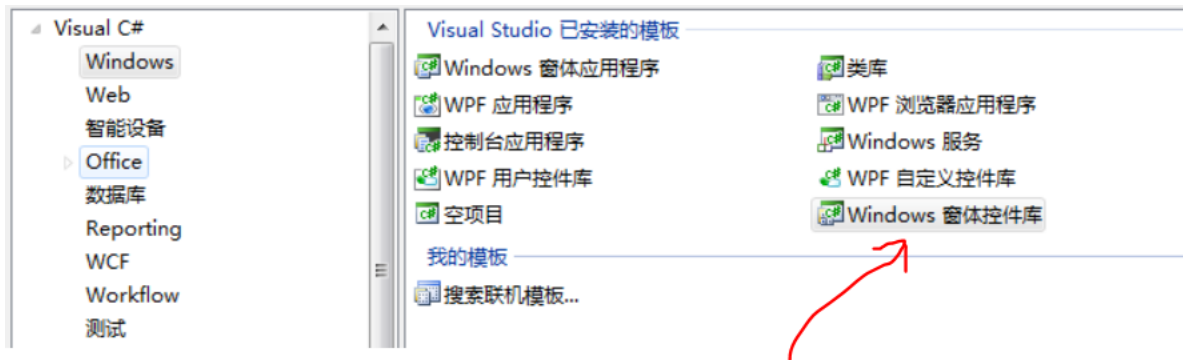
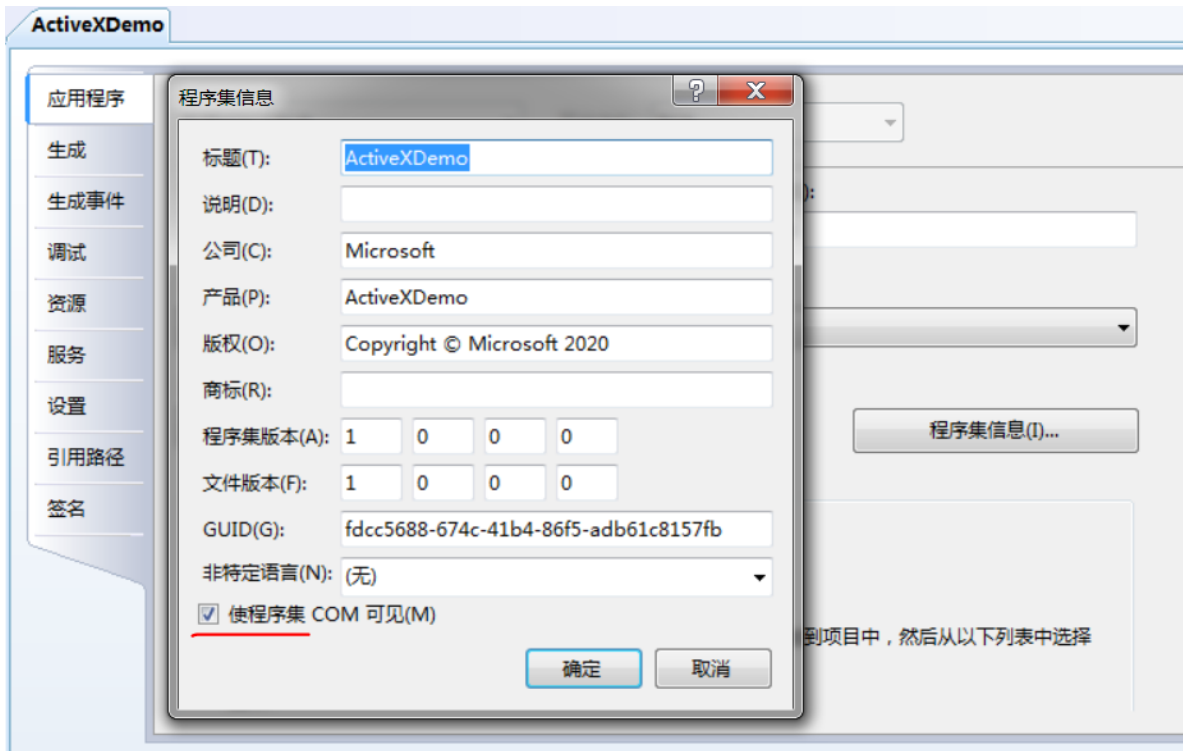


