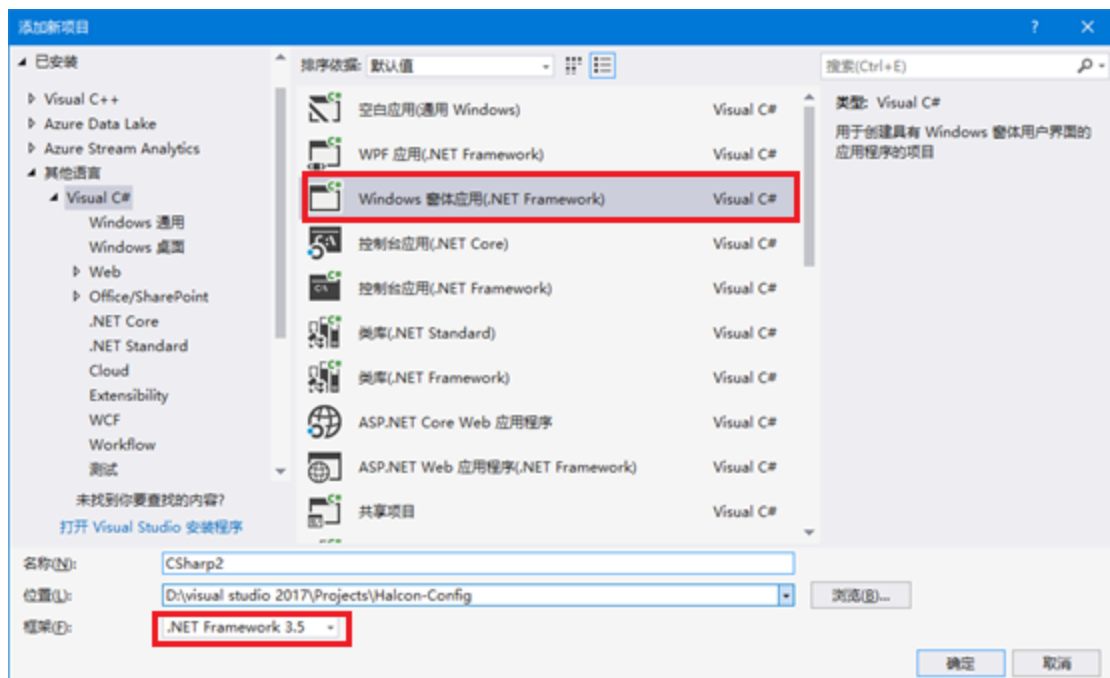


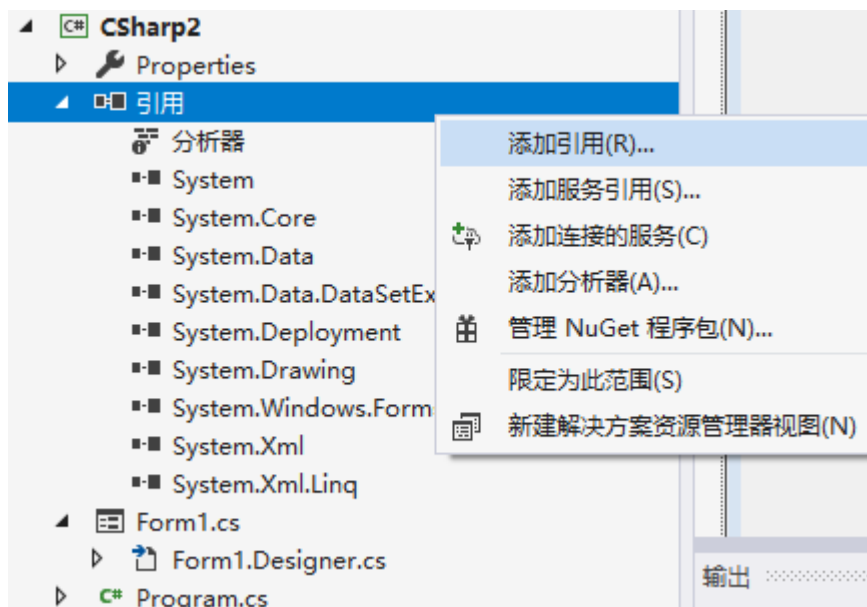
# Halcon 与.NET 混合编程

## 第一种方法(推荐)

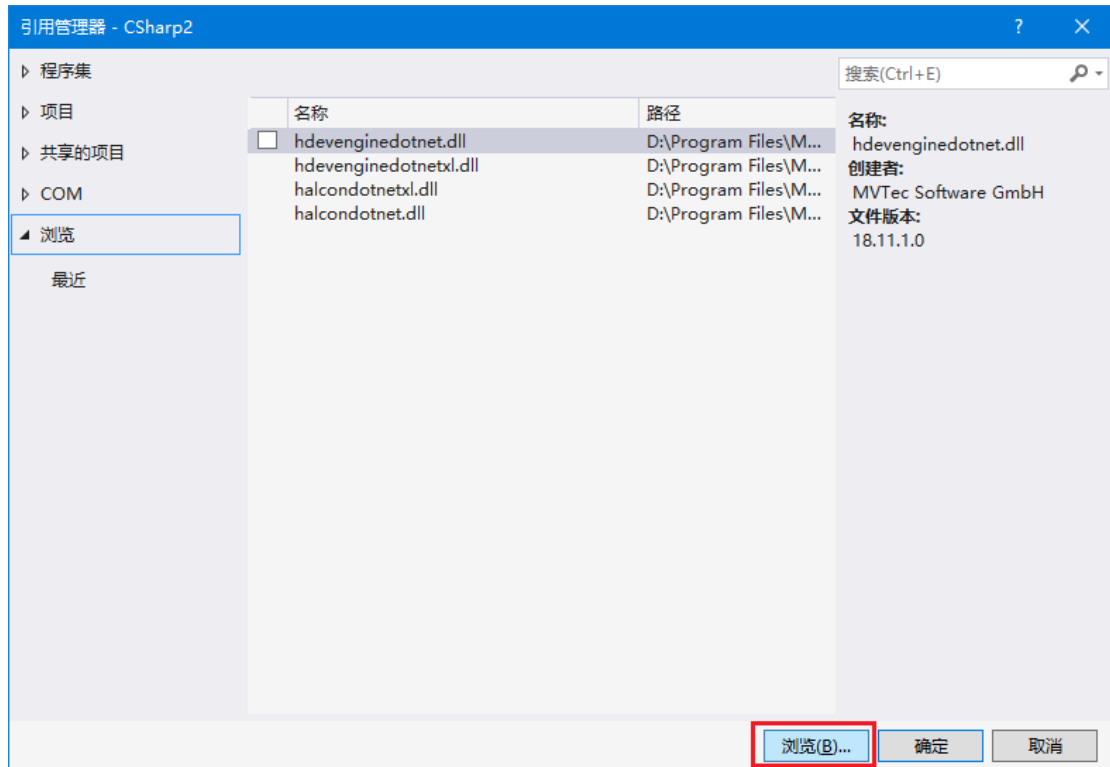
1、新建一个 C#的窗体应用，框架选择.NET Framework 3.5



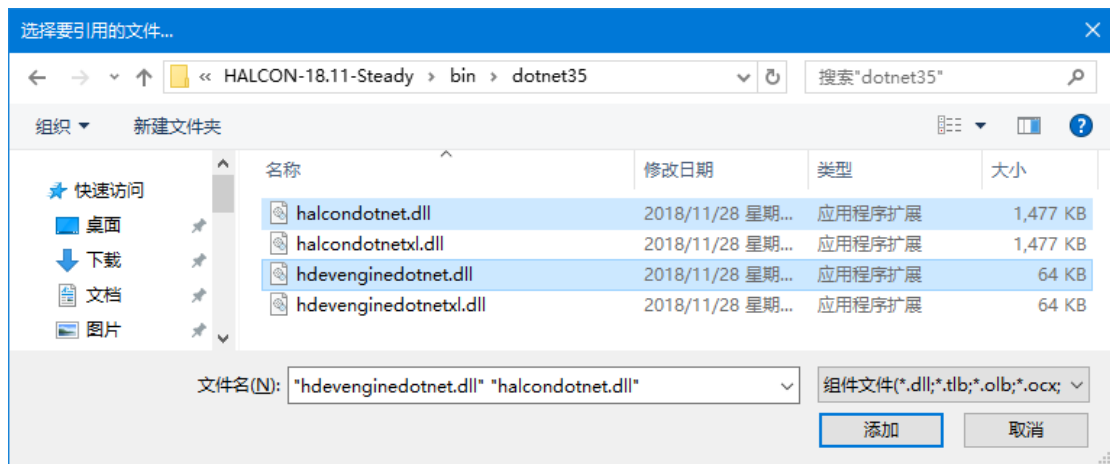
2、右键单击“引用”，然后点击“添加引用”



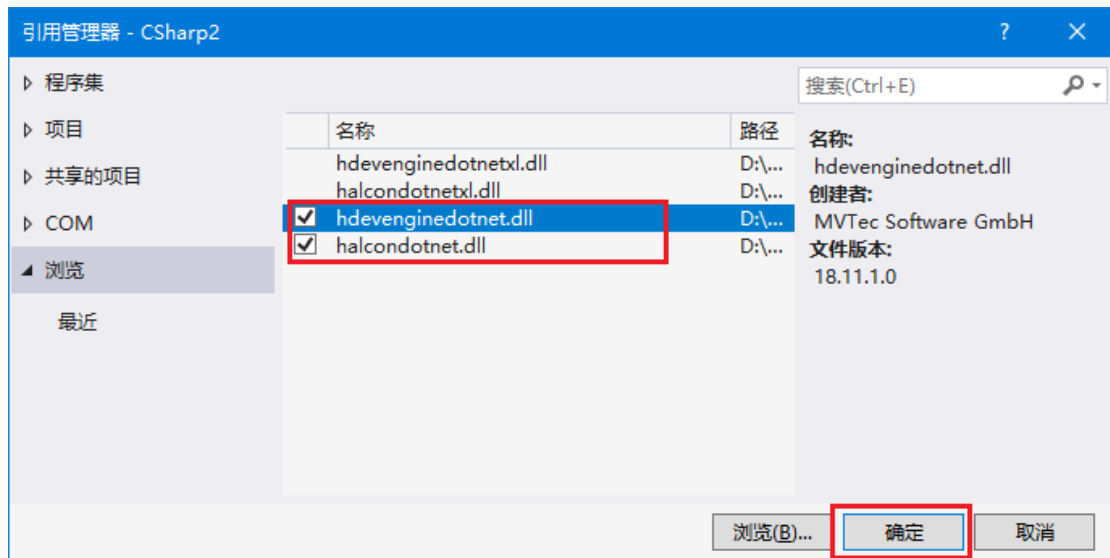
3、点击“浏览”，找到要添加的引用所在的路径



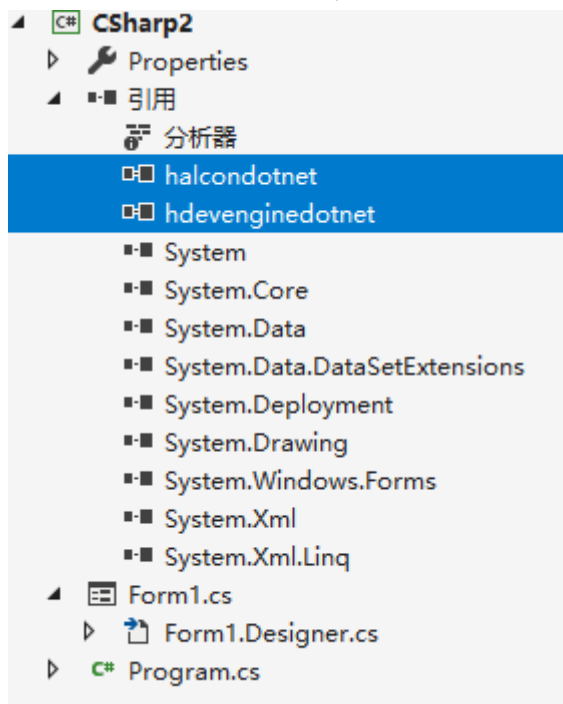
4、从 Halcon 安装路径下 bin 文件夹中的 dotnet35 文件夹中选择“halcondotnet.dll”和“hdevenginedotnet.dll”两个文件，然后点击“添加”



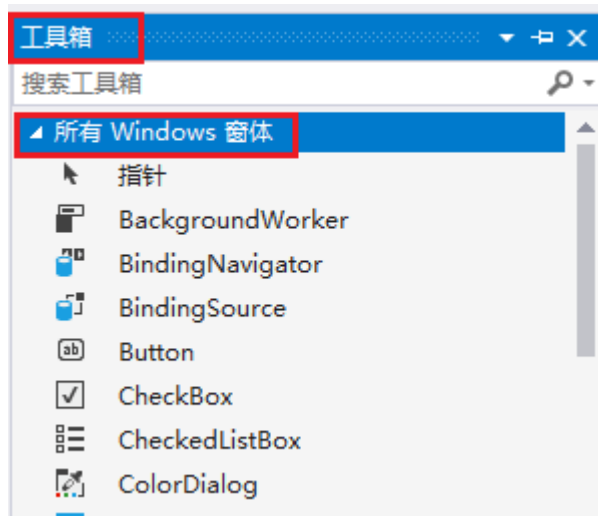
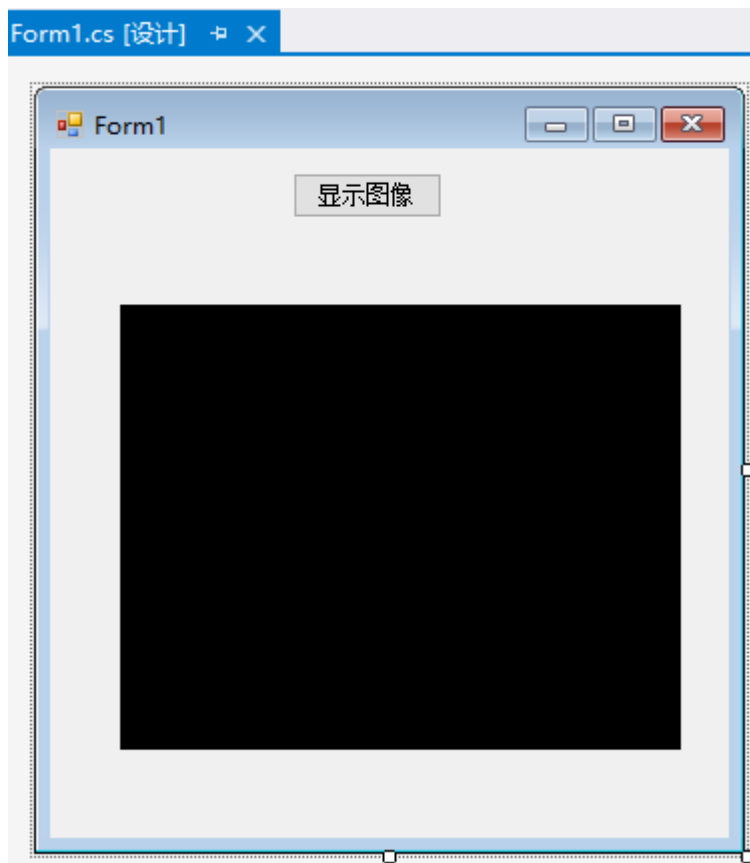
5、第 4 步中点击“添加”后界面如下所示，点击“确定”



