开发节省时间对我这种懒人还是可以的。C#、VB.NET 除了语法差异，对平常用来说基本一样。

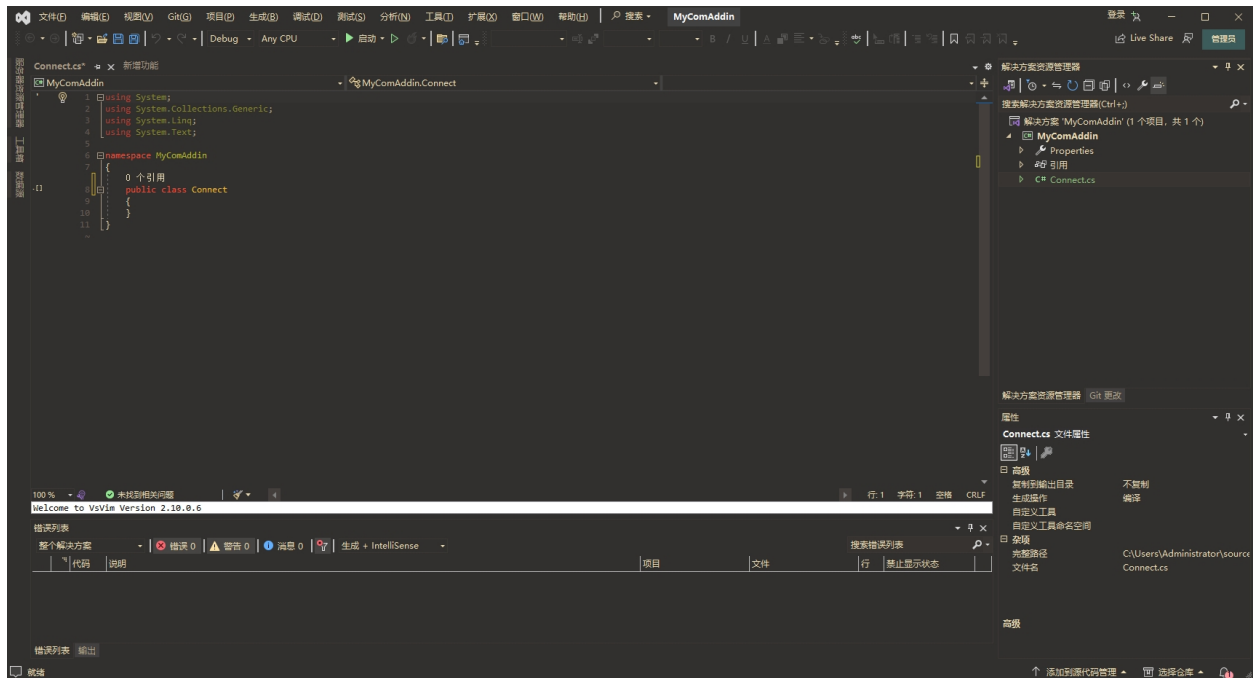
来吧，跟我一起造轮子吧（虽然很多高手很鄙视造轮子，我觉得造一遍轮子对新手练练手还是很有用的。）[3]

1.1 创建项目

VB6 写 comaddin 是引用 EXCEL、OFFICE，然后添加对 IRibbonExtensibility、IDTExtensibility2 [4] 接口的引用，然后实现接口。C# 和 VB.NET 也一样。打开 VST2022, 新建项目，选择 C#，类库（.net framework），为了兼容 XP，框架选择 .net framework 4。



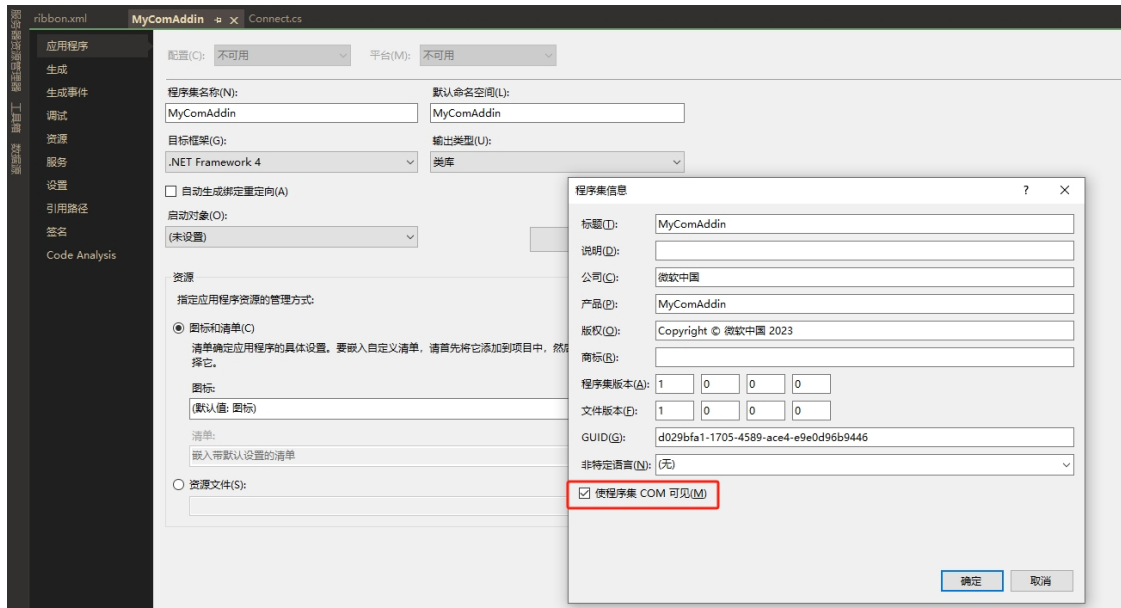
将建立项目的默认 Class1 类名改成 Connect，当然也可以自己随便起名，习惯这样。