# ActiveX控件开发

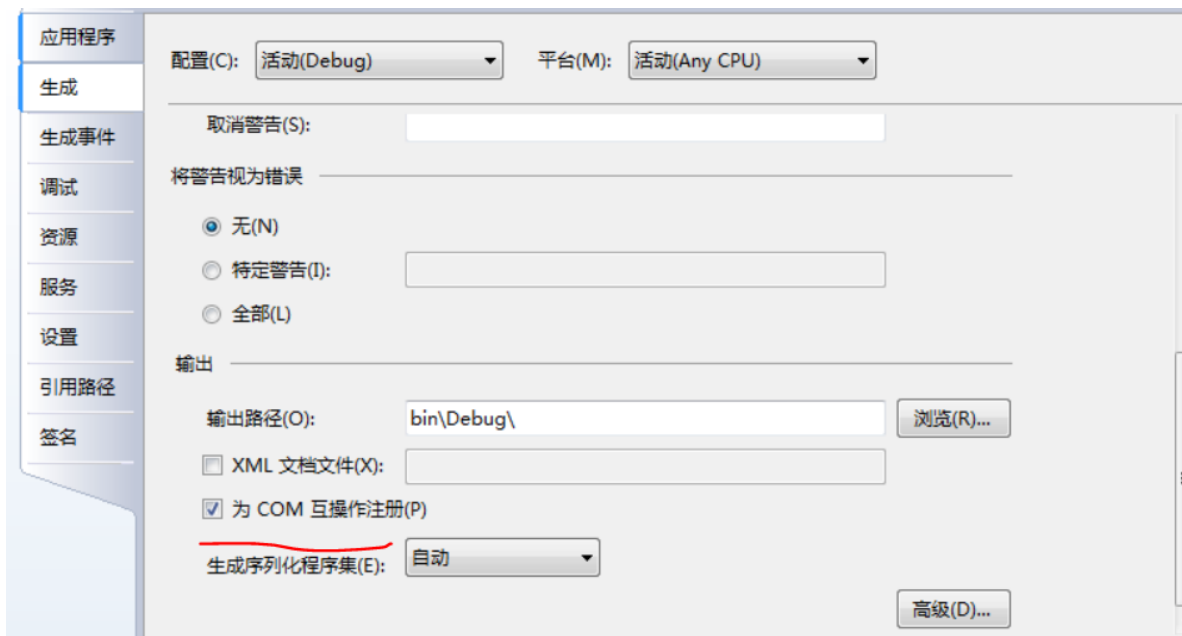
- 新建Windows 窗体控件库



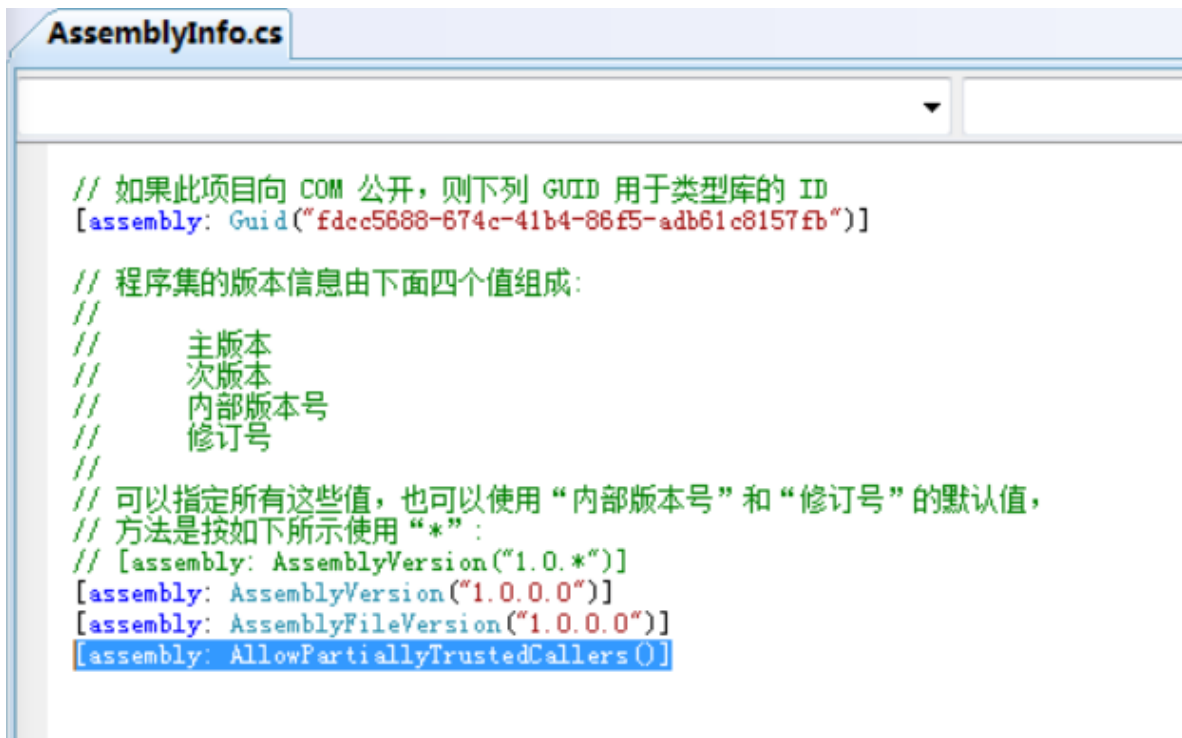
- 项目右键属性->程序集信息，勾选 使程序集COM可见(M)