6、添加引用后项目中会有下面所示的两个引用



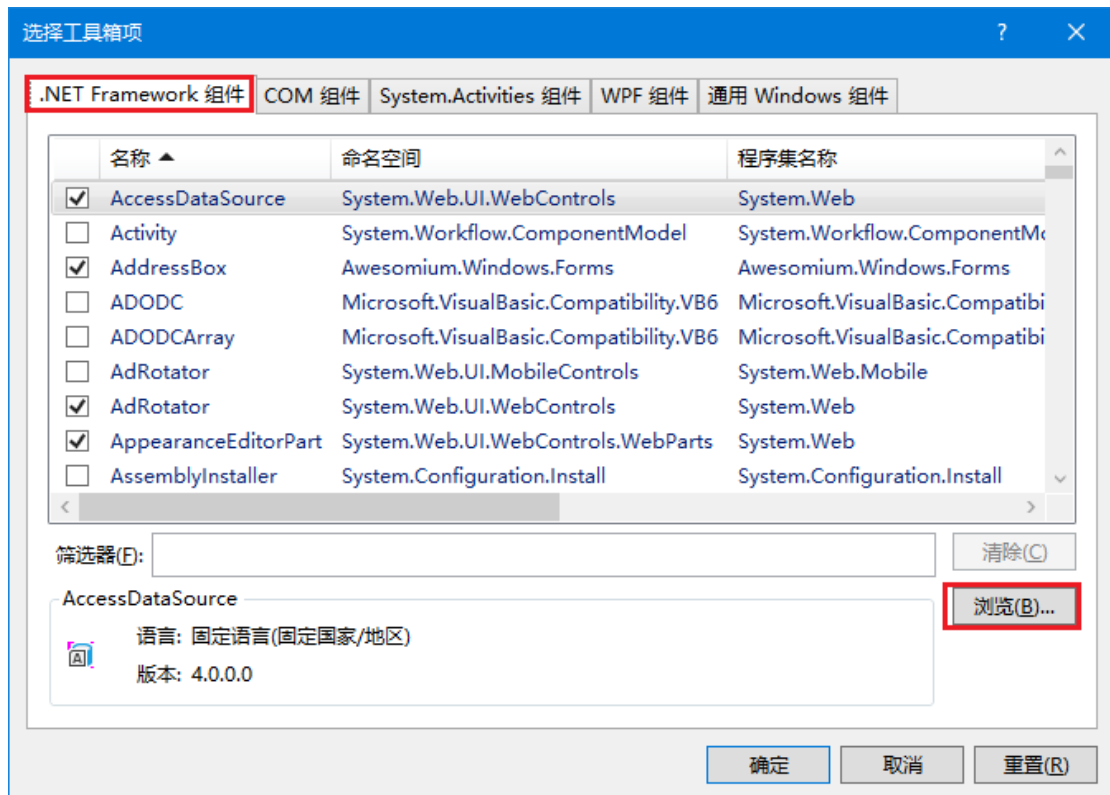
7、切换到 Form1.cs[设计]界面，点击 Form1 窗体后点击“工具箱”



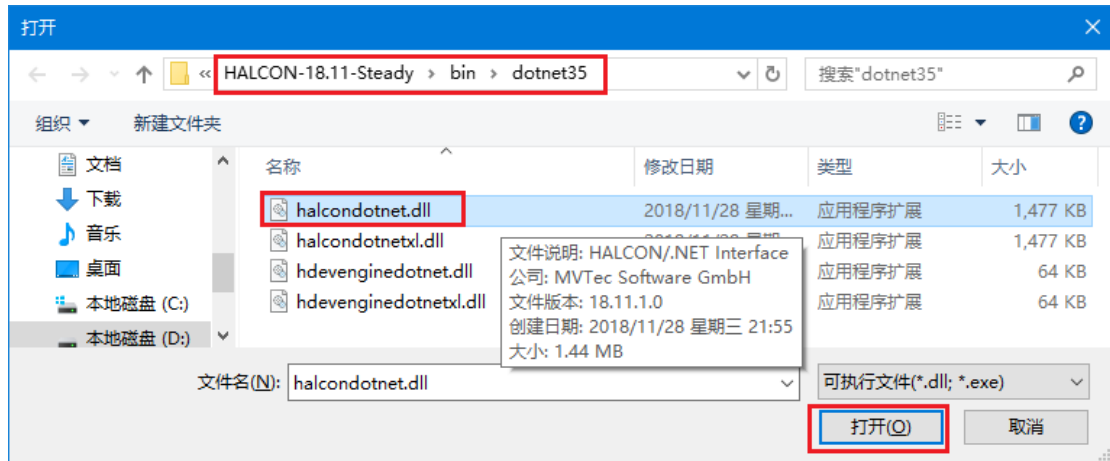
8、右键单击“所有 Windows 窗体”，然后单击“选择项(I)”



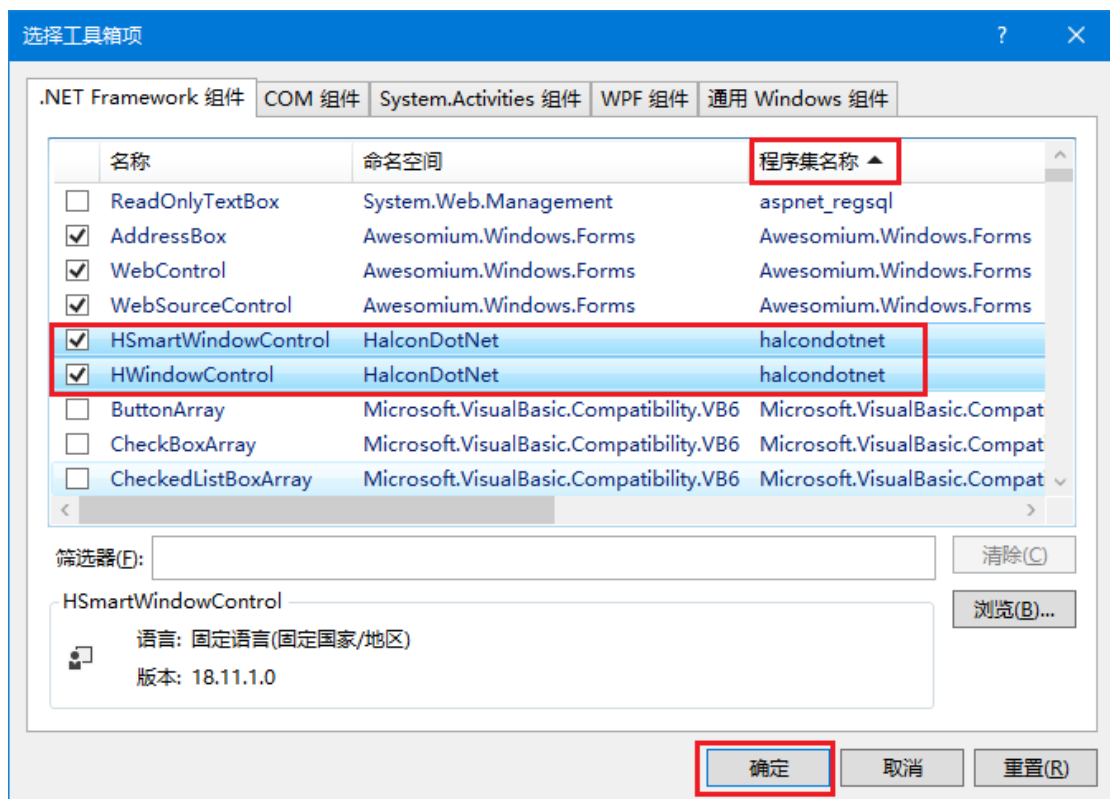
9、在弹出如下界面后，点击“浏览”



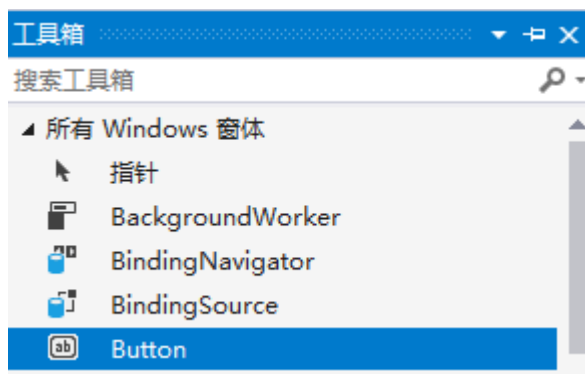
10、找到 Halcon 安装路径下 bin 文件夹中的 dotnet35 文件夹中的“halcondotnet.dll”文件，然后点击“打开”

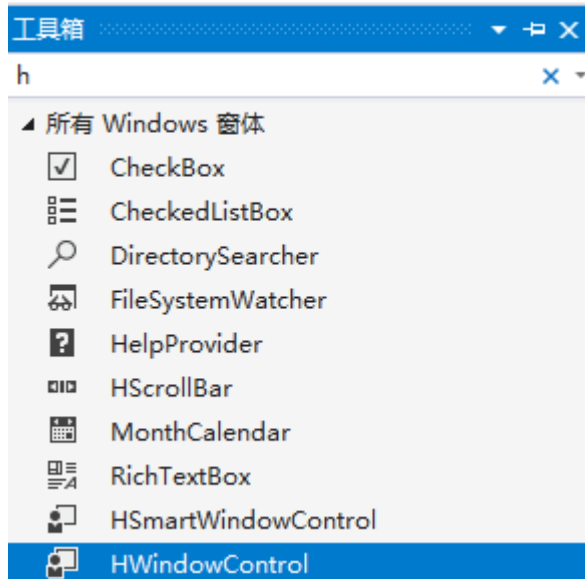
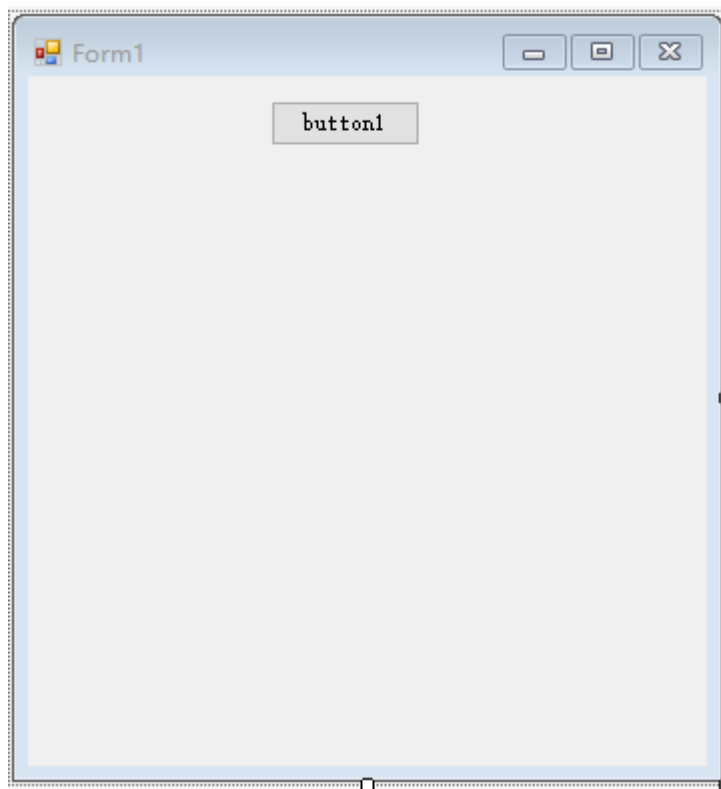


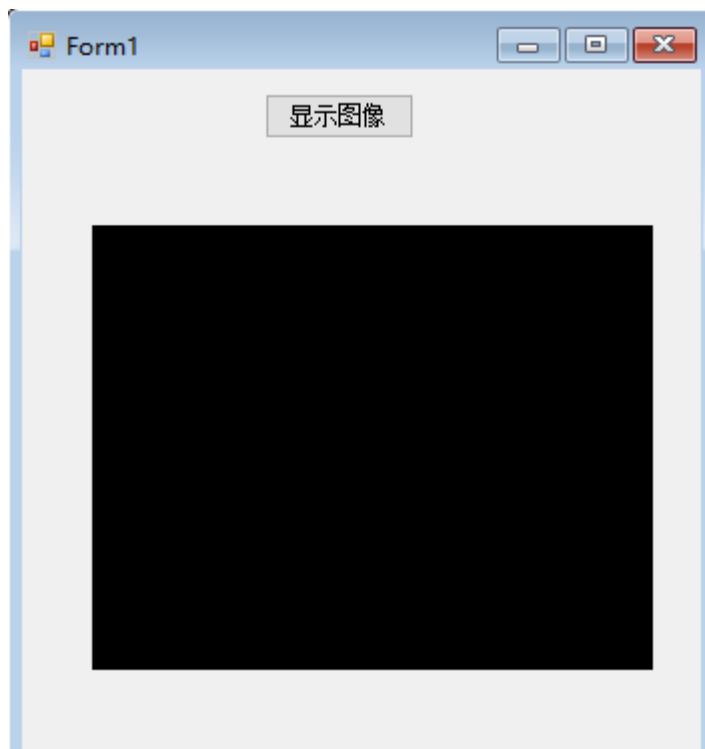
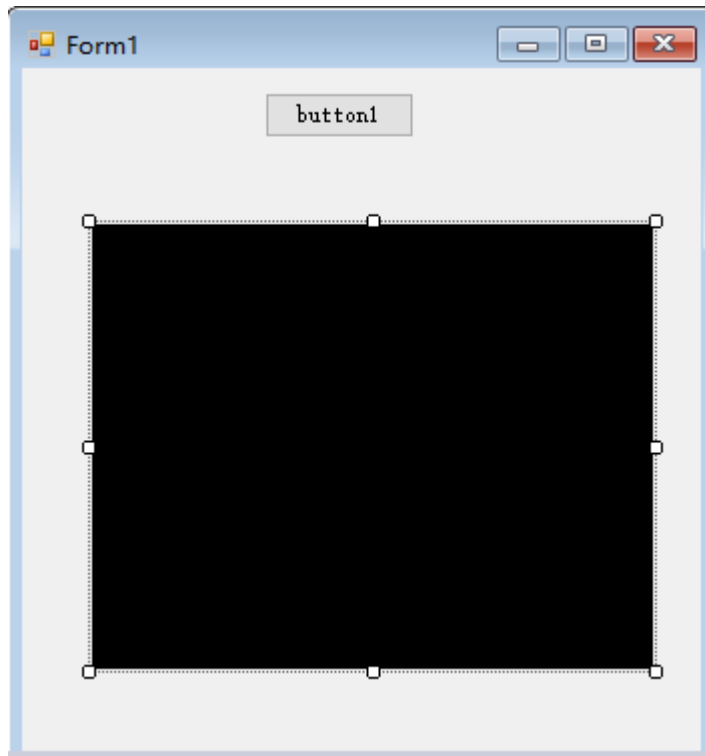
11、在下面的界面中可以看到添加的新的控件，然后点击“确定”



12、在 Form1 窗体中添加一个“Button”控件和一个“HWindowControl”控件，并将“Button”控件的显示文本改成“显示图像”








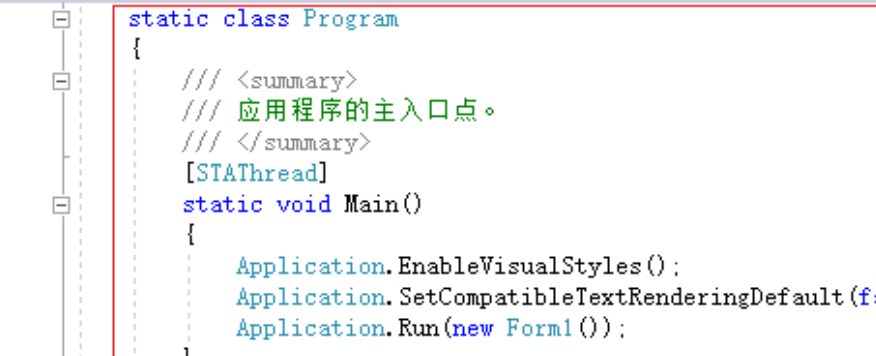
13、在 Program.cs 文件中添加 `using HalconDotNet;`





```
1 using System;
2 using System.Collections.Generic;
3 using System.Linq;
4 using System.Windows.Forms;
5
6 // 第一步
7 using HalconDotNet;
```