1.2 勾选 COM 可见、为 COM 互操作注册、签名、调试 [1]

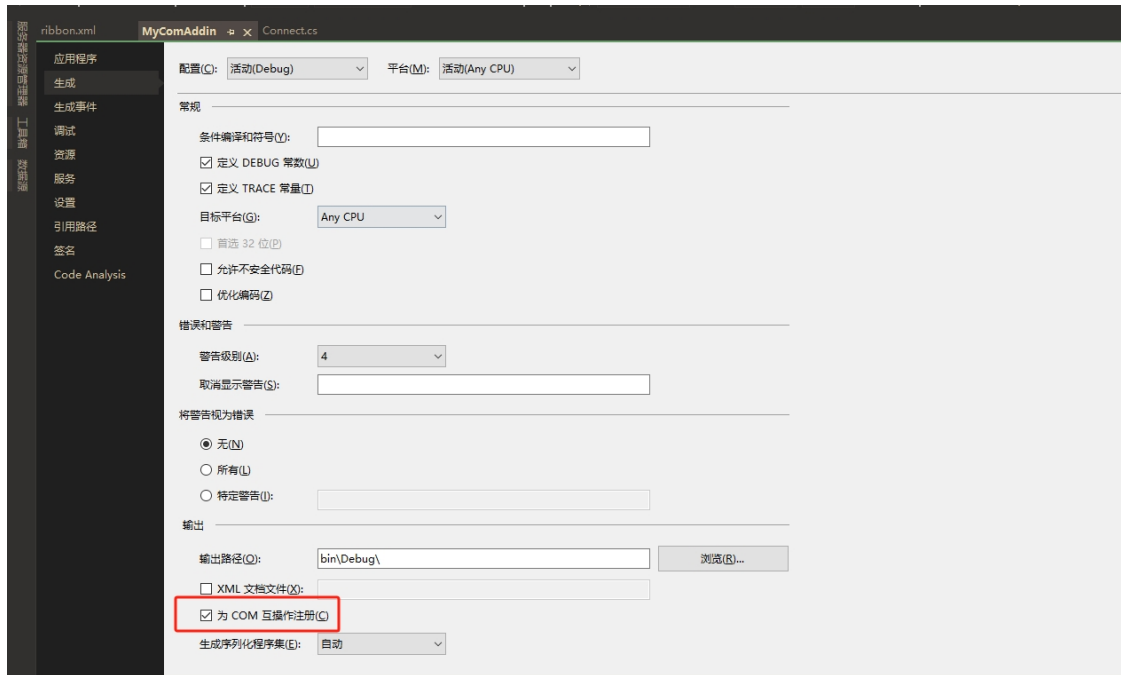
1.2.1 COM 可见

在项目属性->“应用程序”页签中，选择程序集信息，勾选使程序集 COM 可见；



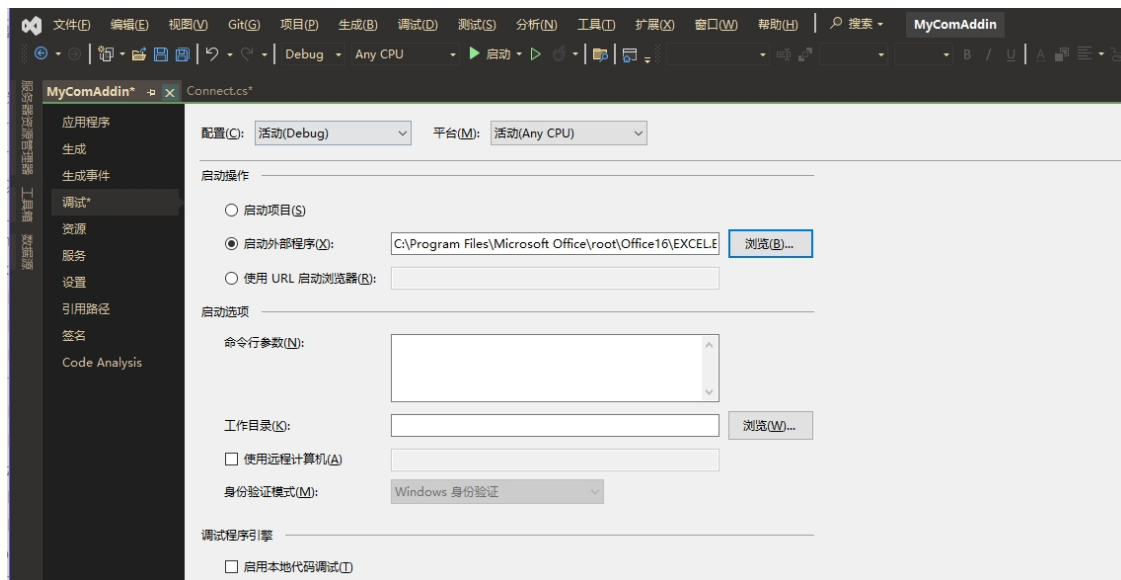
1.2.2 为 COM 互操作注册

在项目属性->“生成”页签中，输出部分勾选“为 COM 互操作注册”；目标平台：“X64”；



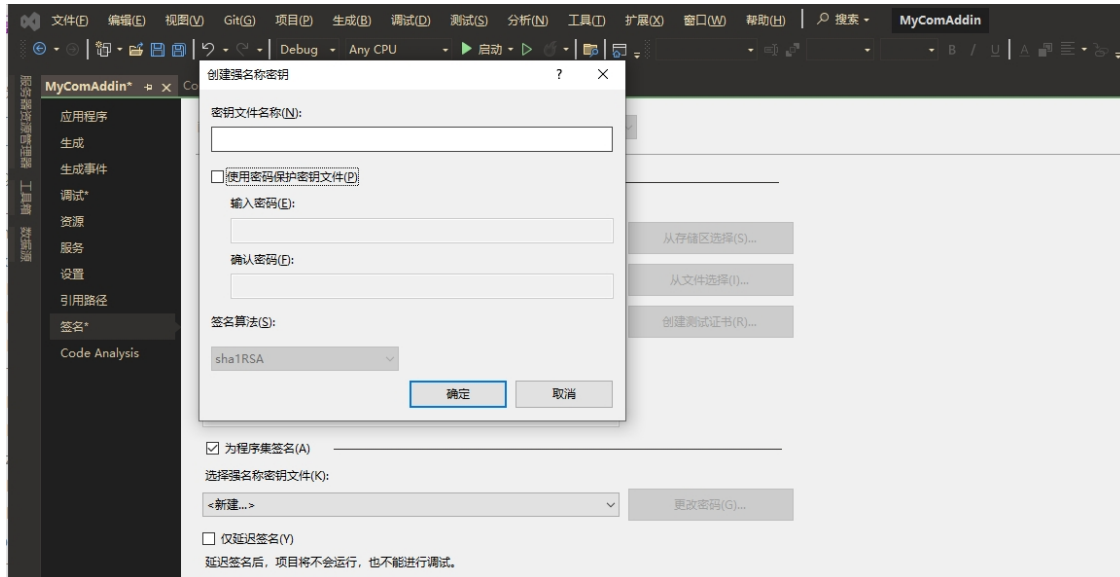
1.2.3 调试

选择项目属性，设置调试，启动外部程序，浏览到计算机里面安装的 EXCEL.EXE 路径。



1.2.4 签名

项目属性里面签名，随便新增一个签名，使用密码保护密钥文件不用勾选；输入文件名称 Conct 后，会在右侧出现 Cont.snk 文件。



1.2.5 添加资源文件

右键新建项，数据，xml 文件；xml 内容如下：

代码 1.2.1 ► XML 示例

```
1 <?xml version="1.0" encoding="utf-8" ?>
2 <customUI xmlns="http://schemas.microsoft.com/office/2006/01/customui"
3   loadImage="GetImage">
4   <ribbon>
5     <tabs>
6       <tab id="tabCustom" label="Custom">
7         <group id="groupHello" label="Hello">
```