- 选择生成，拉到下面，勾选 为COM互操作注册(P)



- 在AssemblyInfo.cs下方添加[assembly: AllowPartiallyTrustedCallers()]



- 新建IOObjectSafety接口，啥也不要改

```
using System;
using System.Collections.Generic;
using System.Linq;
using System.Text;
using System.Runtime.InteropServices;

namespace ActiveXDemo
{
    [ComImport, GuidAttribute("CB5BDC81-93C1-11CF-8F20-00805F2CD064")]
    [InterfaceTypeAttribute(ComInterfaceType.InterfaceIsIUnknown)]
    public interface IOObjectSafety
    {
        [PreserveSig]
    }
}
```

```

        int GetInterfaceSafetyOptions(ref Guid riid,
[MarshalAs(UnmanagedType.U4)] ref int pdwSupportedOptions,
[MarshalAs(UnmanagedType.U4)] ref int pdwEnabledOptions);

        [PreserveSig()]
        int SetInterfaceSafetyOptions(ref Guid riid,
[MarshalAs(UnmanagedType.U4)] int dwOptionSetMask, [MarshalAs(UnmanagedType.U4)]
int dwEnabledOptions);
    }
}

```

- 新建ICommonControl接口, GUID可通过 工具->创建GUID(G) 来生成

```

using System;
using System.Collections.Generic;
using System.Linq;
using System.Text;
using System.Runtime.InteropServices;

namespace ActiveXDemo
{
    [Guid("55D75C5C-B03C-46FD-909D-C3BC6CF0DC4A"),
InterfaceType(ComInterfaceType.InterfaceIsIDispatch)]
    public interface ICommonControl
    {
        [DispId(1)]
        int Test(int a, int b);
    }
}

```

- 实现ICommonControl, IObjectSafety (这个的实现代码不要改动)

```

using System;
using System.Collections.Generic;
using System.ComponentModel;
using System.Drawing;
using System.Data;
using System.Linq;
using System.Text;
using System.Windows.Forms;
using System.Runtime.InteropServices;

namespace ActiveXDemo
{
    [Guid("C3E3D291-2D00-4D3F-AD0F-23423EBAAEBC"),
ProgId("ActiveXDemo.UserControl1"), ComVisible(true),
ClassInterface(ClassInterfaceType.None),
ComDefaultInterface(typeof(ICommonControl))]
    public partial class UserControl1 : UserControl, ICommonControl,
IObjectSafety
    {
        public UserControl1()
        {
            InitializeComponent();
        }
    }
}