14、在 Program.cs 文件中添加 Halcon 导出的 C#代码中的部分代码，如下图所示



```
Program.cs x Form1.cs [设计]
C# CSharp2
static class Program
{
    /// <summary>
    /// 应用程序的主入口点。
    /// </summary>
    [STAThread]
    static void Main()
    {
        Application.EnableVisualStyles();
        Application.SetCompatibleTextRenderingDefault(false);
        Application.Run(new Form1());
    }
}

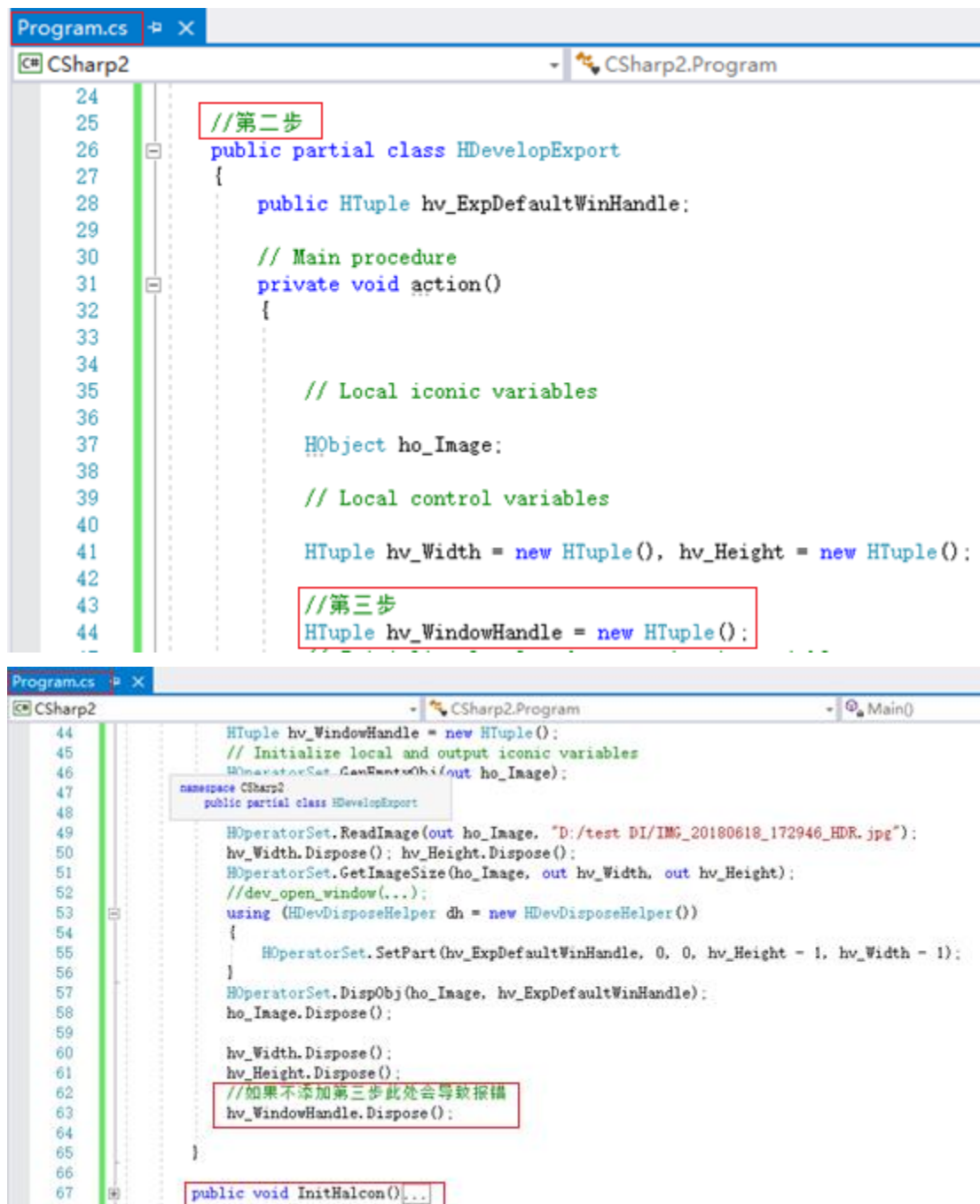
//第二步
public partial class HDevelopExport
{
    public HTuple hv_ExpDefaultWinHandle;

    // Main procedure
    private void action()...

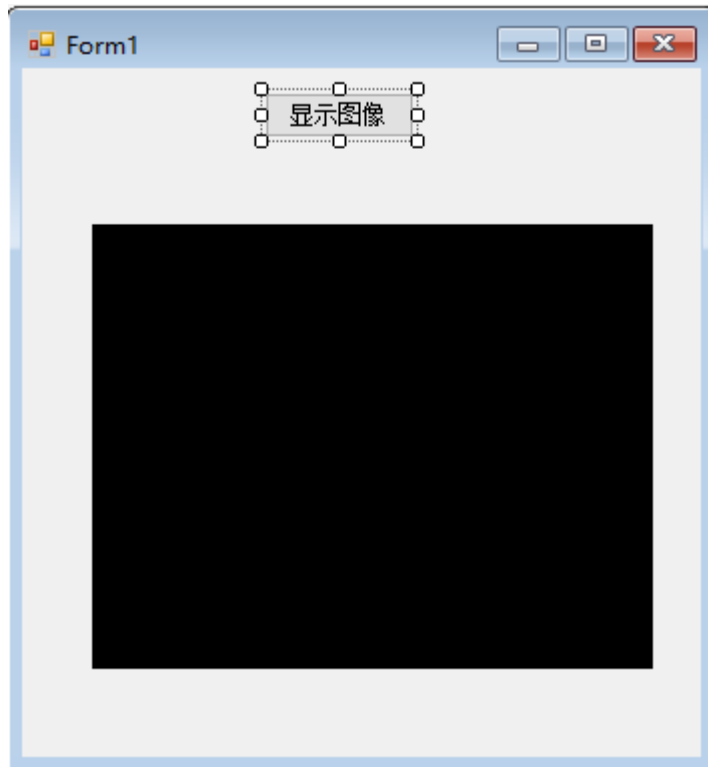
    public void InitHalcon()...

    public void RunHalcon(HTuple Window)...
}
```

15、在 Halcon 导出的代码中添加一行代码 `HTuple hv_WindowHandle = new HTuple();`，否则会有一处错误，个人认为此为 Halcon 的 bug，后续版本也许会修复

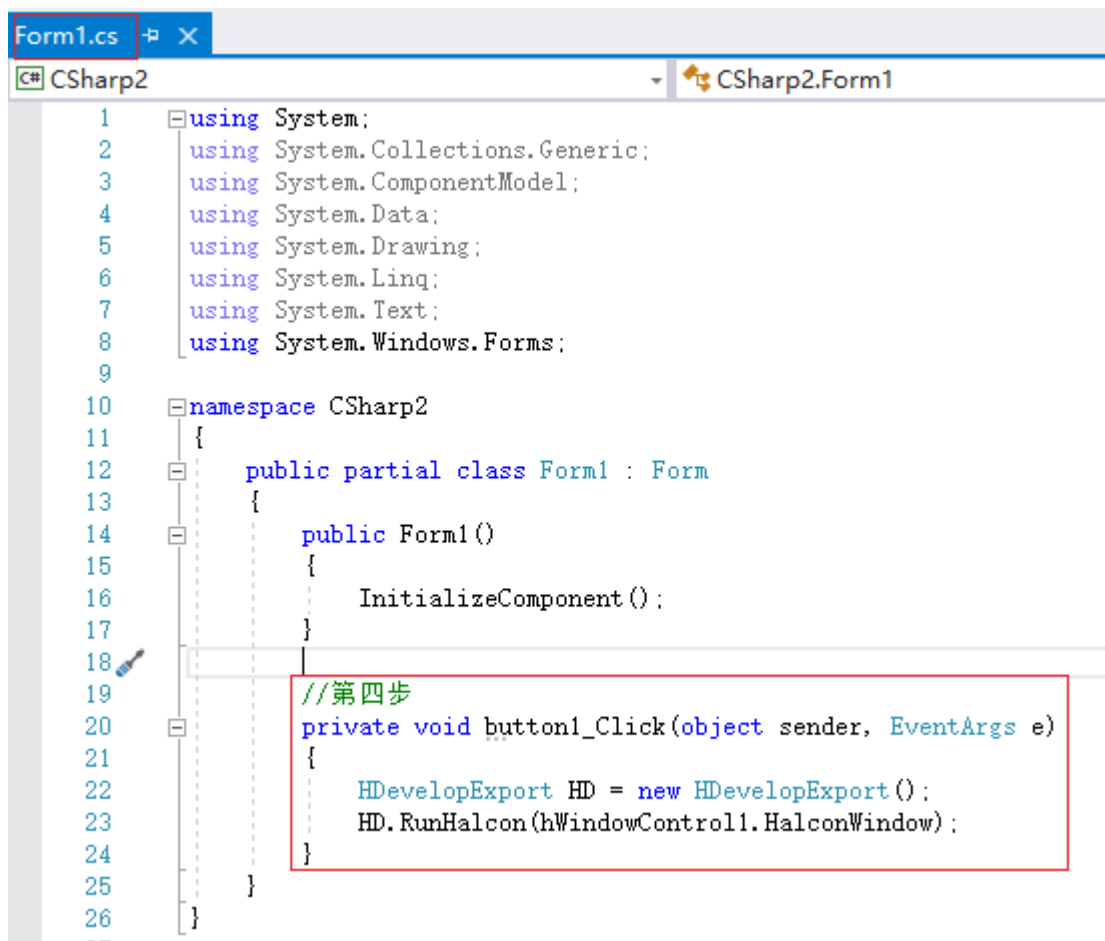


16、双击 Form1 中的“显示图像”



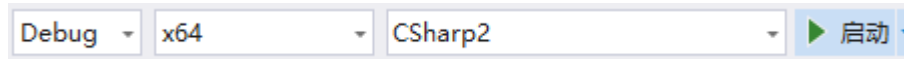
17、在button的Click事件中添加代码

```
HDevelopExport HD = new HDevelopExport();  
HD.RunHalcon(hWindowControll.HalconWindow);
```

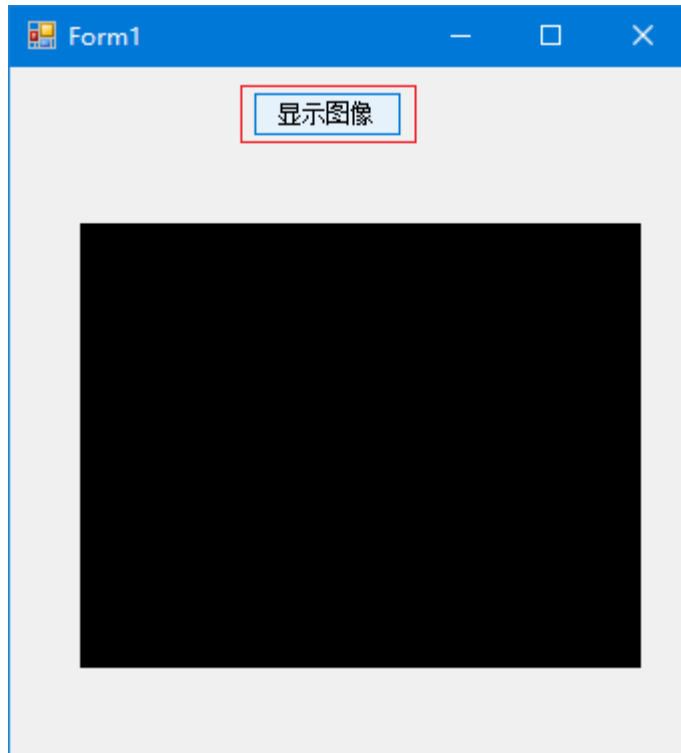


```
1  using System;
2  using System.Collections.Generic;
3  using System.ComponentModel;
4  using System.Data;
5  using System.Drawing;
6  using System.Linq;
7  using System.Text;
8  using System.Windows.Forms;
9
10 namespace CSharp2
11 {
12     public partial class Form1 : Form
13     {
14         public Form1()
15         {
16             InitializeComponent();
17         }
18
19         //第四步
20         private void button1_Click(object sender, EventArgs e)
21         {
22             HDevelopExport HD = new HDevelopExport();
23             HD.RunHalcon(hWindowControl1.HalconWindow);
24         }
25     }
26 }
```

18、由于本人安装的 Halcon 只有 x64 版本，因此此处选择 x64 版本，如果是 x86 则选择 x86，如果两者都安装了则选择 Any CPU，然后点击“启动”



19、运行后界面如下图所示，点击“显示图像”

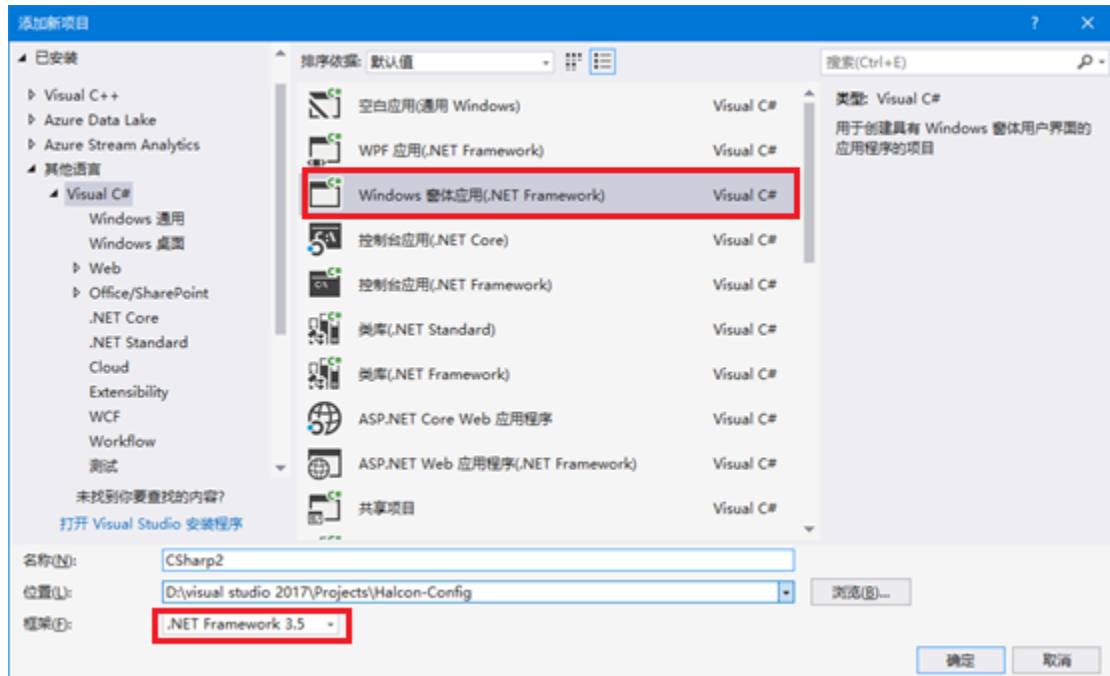


20、如果弹出下面的界面就说明 Halcon 与.NET 的配置成功了，可以进行复杂的程序操作了

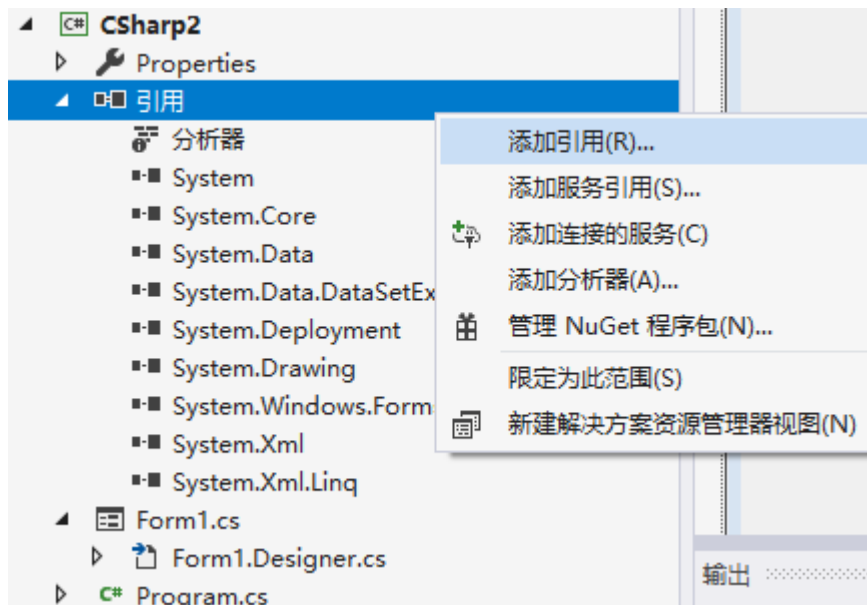


## 第二种方法（不推荐）

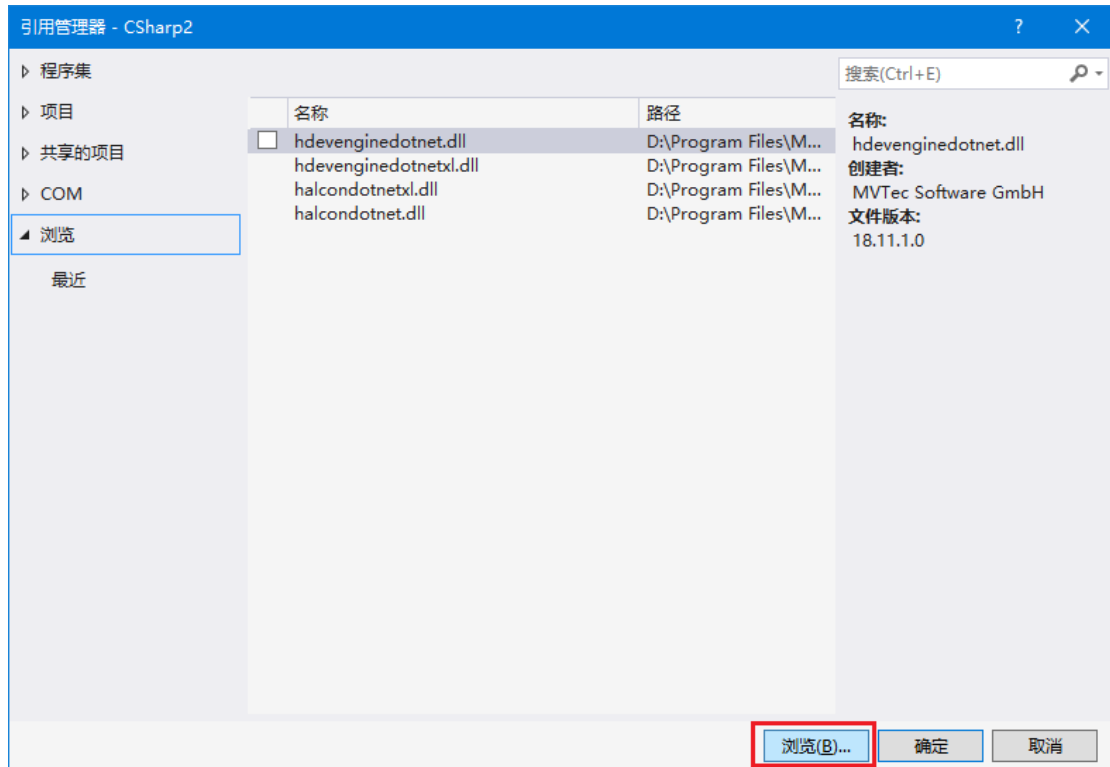
1、新建一个 C# 的窗体应用，框架选择 .NET Framework 3.5



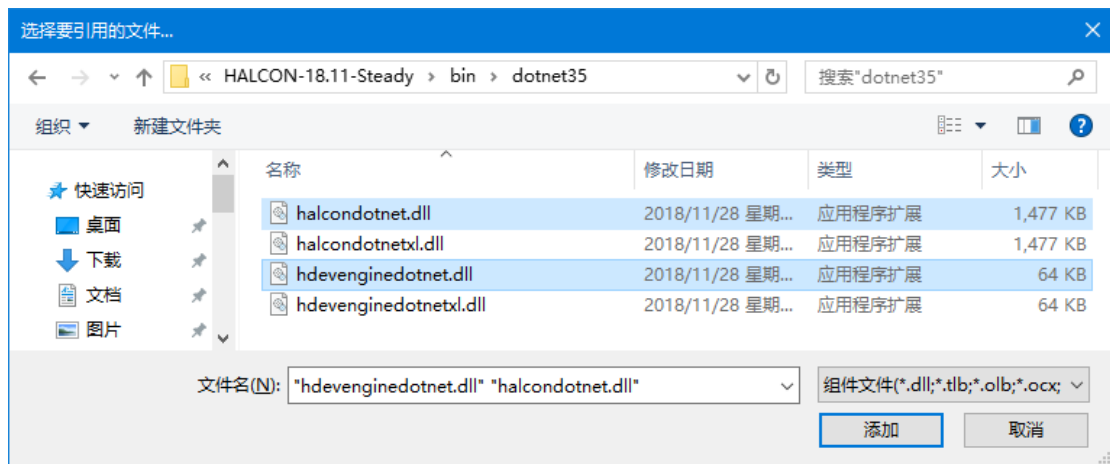
2、右键单击“引用”，然后点击“添加引用”