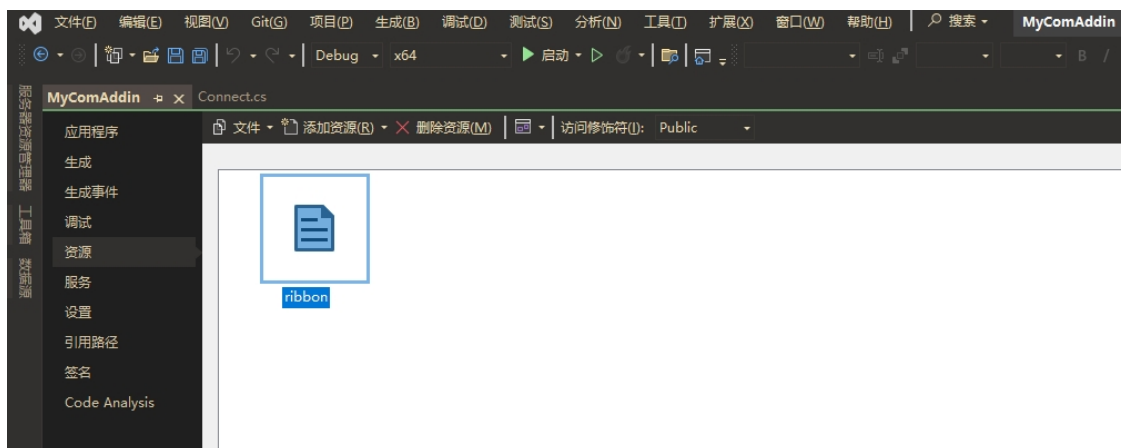
```
7      <button id="buttonHello"
8          ↪ label="HelloExcel"
9          ↪ size="large"
10         ↪ screentip="Press this for a
11         ↪ 'Hello World!' message"
12         ↪ onAction="showHello"
13         ↪ imageMso="HappyFace" />
14
15     </group>
16
17 </tab>
18
19 </tabs>
20
21 </ribbon>
22
23 </customUI>
```

相应 xml 标签的说明可以查看文档,重点说明下 onAction="showHello" 和 imageMso="HappyFace" 两个标记:

- onAction="showHello", Ribbon 区域点击相应按钮或图片后执行的代码; 具体实现由实际业务需求决定, 测试例子见代码1.4.1;
- imageMso="HappyFace", imageMso 是系统内置图标, Value 为图标的 id;



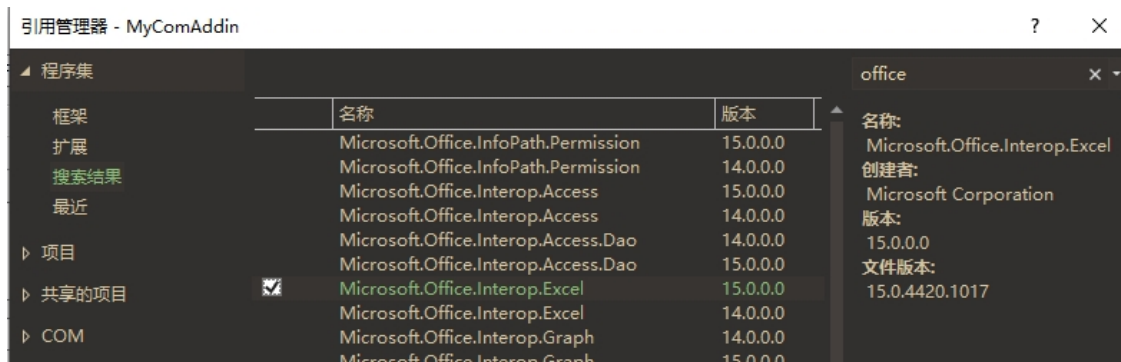
保存 xml 文件后, 将 xml 文件拖入项目属性的资源页签, 结果如下:



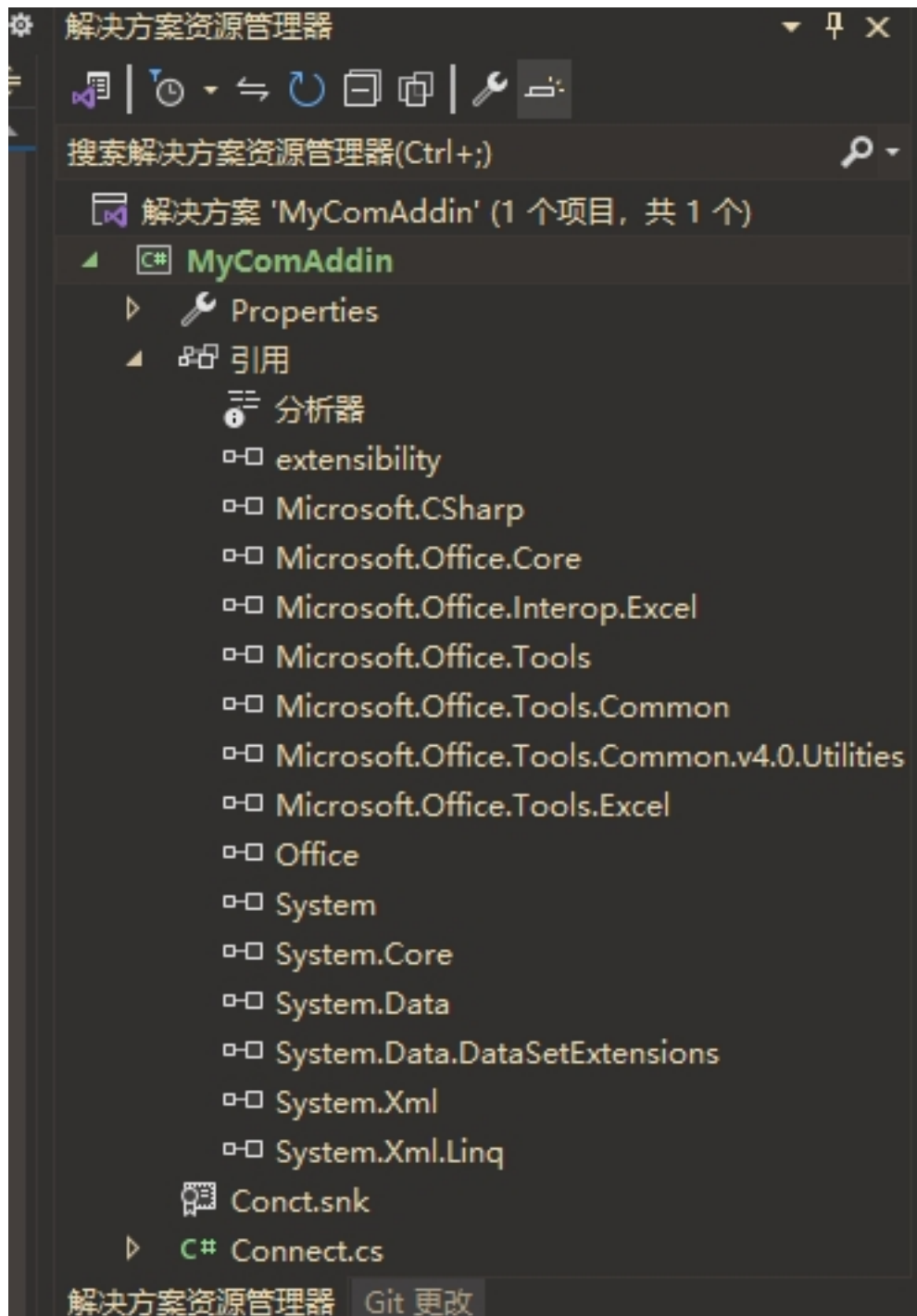
重要一点，需要修改GetCustomUII(**string** RibbonID)函数内容，具体参考1.4.1

1.3 添加引用

接下来添加 OFFICE 和 EXCEL 引用,我一般习惯添加 OFFICE14(EXCEL14 对应 OFFICE2010、EXCEL15 对应 OFFICE2013、EXCEL16 对应 Office2016)。



添加 Microsoft.Office.Core、Microsoft.Office.Interop.Excel、Microsoft.Office.Tools、Office、Microsoft.Office.Tools.Common; System.Windows.Forms; 添加 Extensibility 引用（实现 ID- TExtensibility2 接口的），最后引用如下：



在 connect 类顶部导入命名空间:

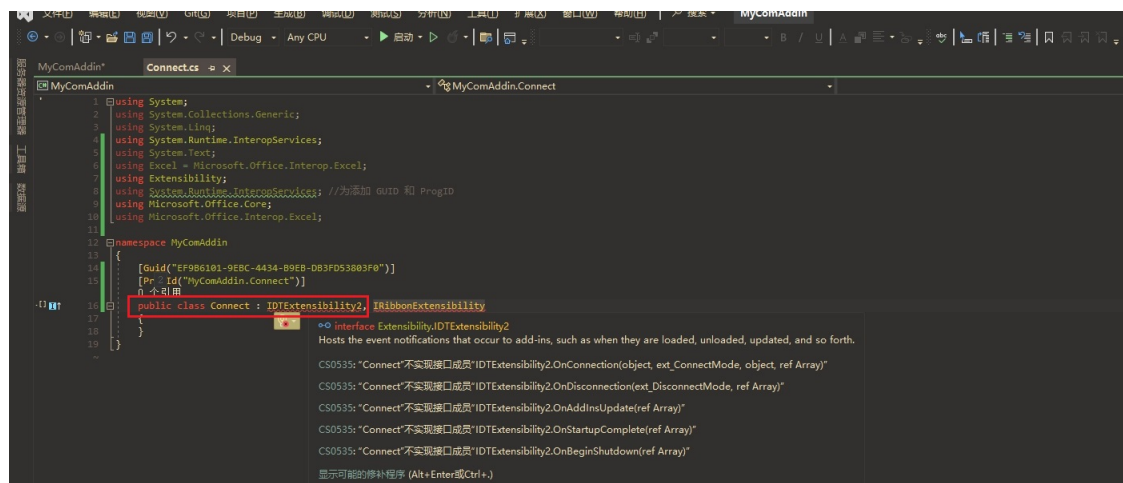
代码 1.3.1 ▶ 添加引用

```
1 using System.Reflection;  
2 using System.Windows.Forms;
```



```
3 using System.Runtime.InteropServices;
4 using Excel = Microsoft.Office.Interop.Excel;
5 using Extensibility;
6 using System.Runtime.InteropServices; //为添加 GUID 和 ProgID
7 using Microsoft.Office.Core;
8 using Microsoft.Office.Interop.Excel;
```