```

```

    }

    private void button1_Click(object sender, EventArgs e)
    {
        label1.Text = textBox1.Text;
    }

    #region IOObjectSafety 成员

    private const string _IID_IDispatch = "{00020400-0000-0000-C000-000000000046}";
    private const string _IID_IDispatchEx = "{a6ef9860-c720-11d0-9337-00a0c90dcaa9}";
    private const string _IID_IPersistStorage = "{0000010A-0000-0000-C000-000000000046}";
    private const string _IID_IPersistStream = "{00000109-0000-0000-C000-000000000046}";
    private const string _IID_IPersistPropertyBag = "{37D84F60-42CB-11CE-8135-00AA004BB851}";

    private const int INTERFACESAFE_FOR_UNTRUSTED_CALLER = 0x00000001;
    private const int INTERFACESAFE_FOR_UNTRUSTED_DATA = 0x00000002;
    private const int S_OK = 0;
    private const int E_FAIL = unchecked((int)0x80004005);
    private const int E_NOINTERFACE = unchecked((int)0x80004002);

    private bool _fSafeForScripting = true;
    private bool _fSafeForInitializing = true;

    public int GetInterfaceSafetyOptions(ref Guid riid, ref int pdwSupportedOptions, ref int pdwEnabledOptions)
    {
        int Rslt = E_FAIL;

        string strGUID = riid.ToString("B");
        pdwSupportedOptions = INTERFACESAFE_FOR_UNTRUSTED_CALLER | INTERFACESAFE_FOR_UNTRUSTED_DATA;
        switch (strGUID)
        {
            case _IID_IDispatch:
            case _IID_IDispatchEx:
                Rslt = S_OK;
                pdwEnabledOptions = 0;
                if (_fSafeForScripting == true)
                    pdwEnabledOptions = INTERFACESAFE_FOR_UNTRUSTED_CALLER;
                break;
            case _IID_IPersistStorage:
            case _IID_IPersistStream:
            case _IID_IPersistPropertyBag:
                Rslt = S_OK;
                pdwEnabledOptions = 0;
                if (_fSafeForInitializing == true)
                    pdwEnabledOptions = INTERFACESAFE_FOR_UNTRUSTED_DATA;
                break;
            default:
                Rslt = E_NOINTERFACE;
                break;
        }
    }