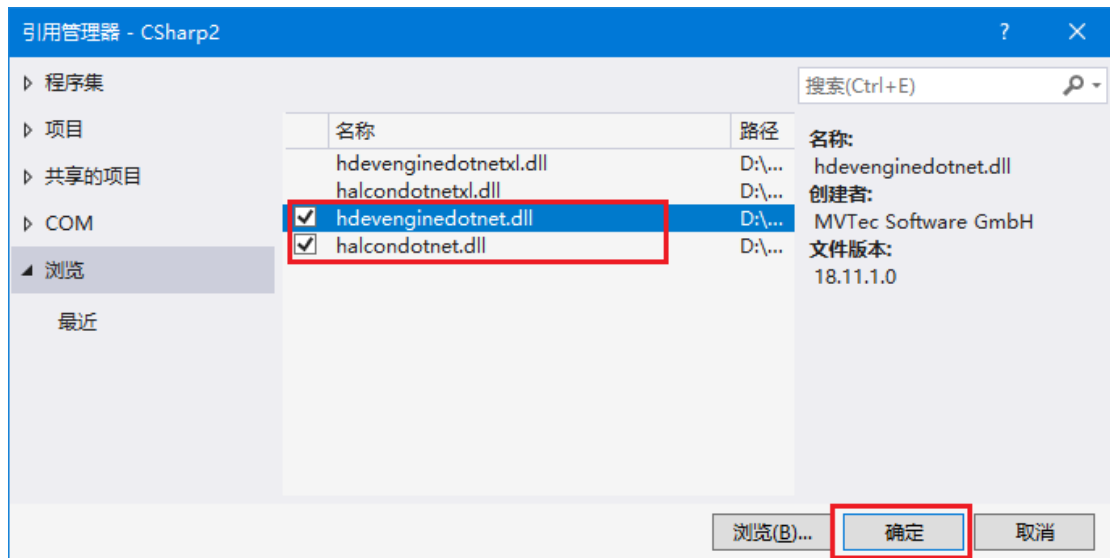
3、点击“浏览”，找到要添加的引用所在的路径



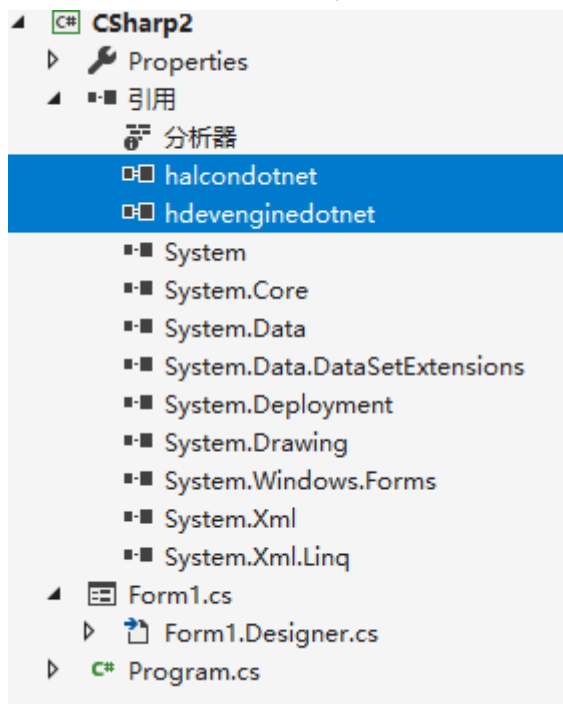
4、从 Halcon 安装路径下 bin 文件夹中的 dotnet35 文件夹中选择“halcondotnet.dll”和“hdevenginedotnet.dll”两个文件，然后点击“添加”



5、第 4 步中点击“添加”后界面如下所示，点击“确定”

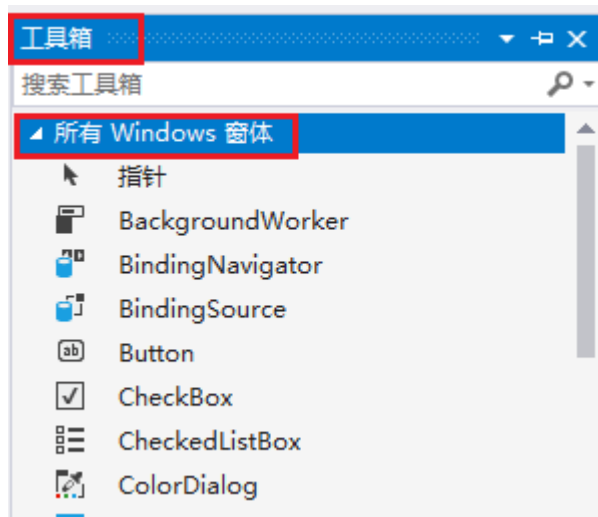


6、添加引用后项目中会有下面所示的两个引用



7、切换到 Form1.cs[设计]界面，点击 Form1 窗体后点击“工具箱”

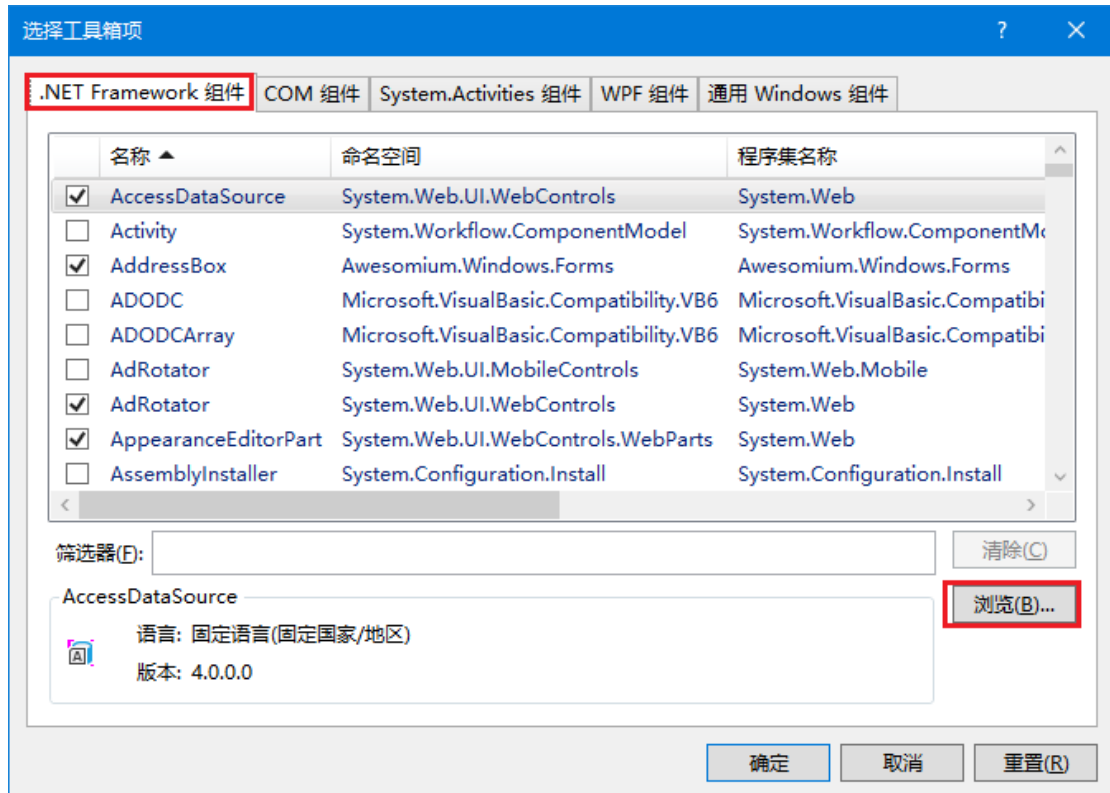




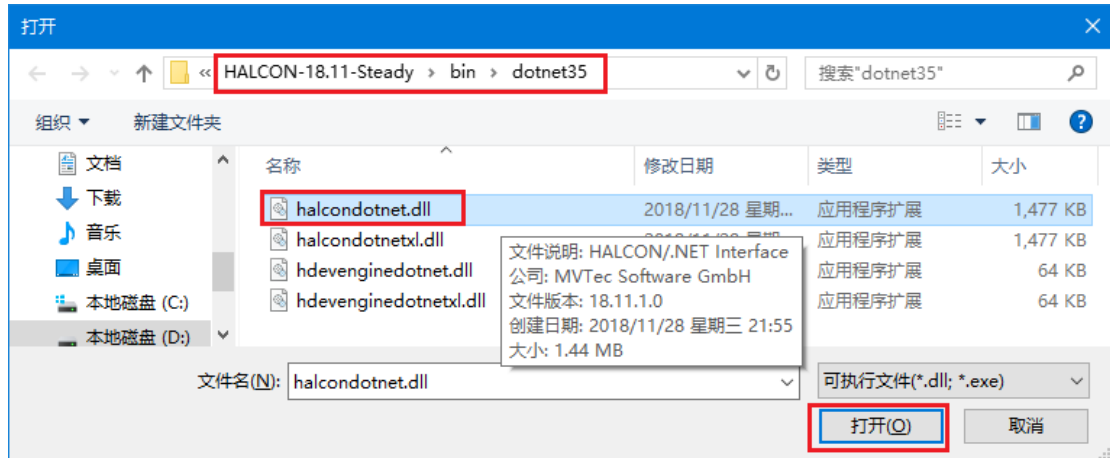
8、右键单击“所有 Windows 窗体”，然后单击“选择项(I)”



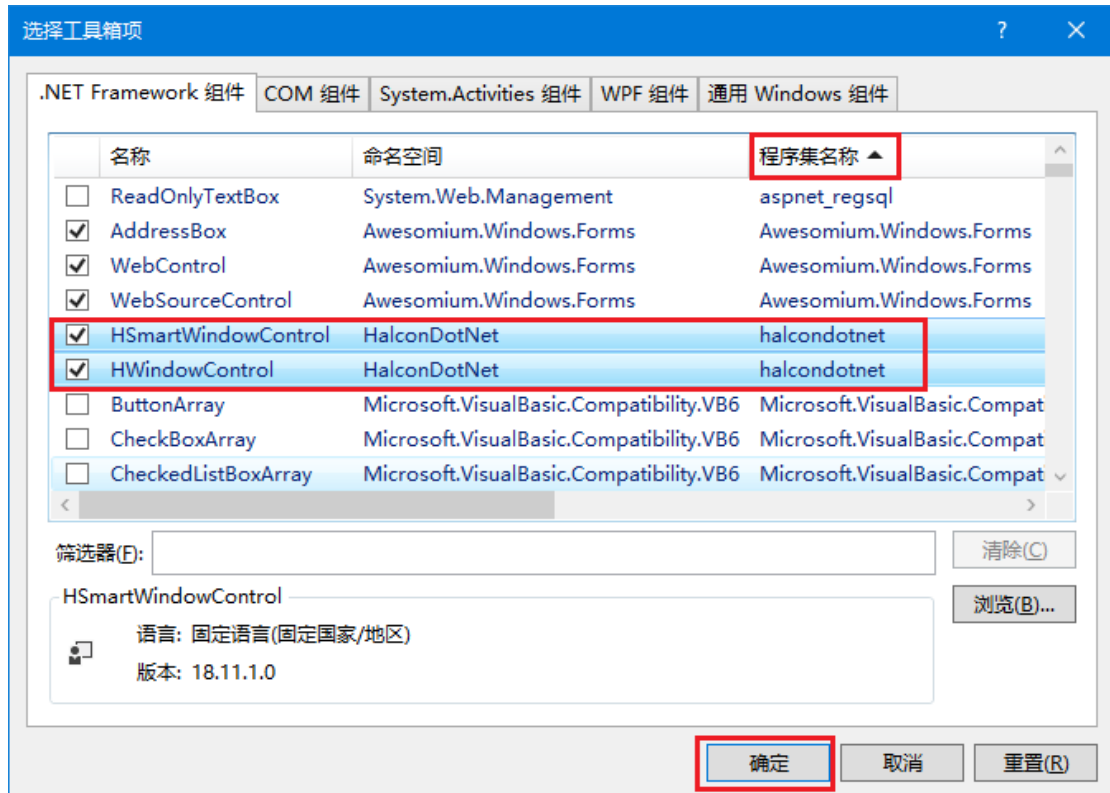
9、在弹出如下界面后，点击“浏览”



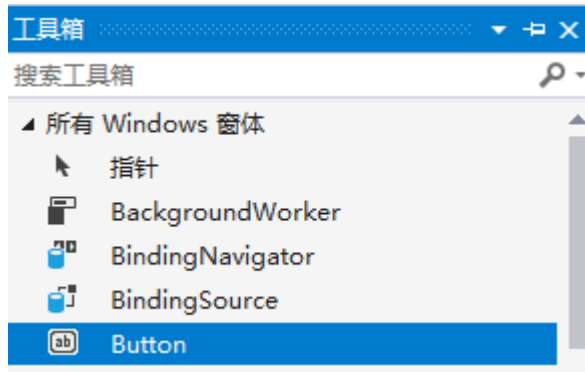
10、找到 Halcon 安装路径下 bin 文件夹中的 dotnet35 文件夹中的“halcondotnet.dll”文件，然后点击“打开”