添加 GUID 和 ProgID，可以使用菜单工具直接生成，复制过来；生成的 GUID、ProgID 是为后面系统注册表使用的；



1.4 实现接口

COM 外接程序是一个进程内 COM 服务器或 ActiveX 动态链接库 (DLL)，可通过 IDTEExtensibility2 接口实现，如 Microsoft 外接程序设计器类型库 (Msaddndr.dll) 中所述。所有 COM 加载项都继承自此接口，必须实现其五种方法中的每个方法。具体可参考微软官方 API 说明文档 [5]：How to build an Office COM add-in by using Visual C#.NET

看下红线报错，提示必须为接口实现。分别实现上述接口 (下拉箭头，实现接口)。

```

in
{
    [Guid("EF9B6101-9EBC-4434-B9EB-DB3FD53803F0")]
    [ProgId("MyComAddin.Connect")]
    0 个引用
    public class Connect : IDTExtensibility2, IRibbonExtensibility
    {
        0 个引用
        public void OnConnection(object Application, ext_ConnectMode ConnectMode, object AddInInst, ref Array custom)
        {
            throw new NotImplementedException();
        }

        0 个引用
        public void OnDisconnection(ext_DisconnectMode RemoveMode, ref Array custom)
        {
            throw new NotImplementedException();
        }

        0 个引用
        public void OnAddInsUpdate(ref Array custom)
        {
            throw new NotImplementedException();
        }

        0 个引用
        public void OnStartupComplete(ref Array custom)
        {
            throw new NotImplementedException();
        }

        0 个引用
        public void OnBeginShutdown(ref Array custom)
        {
            throw new NotImplementedException();
        }

        0 个引用
        public string GetCustomUI(string RibbonID)
        {
            throw new NotImplementedException();
        }
    }
}

```

删除所有的Throw New NotImplementedException(); 至此，一个空架子都搭起来了。

1.4.1 实现OnConnection接口

程序入口就是OnConnection，程序结束就是OnDisconnection。在 OnConnection 里面随便写一句MessageBox.Show("C#.NET ComAddin 你好 ");;

```

MyComAddin  Connect.cs  ➤ ✕
MyComAddin
13  [Guid("EF9B6101-9EBC-4434-B9EB-DB3FD53803F0")]
14  [ProgId("MyComAddin.Connect")]
15  0 个引用
16  public class Connect : IDTExtensibility2, IRibbonExtensibility
17  {
18      0 个引用
19      public void OnConnection(object Application, ext_ConnectMode ConnectMode, object AddInInst, ref Array custom)
20      {
21          System.Windows.Forms.MessageBox.Show("C#.NET ComAddin 你好 ");
22      }
23  }

```

1.4.2 实现GetCustomUI接口

GetCustomUI 函数返回项目属性中资源文件；具体代码如下：

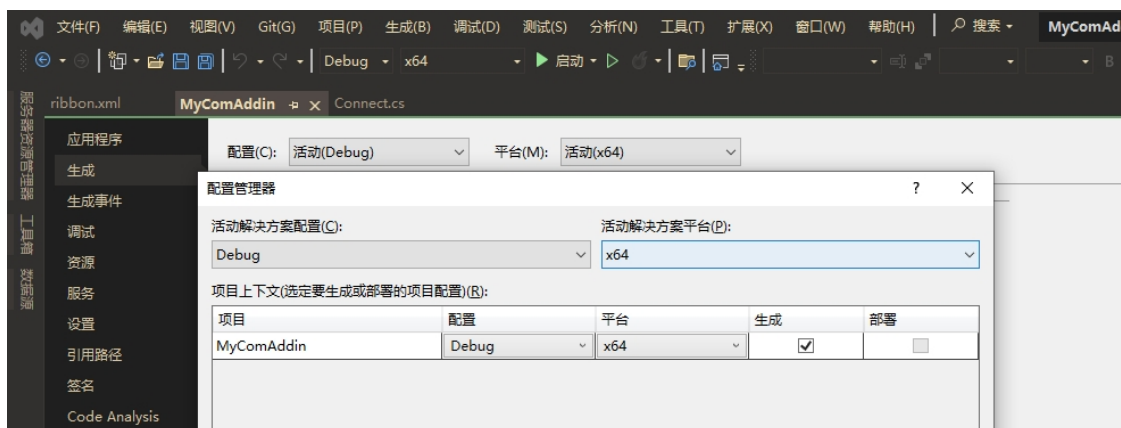
代码 1.4.1 ► GetCustomUI

```
1  public void OnBeginShutdown(ref Array custom)
2  {
3  }
4
5  public string GetCustomUI(string RibbonID)
6  {
7      return MyComAddin.Properties.Resources.ribbon;
8  }
9  public void showHello(IRibbonControl control)
10 {
11
12     MessageBox.Show(" 你点击了我哈哈!");
13 }
```