```

```

        return Rslt;
    }

    public int SetInterfaceSafetyOptions(ref Guid riid, int dwOptionSetMask,
int dwEnabledOptions)
    {
        int Rslt = E_FAIL;
        string strGUID = riid.ToString("B");
        switch (strGUID)
        {
            case _IID_IDispatch:
            case _IID_IDispatchEx:
                if (((dwEnabledOptions & dwOptionSetMask) ==
INTERFACESAFE_FOR_UNTRUSTED_CALLER) && (_fSafeForScripting == true))
                    Rslt = S_OK;
                break;
            case _IID_IPersistStorage:
            case _IID_IPersistStream:
            case _IID_IPersistPropertyBag:
                if (((dwEnabledOptions & dwOptionSetMask) ==
INTERFACESAFE_FOR_UNTRUSTED_DATA) && (_fSafeForInitializing == true))
                    Rslt = S_OK;
                break;
            default:
                Rslt = E_NOINTERFACE;
                break;
        }

        return Rslt;
    }

#endregion

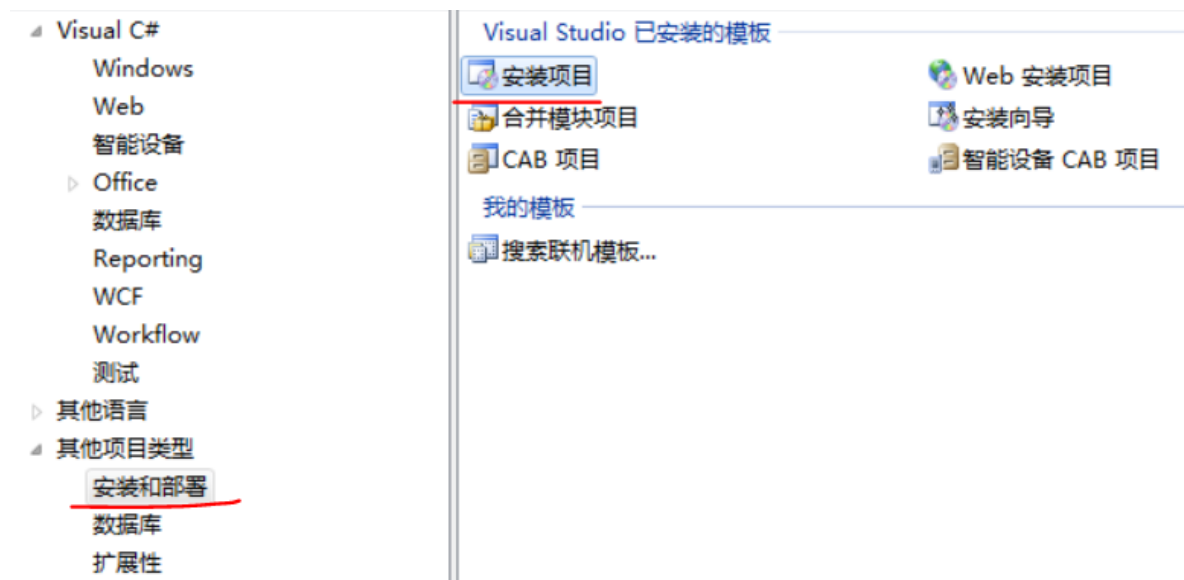
#region ICommonControl 成员

    public int Test(int a, int b)
    {
        return a + b;
    }

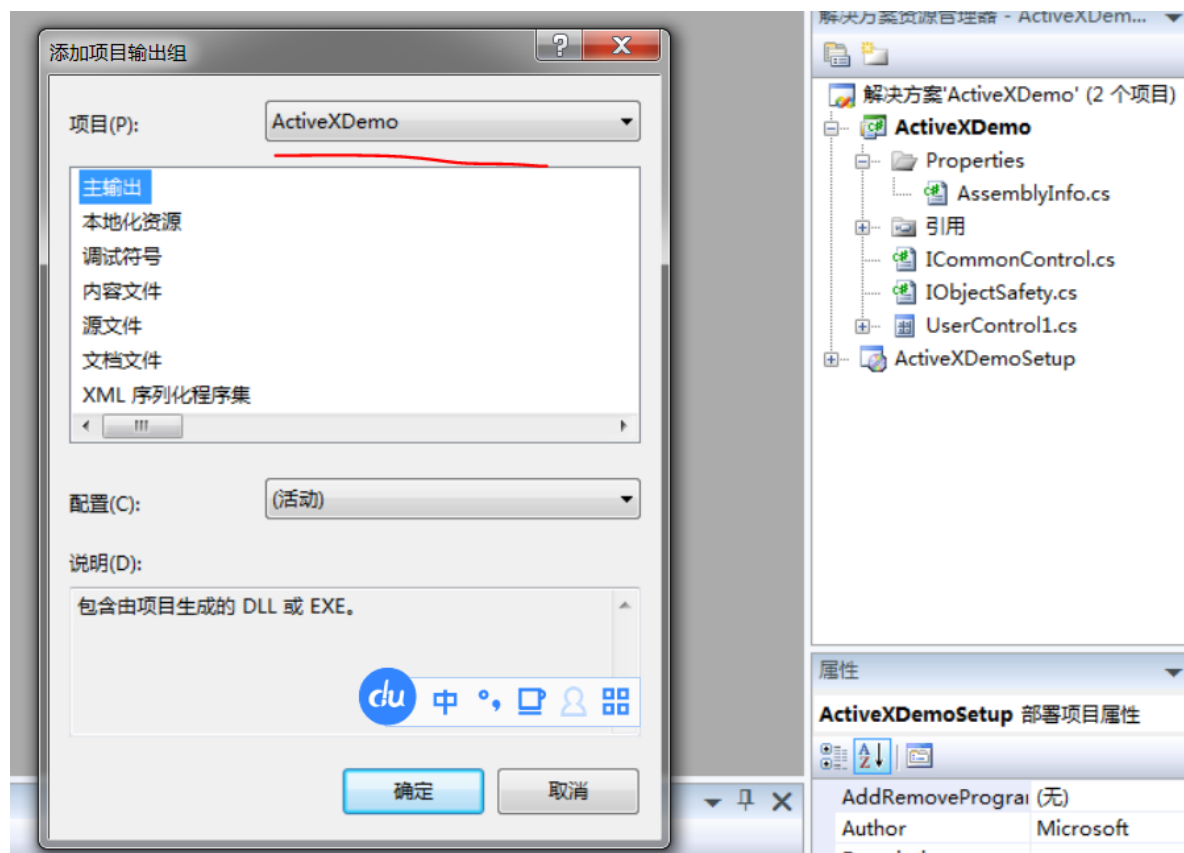
#endregion
}
}