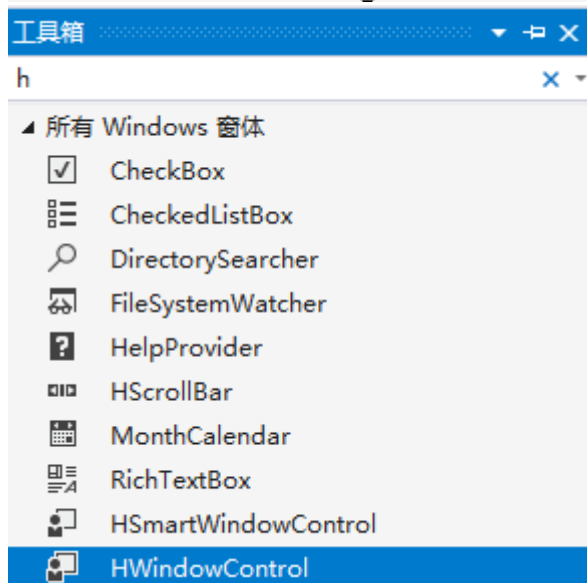
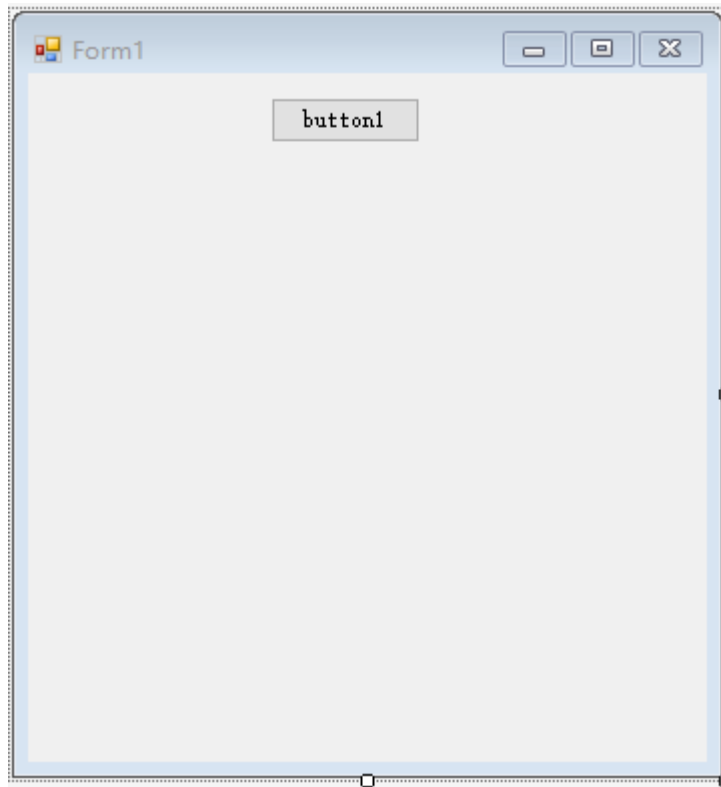


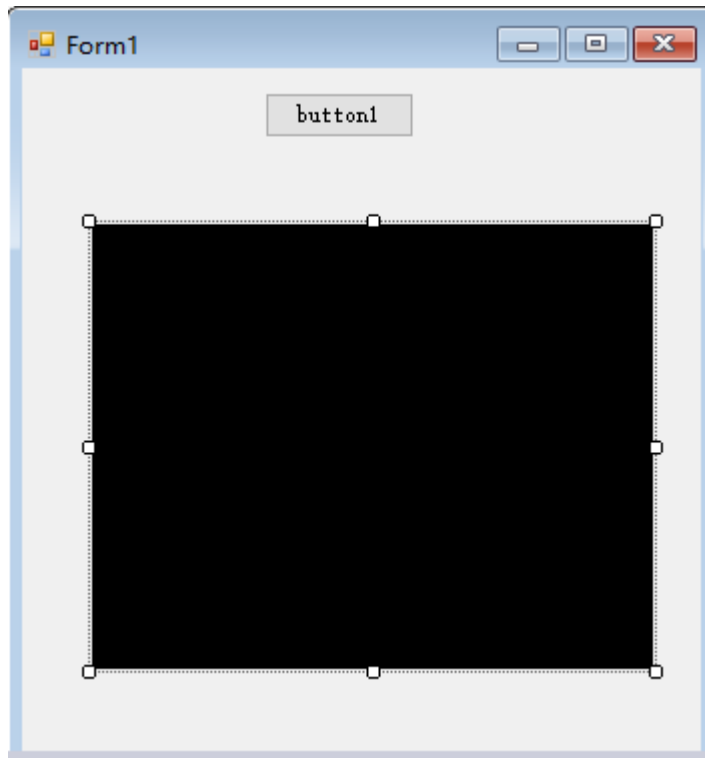
11、在下面的界面中可以看到添加的新的控件，然后点击“确定”



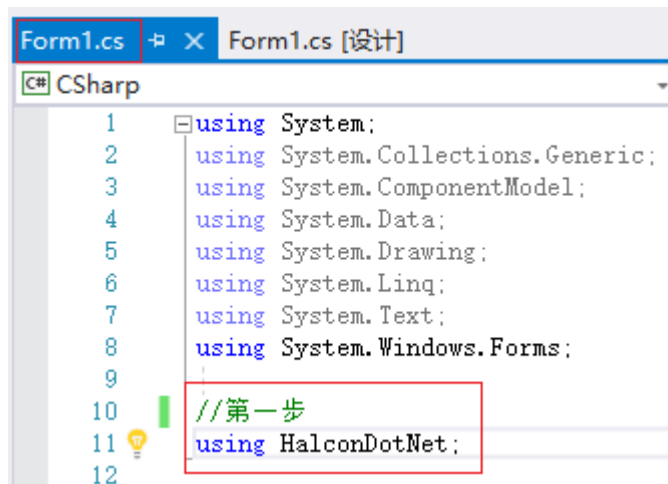
12、在 Form1 窗体中添加一个“Button”控件和一个“HWindowControl”控件，并将“Button”控件的显示文本改成“显示图像”





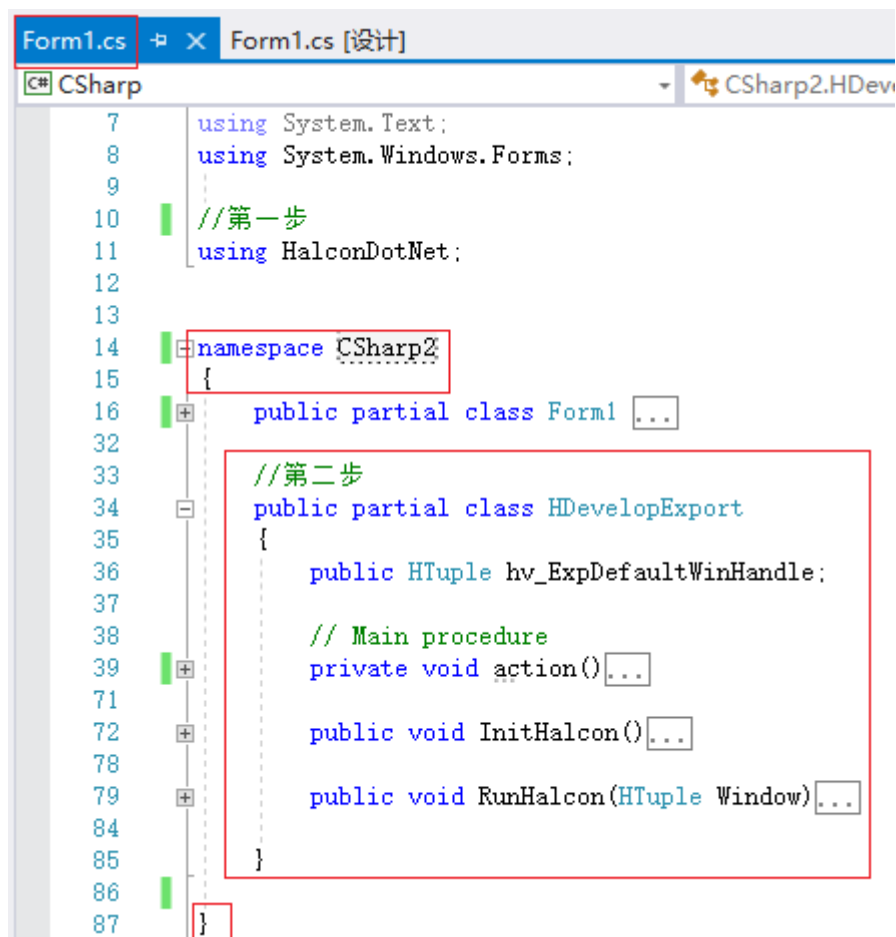


13、双击 Form1 的“显示图像”，进入 Form1.cs，然后添加 `using HalconDotNet;`



```
1 using System;
2 using System.Collections.Generic;
3 using System.ComponentModel;
4 using System.Data;
5 using System.Drawing;
6 using System.Linq;
7 using System.Text;
8 using System.Windows.Forms;
9
10 //第一步
11 using HalconDotNet;
12
```

14、在 Form1.cs 中的 `public partial class Form1` 下面（此处是看到论坛中某网友说必须在此下面，否则会出错，未进行验证）添加 Halcon 导出的 C# 代码，如下图所示



```
7 using System.Text;
8 using System.Windows.Forms;
9
10 //第一步
11 using HalconDotNet;
12
13
14 namespace CSharp2
15 {
16     public partial class Form1 ...
17
18     //第二步
19     public partial class HDevelopExport
20     {
21         public HTuple hv_ExpDefaultWinHandle;
22
23         // Main procedure
24         private void action() ...
25
26         public void InitHalcon() ...
27
28         public void RunHalcon(HTuple Window) ...
29     }
30
31 }
32
```

15、在 Halcon 导出的代码中添加代码 `hv_WindowHandle = new HTuple()`，否则会有一处错误，个人认为此为 Halcon 的 bug，后续版本也许会修复

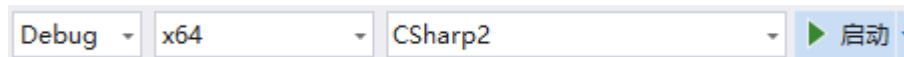
```
46
47
48 //第三步
49 HTuple hv_Width = new HTuple(), hv_Height = new HTuple(), hv_WindowHandle = new HTuple();
50 // Initialize local and output iconic variables
51 HOperatorSet.GenEmptyObj(out ho_Image);
52 //dev_close_window(...);
53 ho_Image.Dispose();
54 HOperatorSet.ReadImage(out ho_Image, "D:/Visual studio 2017/Projects/Halcon-Config/res/IMG_2
55 hv_Width.Dispose(); hv_Height.Dispose();
56 HOperatorSet.GetImageSize(ho_Image, out hv_Width, out hv_Height);
57 //dev_open_window(...);
58 using (HDevDisposeHelper dh = new HDevDisposeHelper())
59 {
60     HOperatorSet.SetPart(hv_ExpDefaultWinHandle, 0, 0, hv_Height - 1, hv_Width - 1);
61 }
62 HOperatorSet.DispObj(ho_Image, hv_ExpDefaultWinHandle);
63 ho_Image.Dispose();
64
65 hv_Width.Dispose();
66 hv_Height.Dispose();
67 //如果第三步不做修改此处会报错
68 hv_WindowHandle.Dispose();
69
70 }
71
72 public void InitHalcon()
73 {
74     // Default settings used in HDevelop
```

16、在Form1.cs中button的Click事件中添加代码，此处Click事件由于上面第13步的操作而产生

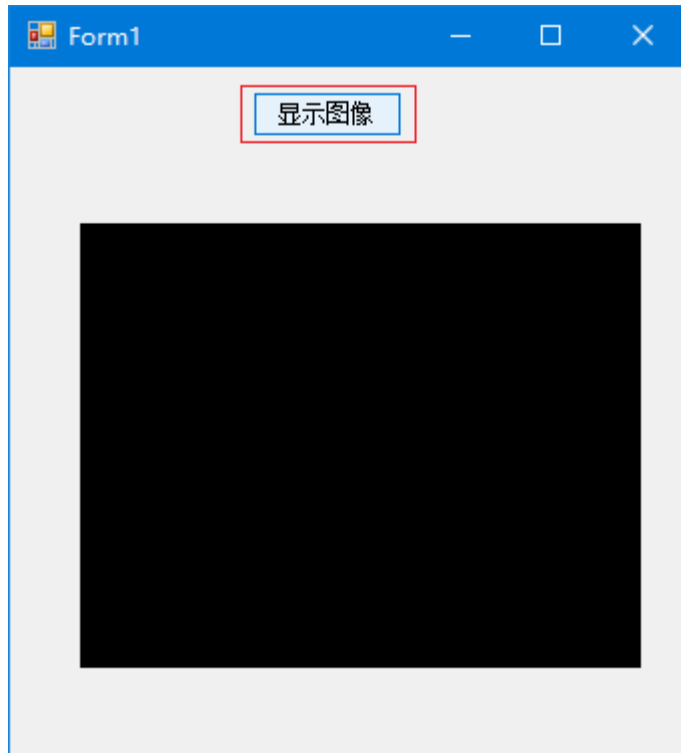
```
HDevelopExport HD = new HDevelopExport();
HD.RunHalcon(hWindowControl1.HalconWindow);
```

```
15 {
16
17 public partial class Form1 : Form
18 {
19     public Form1()
20     {
21         InitializeComponent();
22     }
23
24 //显示图片
25 private void button1_Click(object sender, EventArgs e)
26 {
27     //第四步
28     HDevelopExport HD = new HDevelopExport();
29     HD.RunHalcon(hWindowControl1.HalconWindow);
30 }
31 }
```

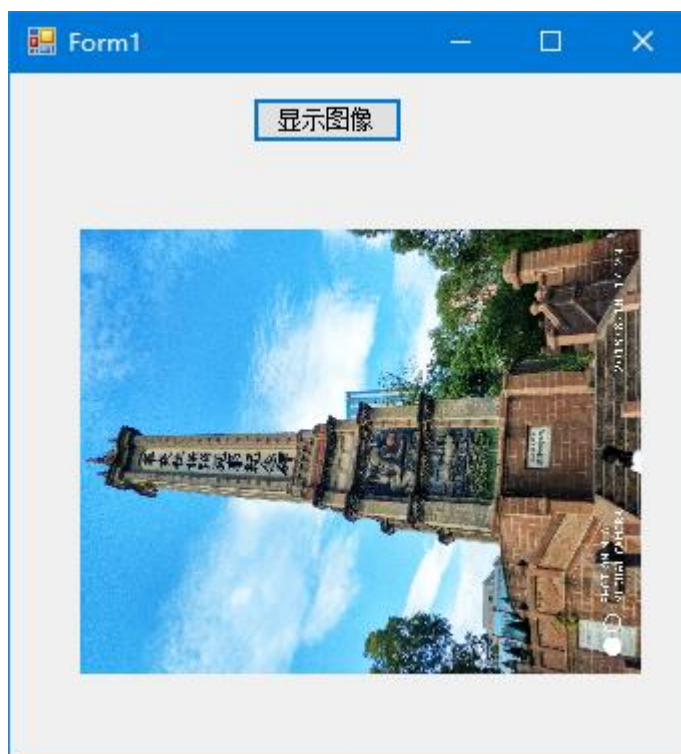
17、由于本人安装的 Halcon 只有 x64 版本，因此此处选择 x64 版本，如果是 x86 则选择 x86，如果两者都安装了则选择 Any CPU，然后点击“启动”



18、运行后界面如下图所示，点击“显示图像”



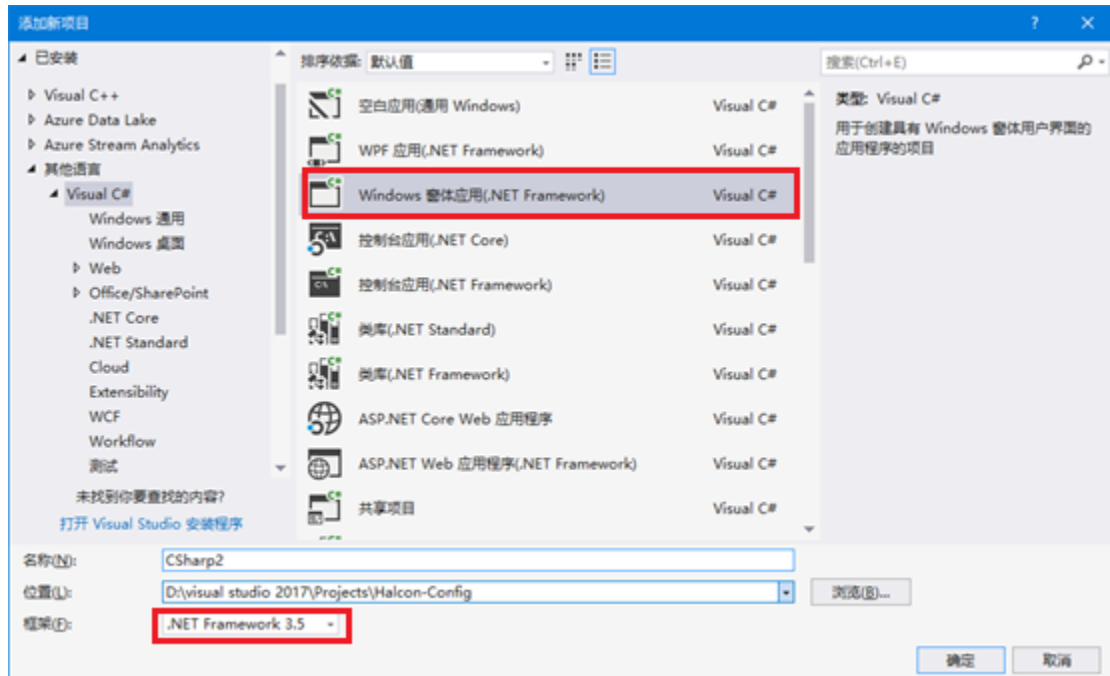
19、如果弹出下面的界面就说明 Halcon 与.NET 的配置成功了，可以进行复杂的程序操作了