编译安装

2.1 编译生成

在“生成”->“配置管理器”里面新建 X64 平台，然后点击生成，项目开始编译生成；成功后即可注册安装测试；



2.2 注册安装

项目右键，生成，然后在文件资源管理器中打开，找到bin\x64\Debug文件夹，找到生成的MyComAddin 文件，复制到windows\system32(32 位系统) 或者 windows\syswow64 (64 位系统) 文件夹里面，

然后开始菜单栏，附件里面找到 CMD 命令，右键以管理员身份打开，切换到

C:\Windows\Microsoft.NET\Framework\v4.0.30319

运行命令，

```
>_
regasm.exe /codebase c:\windows\syswow64\MyComAddin.dll /tlb
c:\windows\syswow64\MyComAddin.tlb
```

(具体路径自己写准确)，注册 32 位 OFFICE 用。

再次切换到 C:\Windows\Microsoft.NET\Framework64\v4.0.30319 运行上述命令，注册 64

位 OFFICE 用。

codebase 参数作用 [6]: /codebase 选项与您使用 Regsvr32.exe 注册 COM 服务器的方式完全等效. 您必须选择 DLL 的特定位置, 并将该位置的路径写入注册表. 这是有风险的, COM 服务器具有强大的 DLL Hell 问题, 因为它们的注册是机器范围的. 更新该 DLL 时, 任何使用该服务器的应用程序都将受到影响. 通常不会以一种好的方式, 这经常会破坏未重新编译以使用更新的服务器器的应用程序.

.NET 通过能够存储由 [AssemblyVersion] 选择的 DLL 的多个版本来改进. Windows 并行缓存的完全等价物, 即托管程序集的 GAC. 因此, 未重建的旧应用程序将继续不受影响地运行, 仍然可以使用旧的 DLL. 这是一个好方法.

在忙于开发服务器时, 建议使用/codebase. 您可以跳过在 GAC 中注册 DLL 的额外必需步骤. 忘记这样做很痛苦, 你的更改似乎不起作用 [7, 8], 因为测试应用程序将加载旧版本. 当你使用/codebase 时, Regasm 会显示一个警告, 警告你关于 DLL Hell 的潜力, 你可以忽略它.

当您让 Visual Studio 通过”项目”>”属性”>”构建”选项卡”注册 COM 互操作”复选框进行注册时, 您将获得 Regasm/codebase/tlb 的完全等效项. 这样做更为可取, 因为它还确保了旧版本的程序集未注册, 从而避免了注册表污染. 但 VS 必须运行升级才能写入注册表.

使用隔离 COM(又名”无注册表 COM”)是最好的方法. 它允许您将 COM 服务器的副本存储在与客户端程序相同的目录中, 而您根本不需要注册它. 然而, 这需要修改客户端程序, 如果您对客户端应用程序没有任何控制权, 或者如果它是其他人喜欢混淆的那种应用程序, 则很难. 例如, Microsoft Office 应用程序.

当然这样还没完, 显示注册成功了, 你打开 EXCEL 却看不到, 因为还有注册表要写入. 新建一个 install.reg 文件, 写入如下注册表项目 [9]:

代码 2.2.1 ▶ 注册 dll

```
1 Windows Registry Editor Version 5.00
2 [HKEY_CURRENT_USER\SOFTWARE\Microsoft\Office\EXCEL\Addins\MyComAddin.C
   ~ onnect]
3 "FriendlyName"="MyComAddin"
4 "Description"="MyComAddin"
5 "LoadBehavior"=dword:00000003
6 "CommandLineSafe"=dword:00000001
7 [HKEY_CURRENT_USER\Software\Kingsoft\Office\ET\AddinsWL]
```

```
8 "MyComAddin.Connect"=""
```

卸载插件:

代码 2.2.2 ▶ 反注册 dll

```
1 Windows Registry Editor Version 5.00
2 [-HKEY_CURRENT_USER\SOFTWARE\Microsoft\Office\EXCEL\Addins\MyComAddin.]
   ↳ Connect]
3 [HKEY_CURRENT_USER\Software\Kingsoft\Office\ET\AddinsWL]
4 "MyComAddin.Connect"=-
```