```

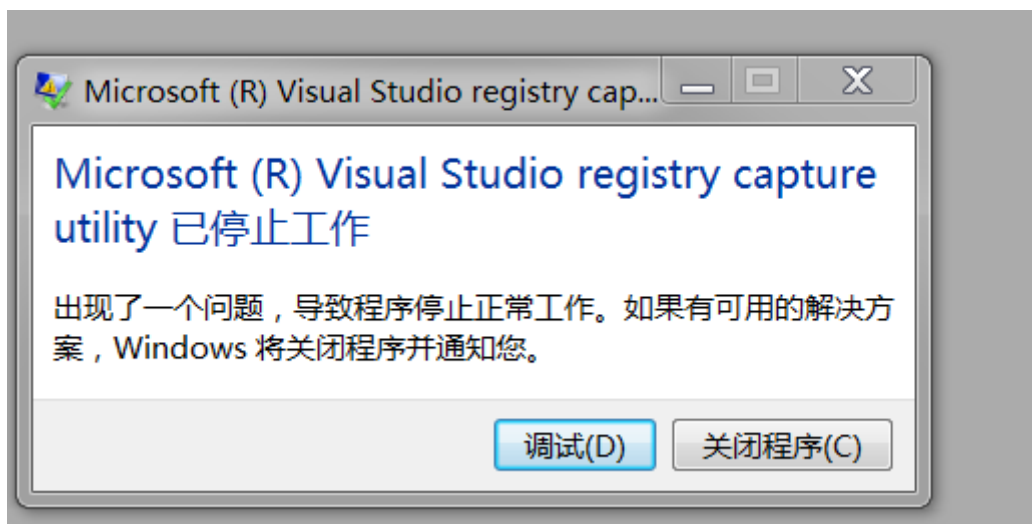
- 在当前解决方案下新建安装项目



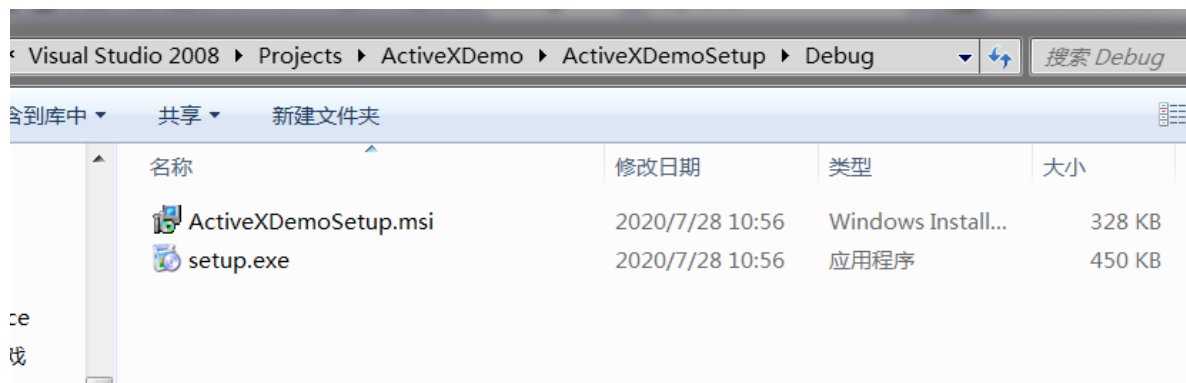
- 右键安装项目->添加项目输出组



- 会出现一个问题，不用在意



- 到安装项目的Debug下，会产生msi和exe程序



- 安装成功后，新建一个web应用程序验证

```
<body>
  <script type="text/javascript">
    var aa = new ActiveXObject("ActiveXDemo.UserControl1");
    var r = aa.Test(1,2);
    console.log(r);
  </script>
</body>
```