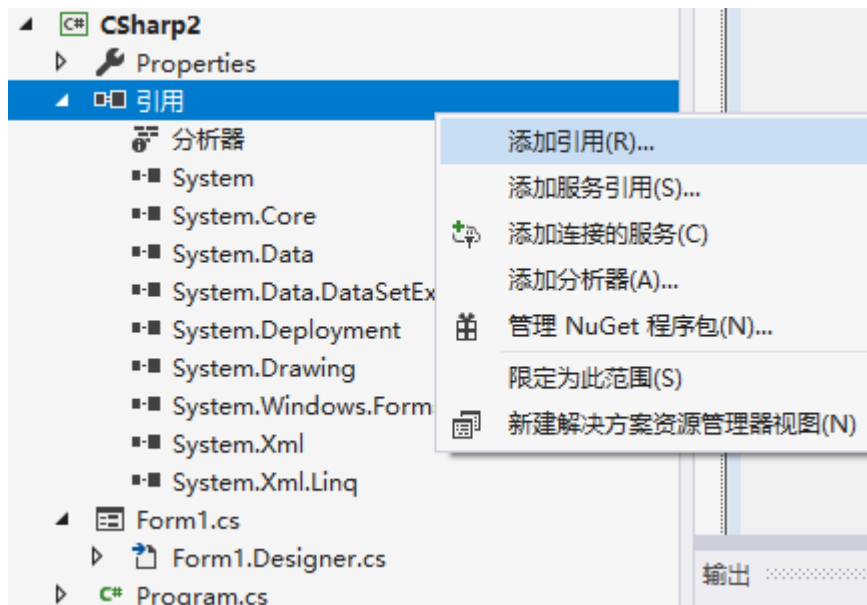


## 第三种方法(步骤繁琐)

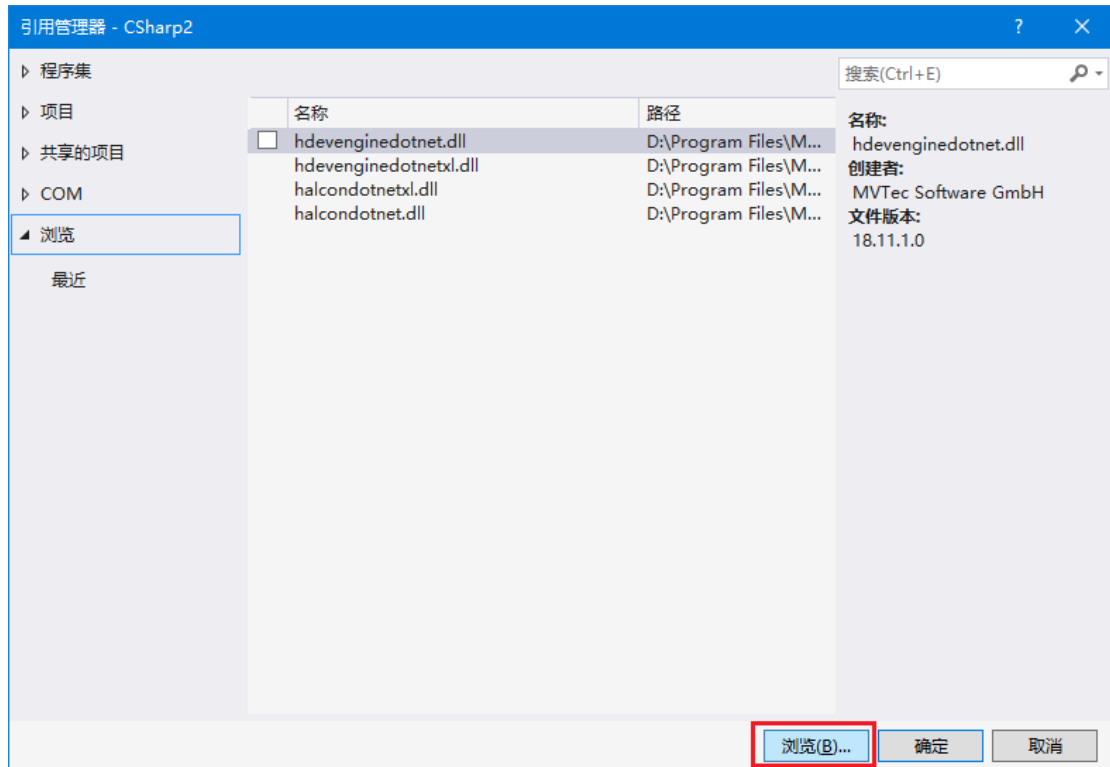
1、新建一个 C#的窗体应用，框架选择.NET Framework 3.5



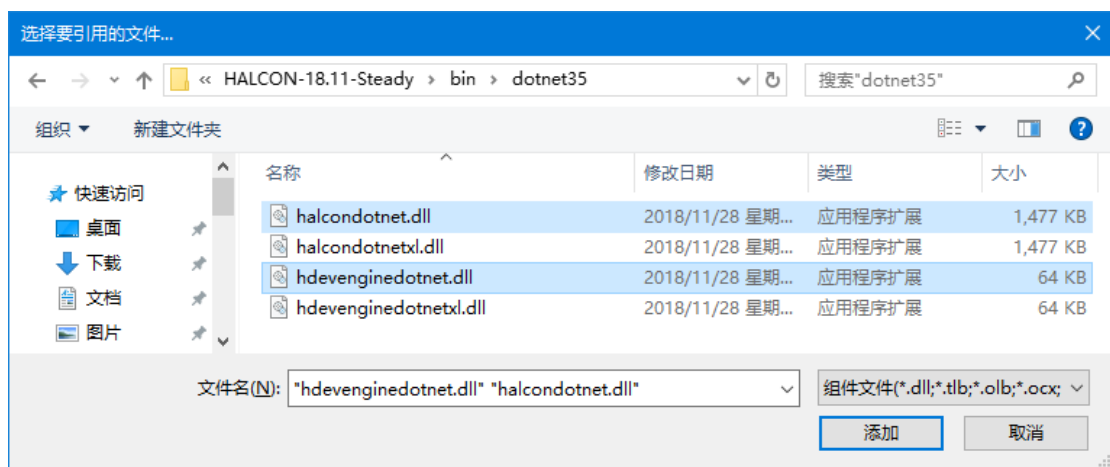
2、右键单击“引用”，然后单击“添加引用”



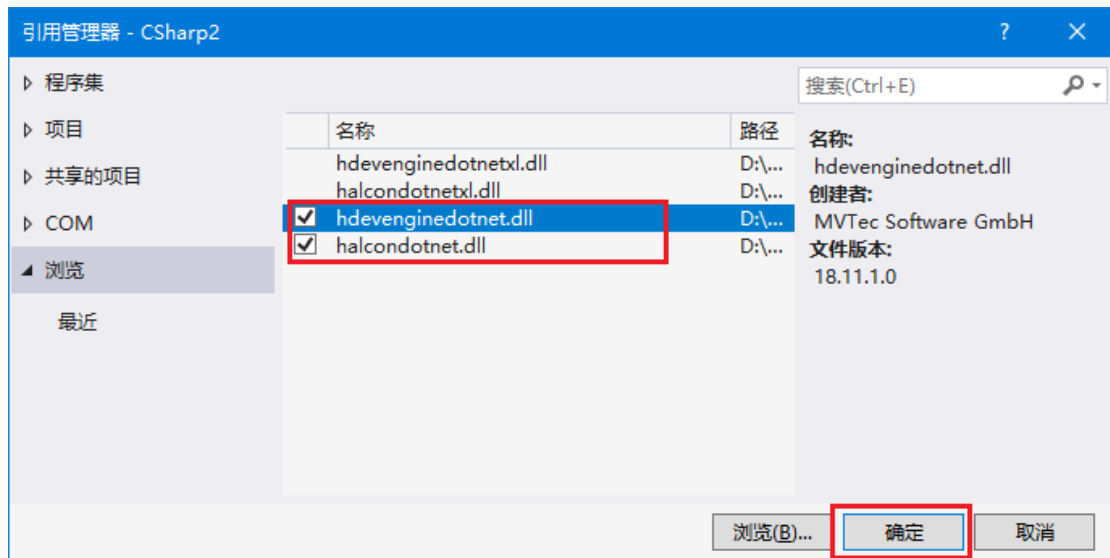
3、点击“浏览”，找到要添加的引用所在的路径



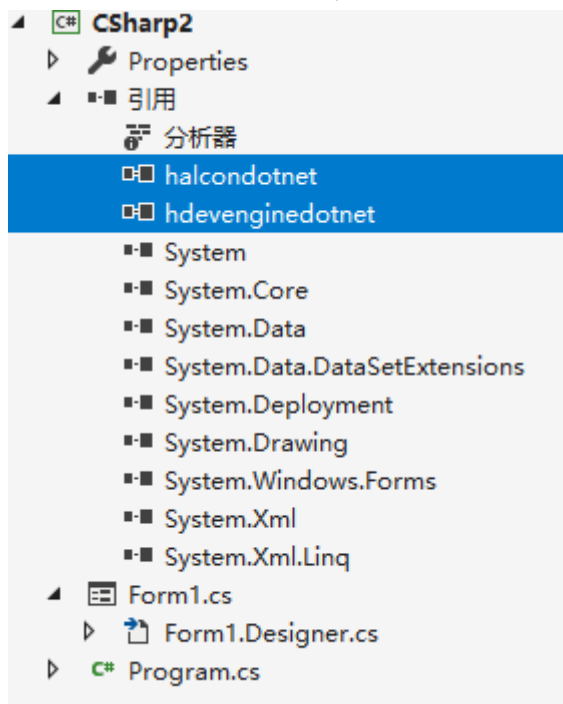
4、从 Halcon 安装路径下 bin 文件夹中的 dotnet35 文件夹中选择“halcondotnet.dll”和“hdevenginedotnet.dll”两个文件，然后点击“添加”



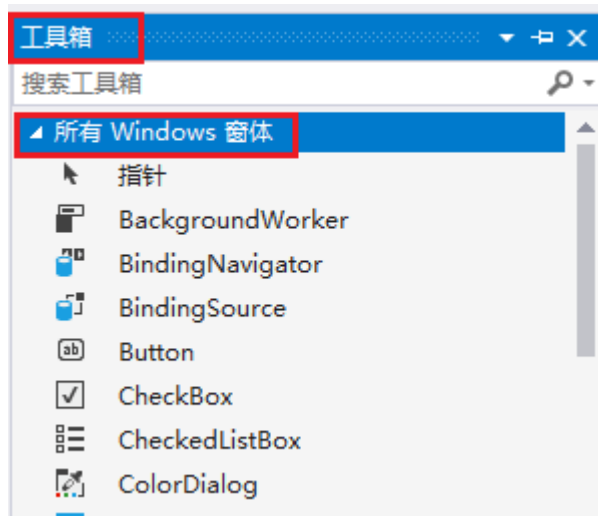
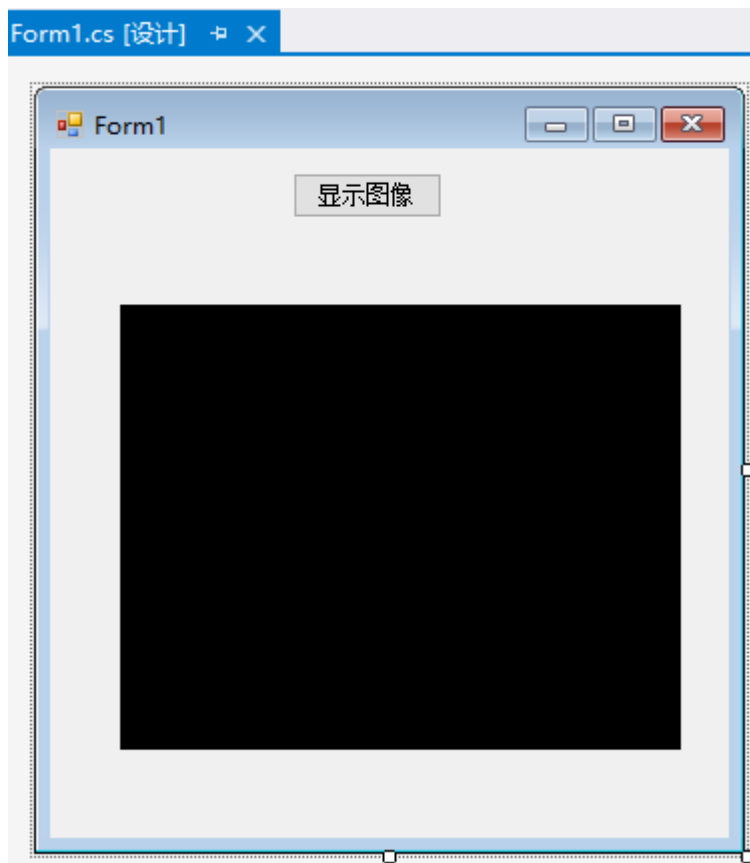
5、第 4 步中点击“添加”后界面如下所示，点击“确定”



6、添加引用后项目中会有下面所示的两个引用



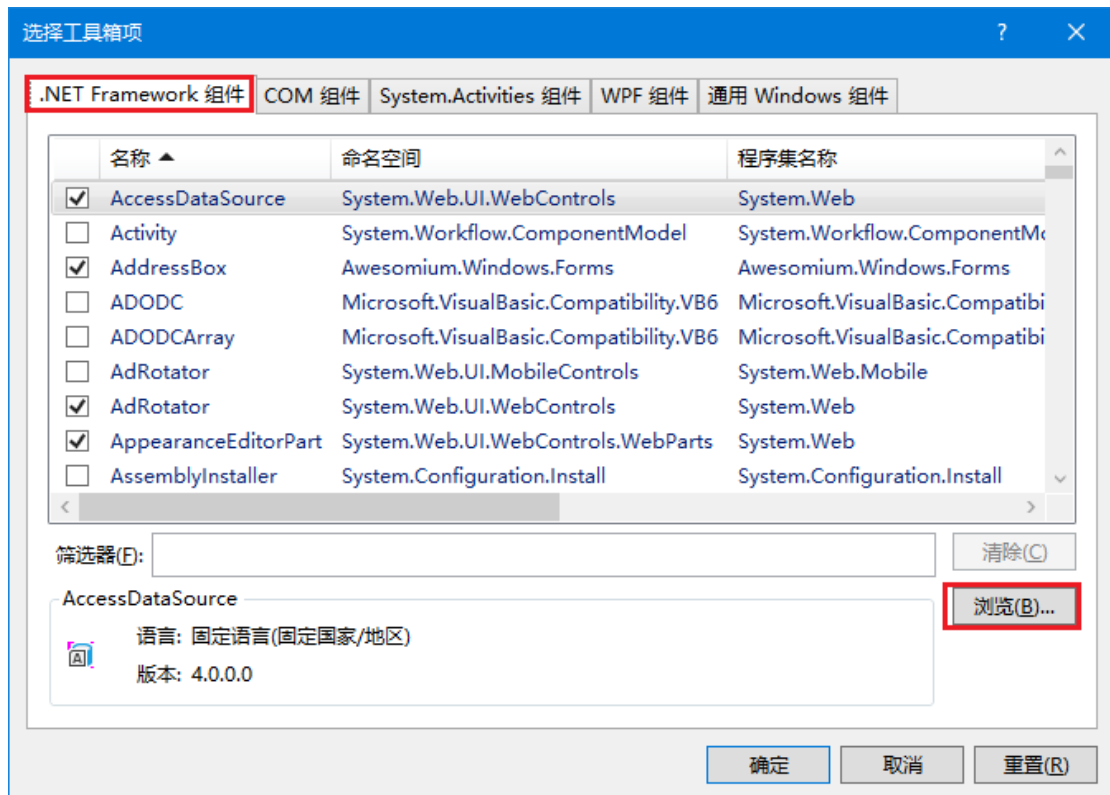
7、切换到 Form1.cs[设计]界面，点击 Form1 窗体后点击“工具箱”



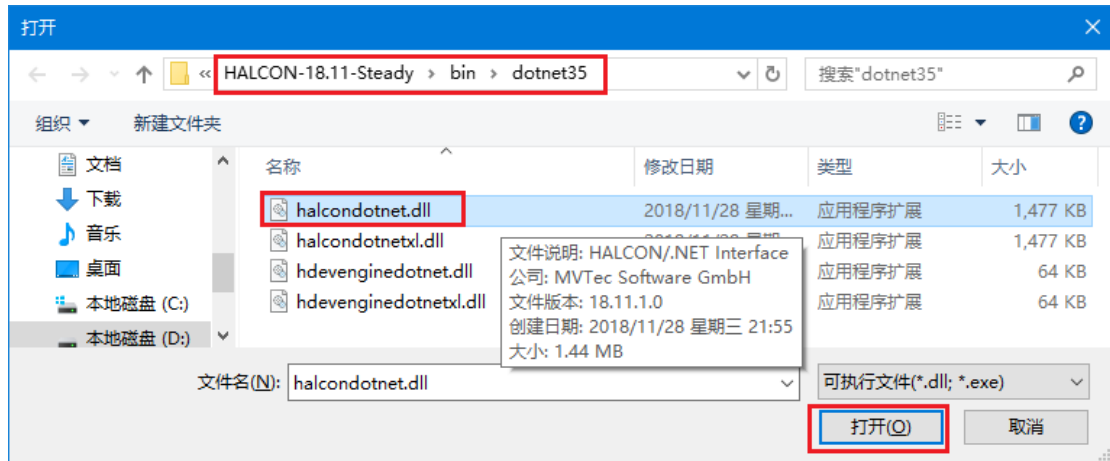
8、右键单击“所有 Windows 窗体”，然后单击“选择项(I)”