再创建安装 bat:

代码 2.2.3 ▶ 安装

```
1 @echo off
2
3 @set baseDir="%cd%"
4 Echo.
5 Echo 【1】 导入注册表
6 regedit /s %baseDir%\install.reg
7 Echo.
8 Echo 【2】 注册类型
9 C:\Windows\Microsoft.NET\Framework64\v4.0.30319\RegAsm.exe /codebase
   ↳ %baseDir%\MyComAddin.dll /tlb:%baseDir%\MyComAddin.tlb
10 Echo.
11 Echo 【3】 添加程序集到 GAC
12 @SETGACUTIL="%baseDir%\NETFX 4.0 Tools\gacutil.exe"
13 %GACUTIL%-i %baseDir%\WPP_test.dll
14 %GACUTIL%-i %baseDir%\PowerPoint.dll
15 %GACUTIL%-i %baseDir%\Office.dll
16 pause
```

创建卸载 bat:

代码 2.2.4 ▶ 卸载

```
1 @echo off
2 @set baseDir="%cd%"
3 Echo.
4 Echo 【1】 从缓存中移除程序集
5 @SETGACUTIL="%baseDir%\NETFX 4.0 Tools\gacutil.exe"
6 rd /s /QC:\Windows\Microsoft.NET\assembly\GAC_MSIL\WPP_test
7 rd /s /QC:\Windows\Microsoft.NET\assembly\GAC_MSIL\PowerPoint
8 rd /s /QC:\Windows\Microsoft.NET\assembly\GAC_MSIL\Office
9 Echo.
10
11 Echo 【2】 注销类型
12 C:\Windows\Microsoft.NET\Framework64\v4.0.30319\RegAsm.exe /u
   %baseDir%\MyComAddin.dll /tlb:%baseDir%\MyComAddin.tlb
13 Echo.
14
15 Echo 【3】 清除注册表
16 regedit /s %baseDir%\uninstall.reg
17 pause
```

至此全部完成，打开 EXCEL 会看到如下效果。

2.3 反注册 [2]

How to Unregister a DLL using RegAsm.exe?[10]

To run the tool, use Visual Studio Developer Command Prompt or Visual Studio Developer PowerShell. Unregistering a DLL using RegAsm.exe is so easy as registering.

Open Command Prompt and run the following command replacing the <dllfilename> name with the name you want to unregister.

```
1 regasm /u <dllfilename>.dll
```

To unregister the DLL completely, you have to unregister the type library of the DLL too. To do so, run the following command.

```
1 regasm <dllfilename> /tlb /unregister
```

2.4 使用插件布署安装

VSTO 开发

3.1 Setup Project

3.2 inno setup 打包 vsto 项目

Index

代码

1.2.1	XML 示例	5	1.4.1	GetCustomUI	11
1.3.1	添加引用	8	2.2.1	注册 dll	13
			2.2.2	反注册 dll	14
			2.2.3	安装	14
			2.2.4	卸载	15

Bibliography

- [1] 来吧! 我在未来等你! Visualstudio 如何进行 onenote 插件开发. https://blog.csdn.net/qq_40640910/article/details/129998186.
- [2] 柠檬草的味道 qw. C# 写的 com 组件注册问题兼论微软 regasm 注册的 bug. <https://blog.csdn.net/a331003411/article/details/127497372>.
- [3] qq_24968317. Vb.net comaddin 从零开发, 跟我一起造轮子, 嗨起来. https://blog.csdn.net/qq_24968317/article/details/107615473.
- [4] flylib. Implementing idtextensibility2. https://flylib.com/books/en/2.53.1/implementing_idtextensibility2.html.
- [5] microsoft. How to build an office com add-in by using visual c# .net. <https://learn.microsoft.com/en-US/previous-versions/office/troubleshoot/office-developer/office-com-add-in-using-visual-c>.
- [6] itecnote. Regasm -when is the /codebase option applicable. <https://itecnote.com/tecnote/regasm-when-is-the-codebase-option-applicable/>.
- [7] microsoft. Troubleshooting com add-in load failures. <https://learn.microsoft.com/zh-cn/archive/blogs/vsod/troubleshooting-com-add-in-load-failures>.
- [8] microsoft. Not loaded. a runtime error occurred during the loading of the com add-in. https://answers.microsoft.com/en-us/outlook_com/forum/all/not-loaded-a-runtime-error-occurred-during-the/3f50b327-8edd-4931-b93e-5280b7734d7a.
- [9] 多见多闻. C# wps 插件开发流程. <https://www.cnblogs.com/guangzhiruijie/p/17217942.html>.
- [10] thewindowsclub. How to unregister a dll using regasm.exe. <https://www.thewindowsclub.com/what-is-regasm-exe-how-to-unregister-a-dll>.