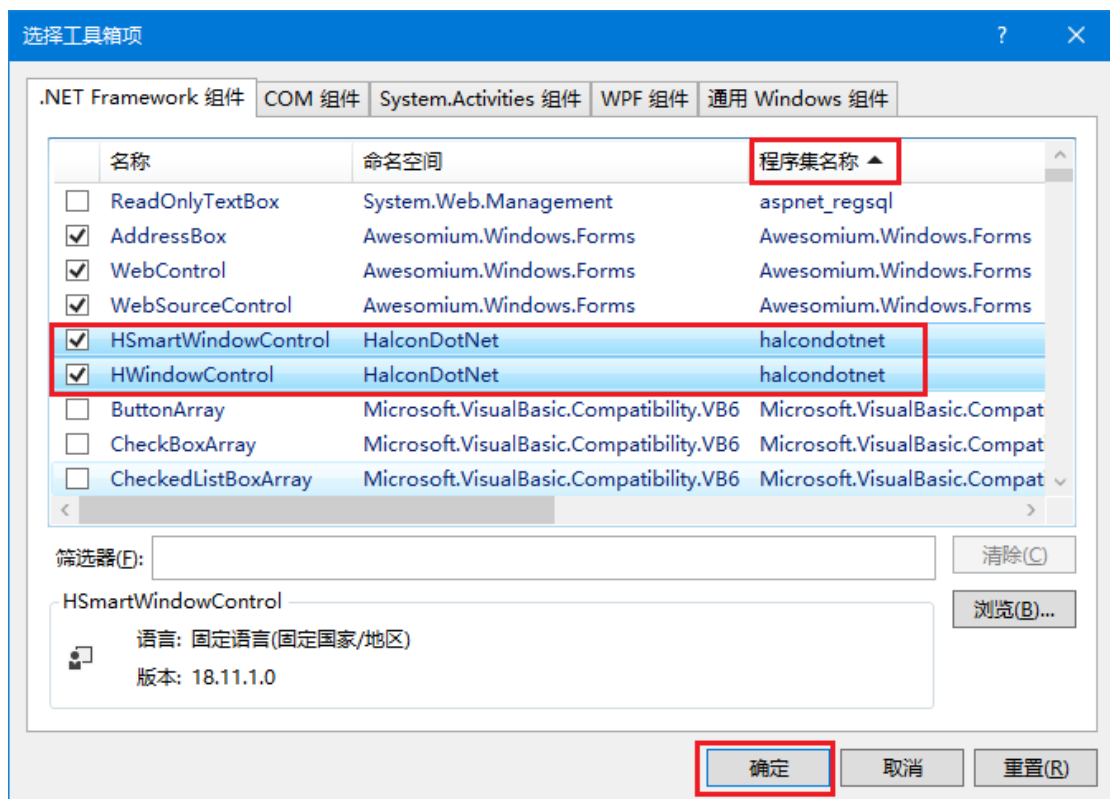
9、在弹出如下界面后，点击“浏览”



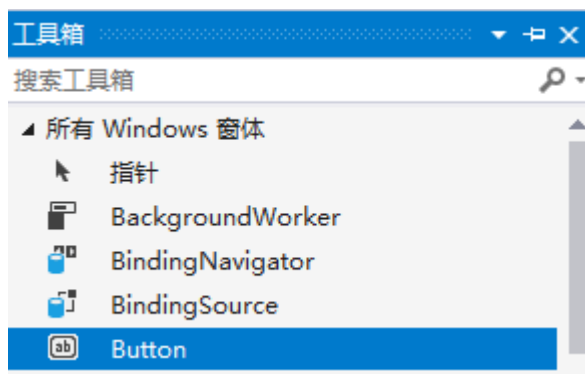
10、找到 Halcon 安装路径下 bin 文件夹中的 dotnet35 文件夹中的“halcondotnet.dll”文件，然后点击“打开”

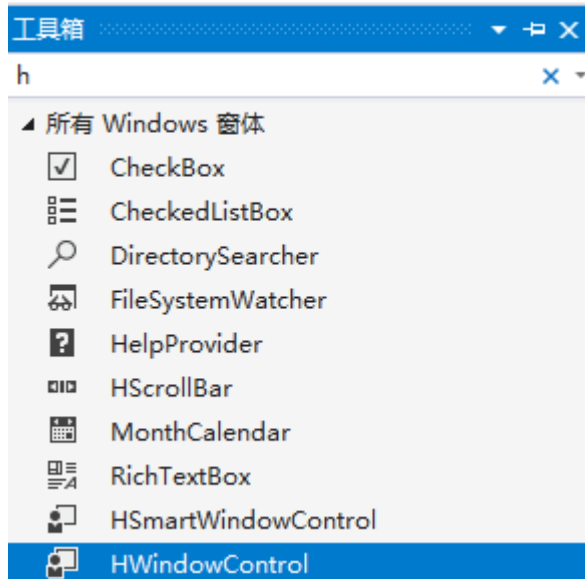
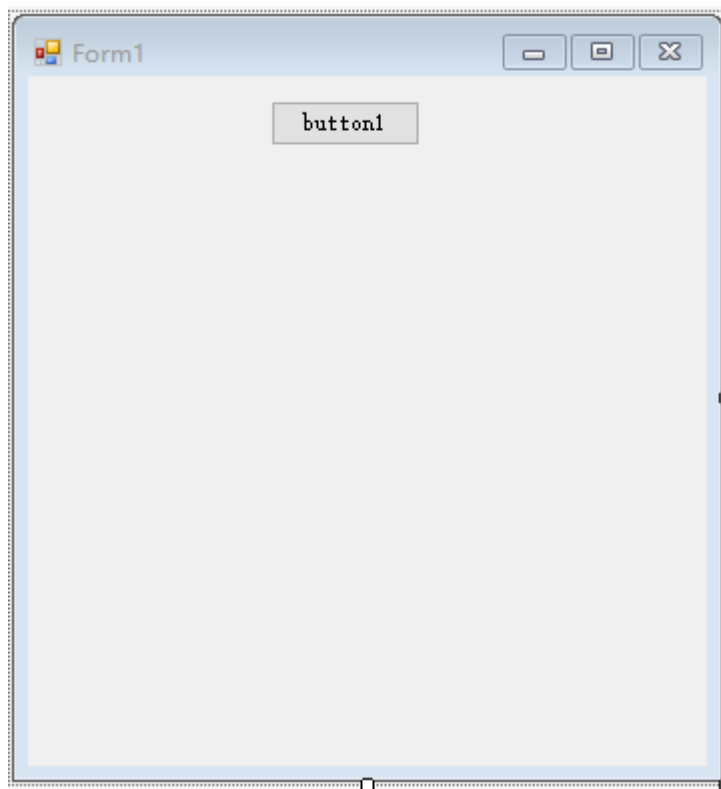


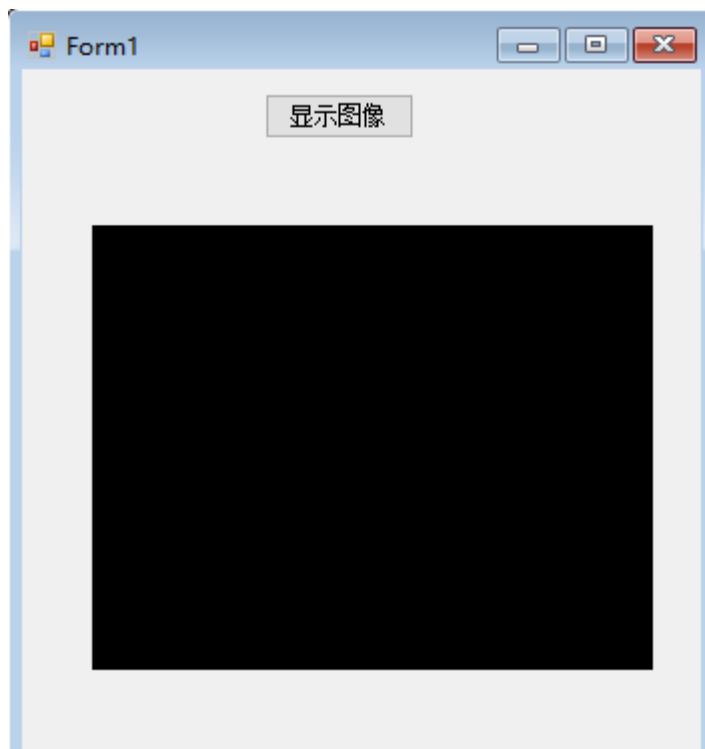
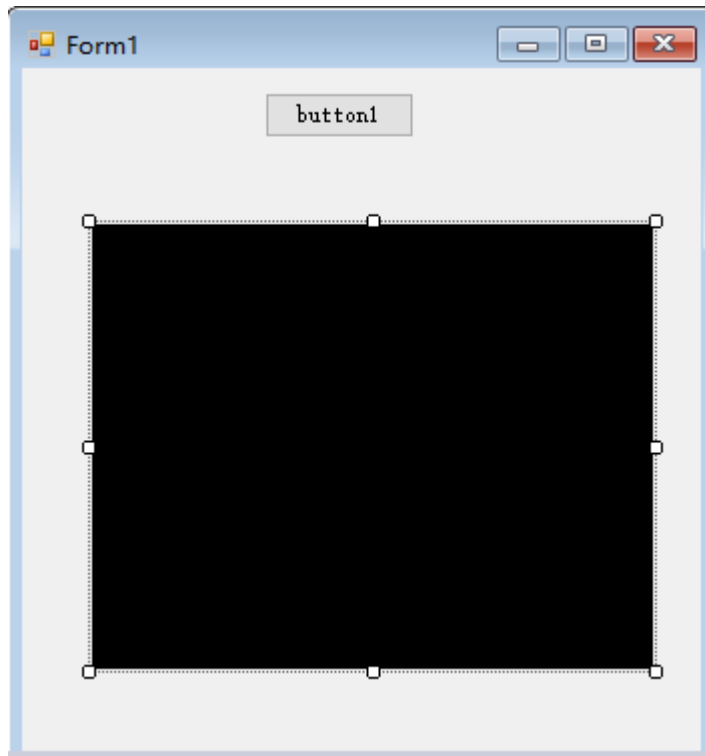
11、在下面的界面中可以看到添加的新的控件，然后点击“确定”



12、在 Form1 窗体中添加一个“Button”控件和一个“HWindowControl”控件，并将“Button”控件的显示文本改成“显示图像”

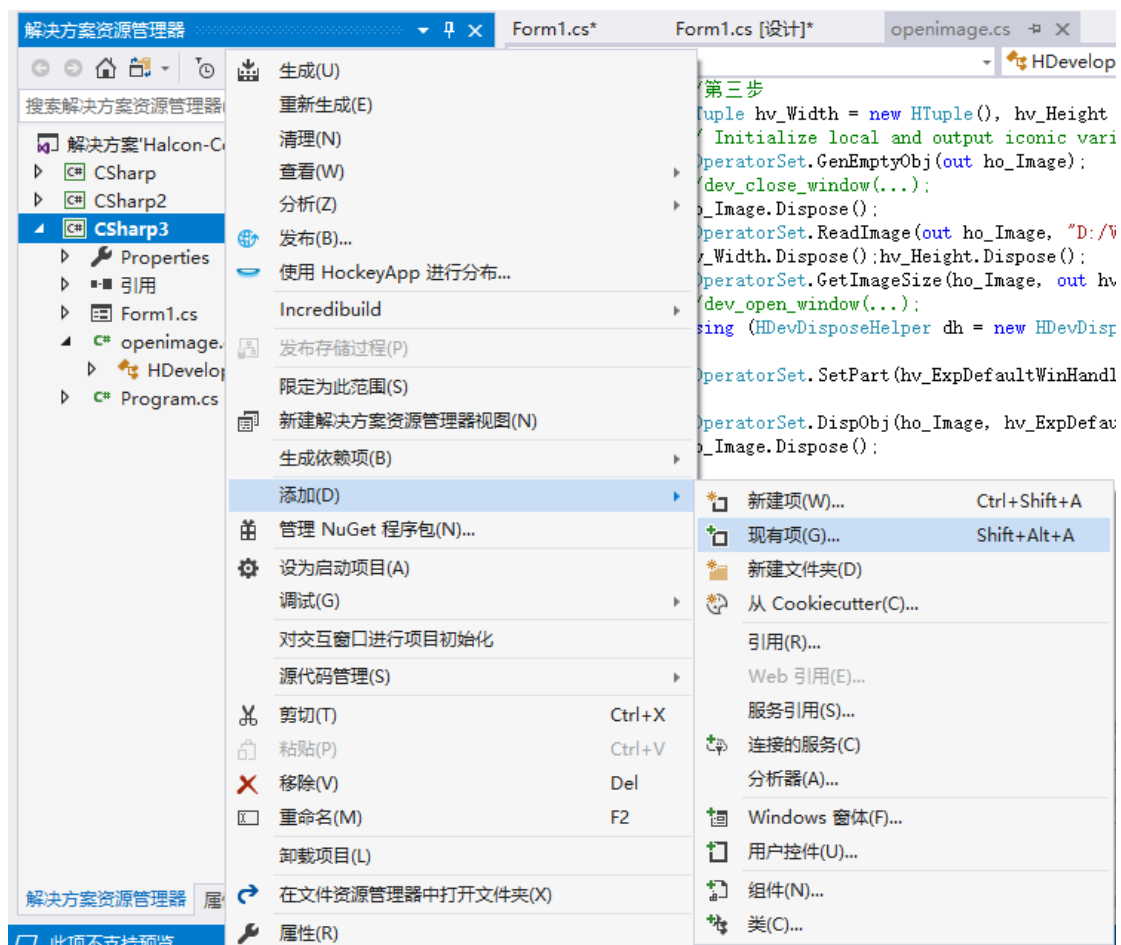




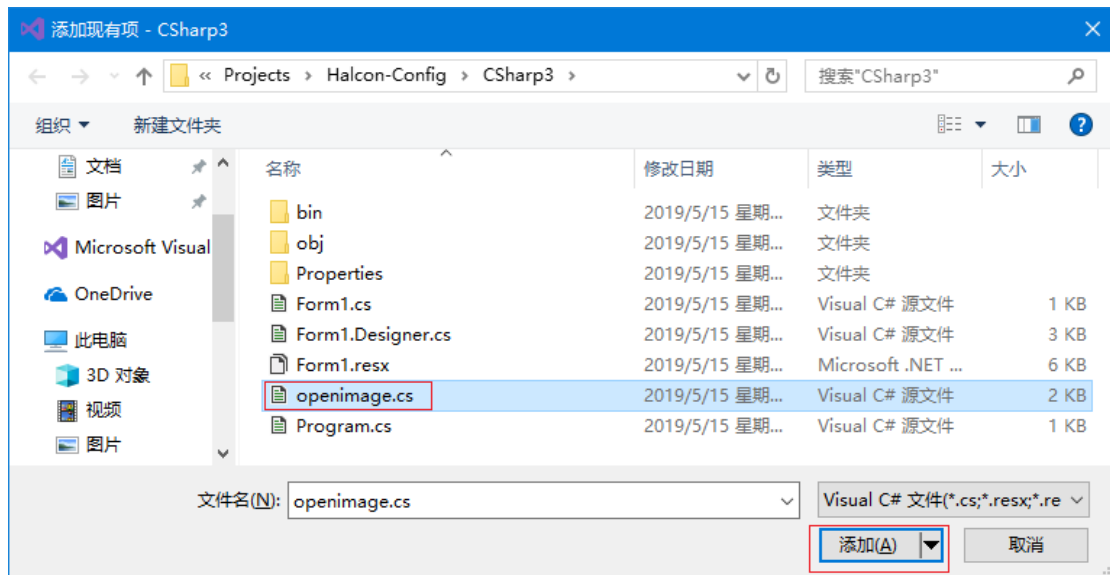


13、右键单击当前的项目名称，在“添加”中点击“现有项(G)”

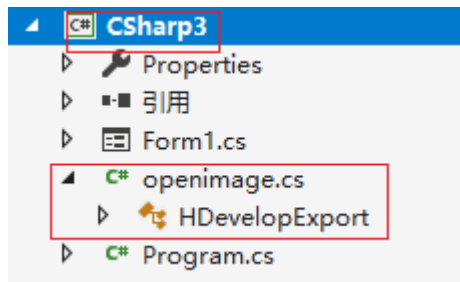




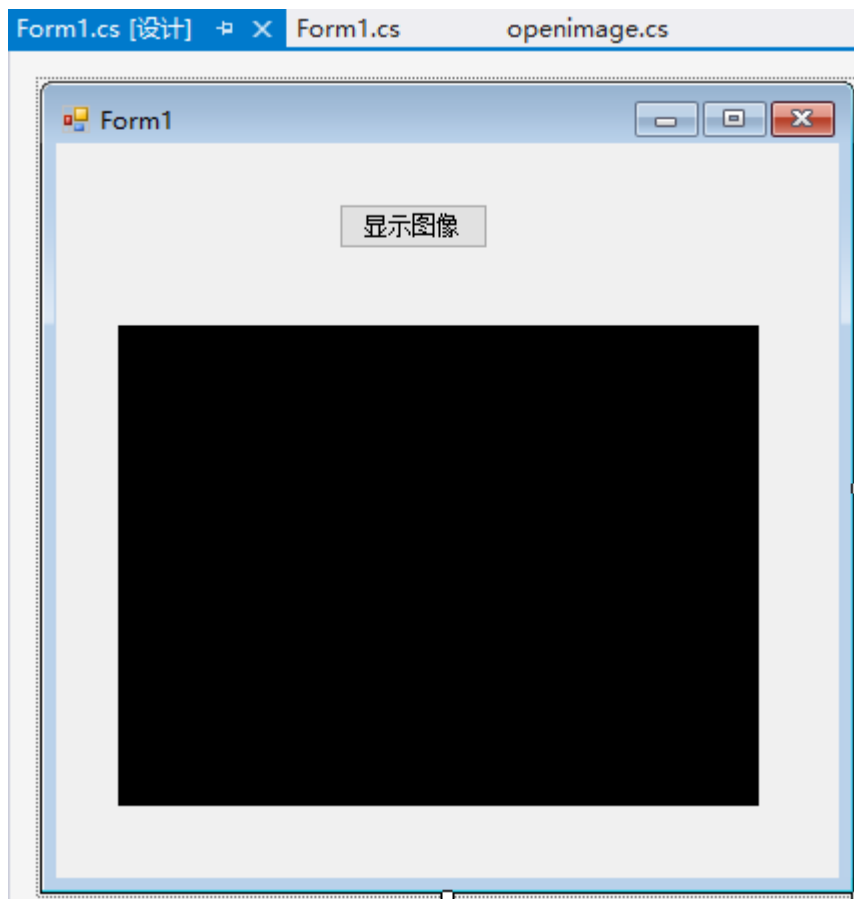
14、找到Halcon导出的C#源代码文件(本人导出的源代码文件是openImage.cs)，点击“添加”



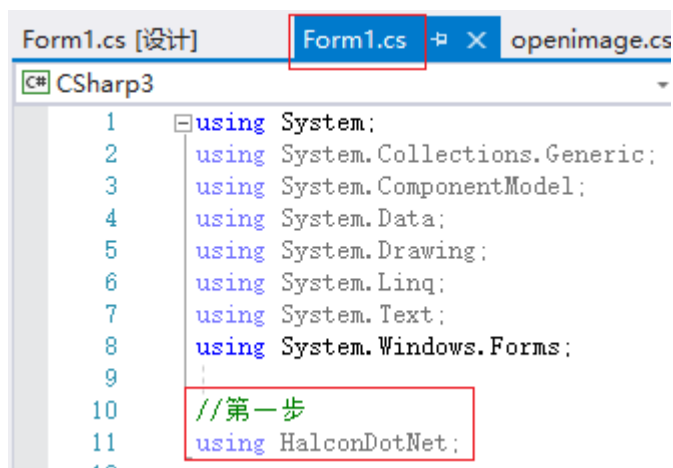
15、添加后可以在项目中看到此文件，如下图所示



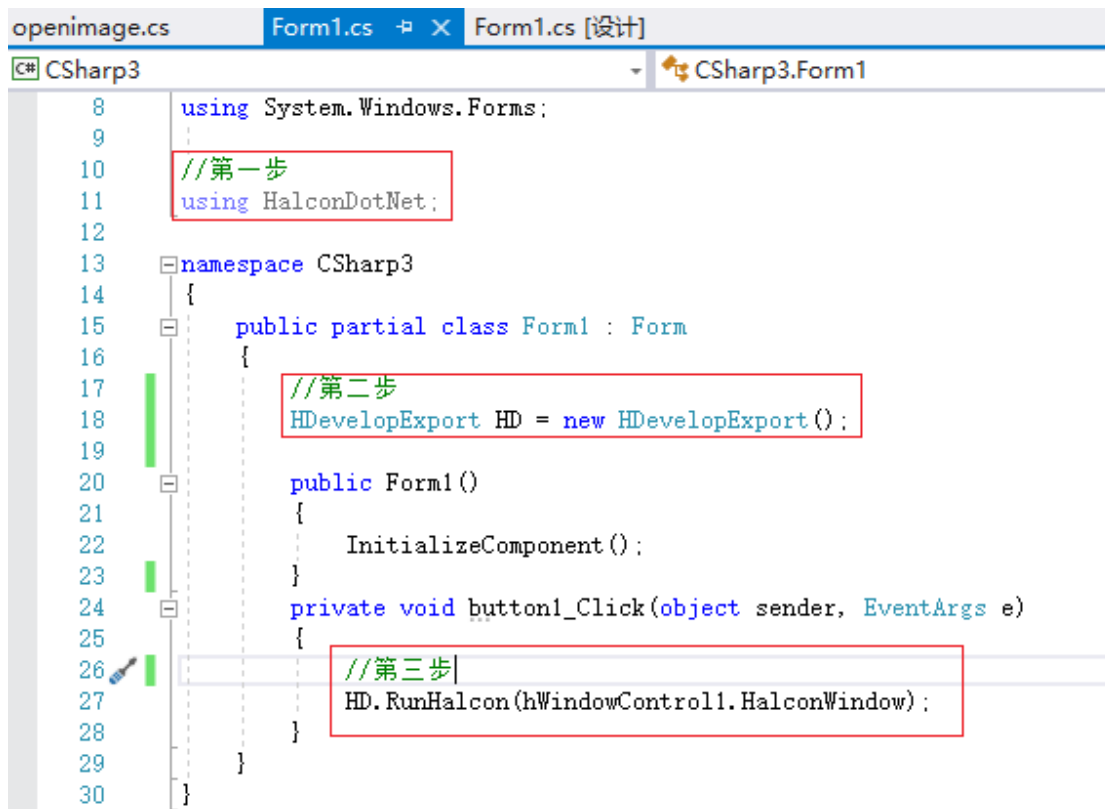
16、双击“Form1.cs[设计]”下 Form1 界面中的“显示图像”，弹出 Form1.cs 文件



17、在 Form1.cs 文件中添加 `using HalconDotNet;`



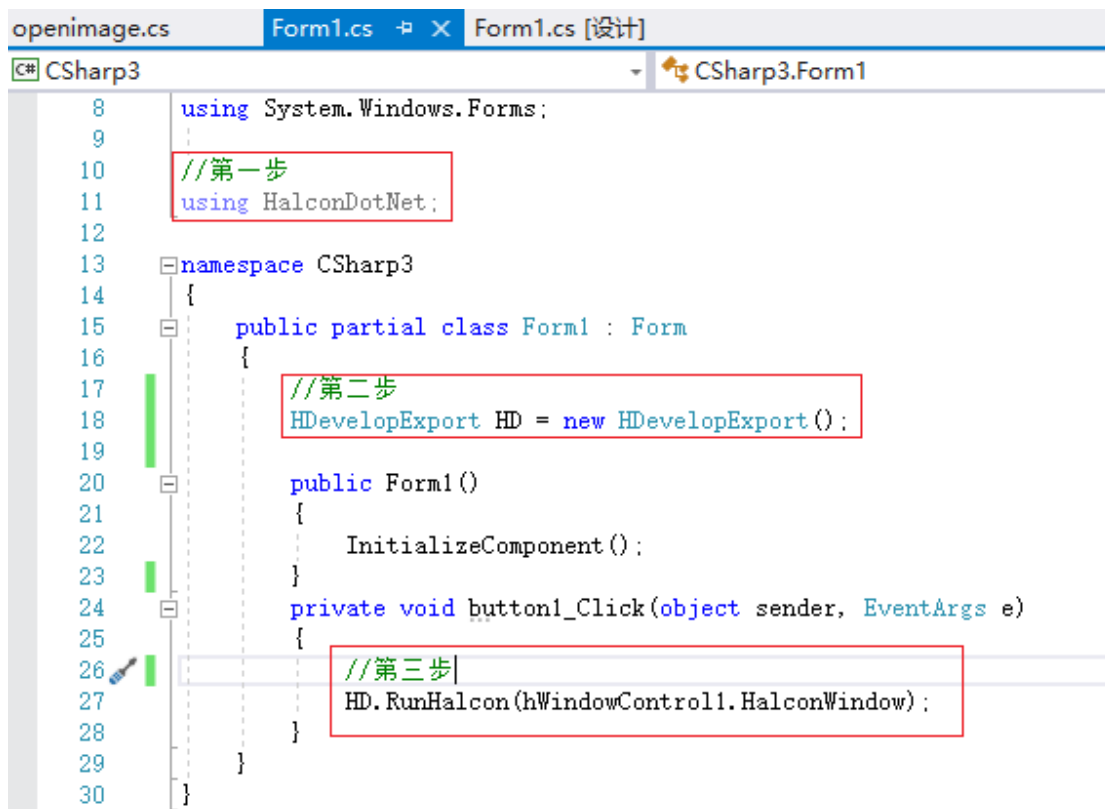
18、在 Form1.cs 文件中添加 `HDevelopExport HD = new HDevelopExport();`;



```
8 using System.Windows.Forms;
9
10 //第一步
11 using HalconDotNet;
12
13 namespace CSharp3
14 {
15     public partial class Form1 : Form
16     {
17         //第二步
18         HDevelopExport HD = new HDevelopExport();
19
20         public Form1()
21         {
22             InitializeComponent();
23         }
24         private void button1_Click(object sender, EventArgs e)
25         {
26             //第三步
27             HD.RunHalcon(hWindowControl1.HalconWindow);
28         }
29     }
30 }
```

19、在 Form1.cs 文件中 button 的 Click 事件(第 16 步中生成)中添加代码

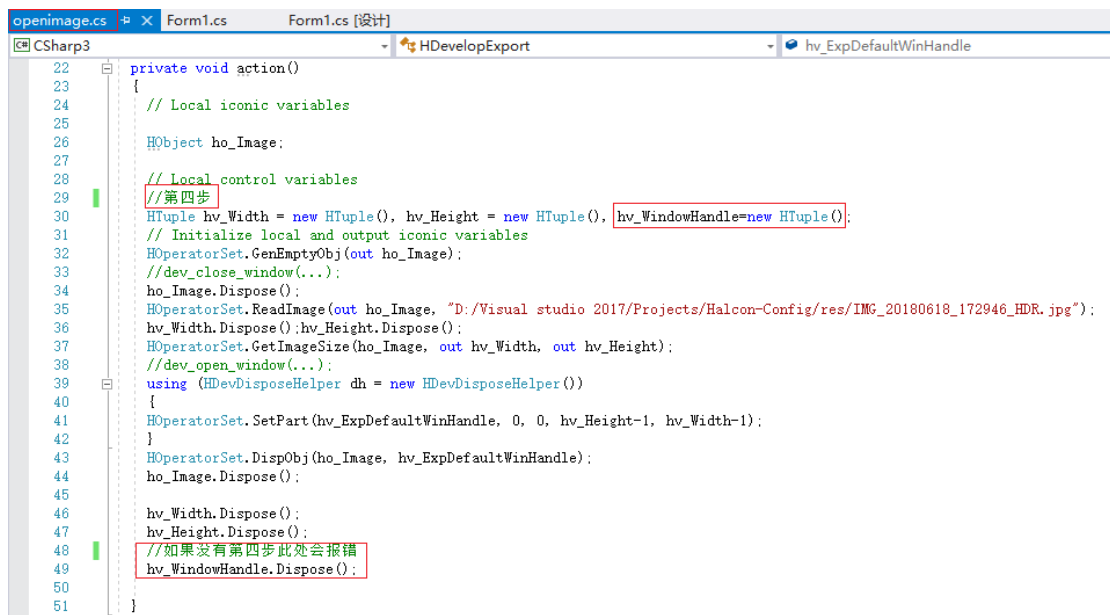
`HD.RunHalcon(hWindowControl1.HalconWindow);`;



```
8 using System.Windows.Forms;
9
10 //第一步
11 using HalconDotNet;
12
13 namespace CSharp3
14 {
15     public partial class Form1 : Form
16     {
17         //第二步
18         HDevelopExport HD = new HDevelopExport();
19
20         public Form1()
21         {
22             InitializeComponent();
23         }
24         private void button1_Click(object sender, EventArgs e)
25         {
26             //第三步
27             HD.RunHalcon(hWindowControl1.HalconWindow);
28         }
29     }
30 }
```

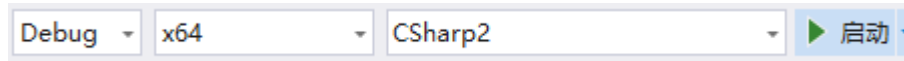
20、在 openImage.cs 文件中的 action 方法中添加代码 `hV_WindowHandle=new HTuple();`

否则会有一处错误，个人认为此为 Halcon 的 bug，后续版本也许会修复

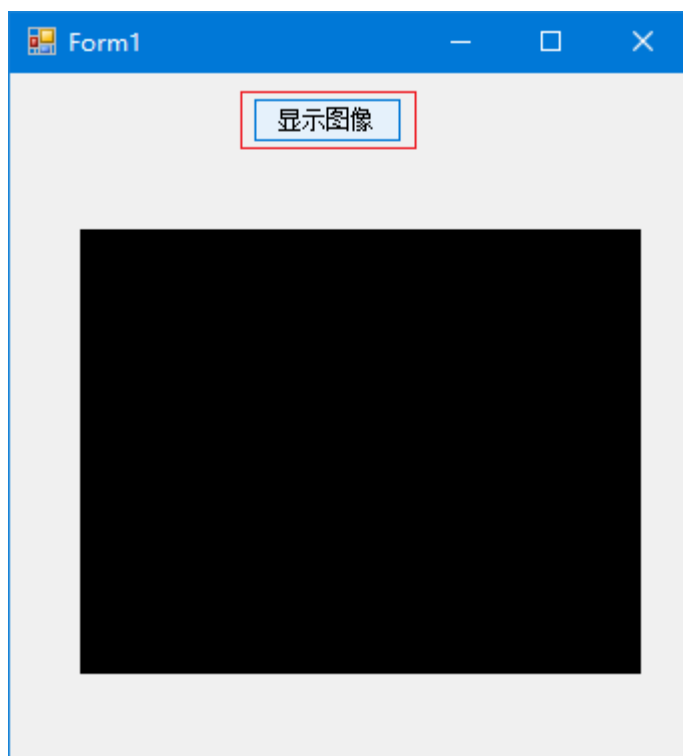


```
22 private void action()
23 {
24     // Local iconic variables
25
26     HObject ho_Image;
27
28     // Local control variables
29     //第四步
30     HTuple hv_Width = new HTuple(), hv_Height = new HTuple(), hv_WindowHandle=new HTuple();
31     // Initialize local and output iconic variables
32     HOperatorSet.GenEmptyObj(out ho_Image);
33     //dev_close_window(...);
34     ho_Image.Dispose();
35     HOperatorSet.ReadImage(out ho_Image, "D:/Visual studio 2017/Projects/Halcon-Config/res/IMG_20180618_172946_HDR.jpg");
36     hv_Width.Dispose(); hv_Height.Dispose();
37     HOperatorSet.GetImageSize(ho_Image, out hv_Width, out hv_Height);
38     //dev_open_window(...);
39     using (HDevDisposeHelper dh = new HDevDisposeHelper())
40     {
41         HOperatorSet.SetPart(hv_ExpDefaultWinHandle, 0, 0, hv_Height-1, hv_Width-1);
42     }
43     HOperatorSet.DispObj(ho_Image, hv_ExpDefaultWinHandle);
44     ho_Image.Dispose();
45
46     hv_Width.Dispose();
47     hv_Height.Dispose();
48     //如果没有第四步此处会报错
49     hv_WindowHandle.Dispose();
50
51 }
```

21、由于本人安装的 Halcon 只有 x64 版本，因此此处选择 x64 版本，如果是 x86 则选择 x86，如果两者都安装了则选择 Any CPU，然后点击“启动”



22、运行后界面如下图所示，点击“显示图像”



23、如果弹出下面的界面就说明 Halcon 与.NET 的配置成功了，可以进行复杂的程序操作了

