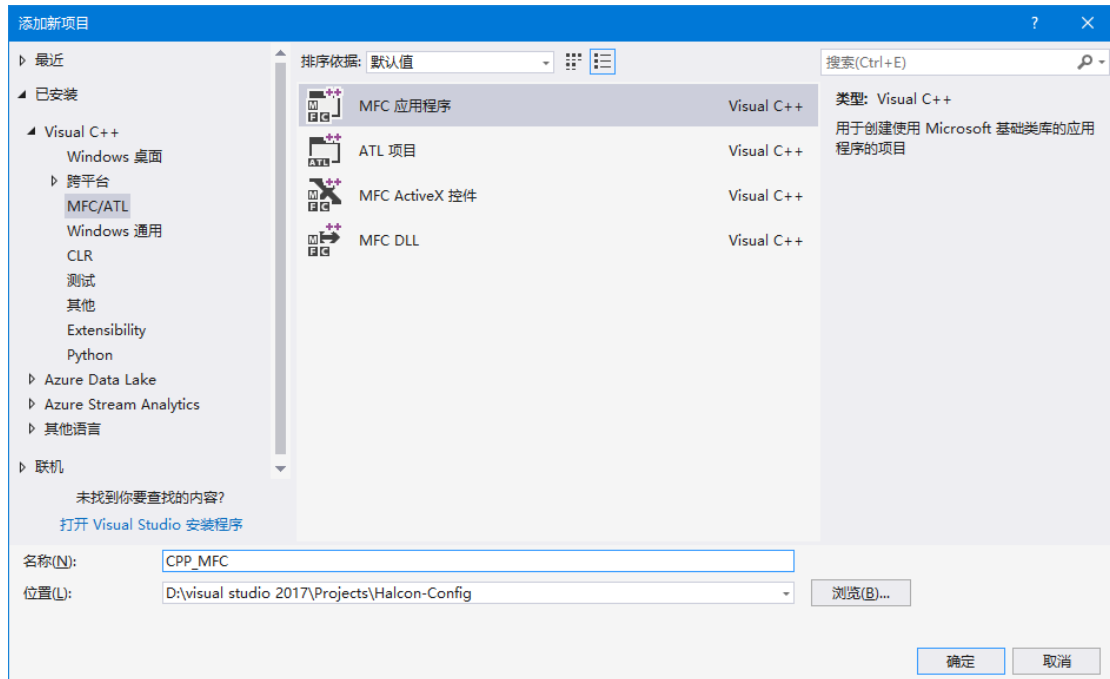


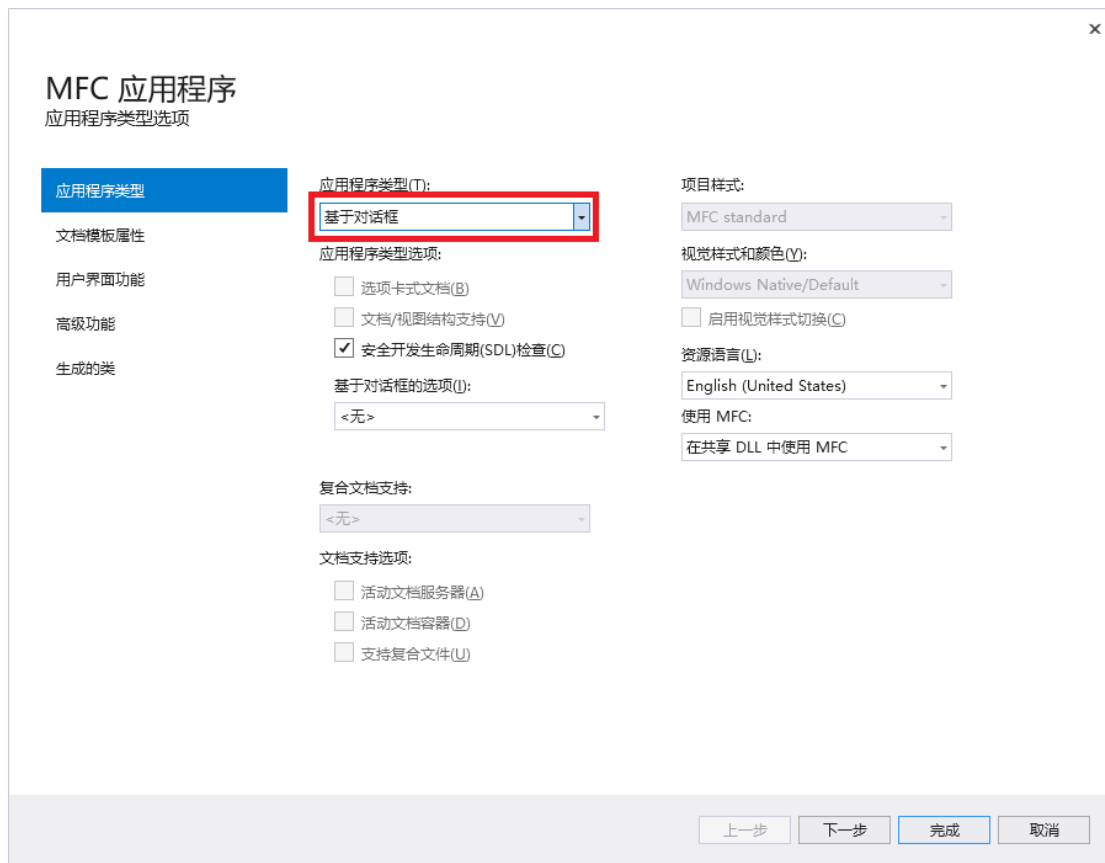
# Halcon 与 C++混合编程

## 基础以及 Halcon 在 MFC 中的简单使用

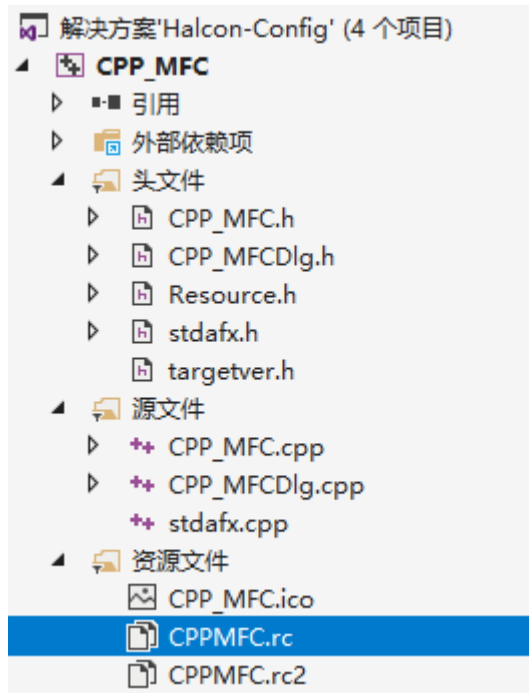
### 1、新建一个 MFC 应用程序项目



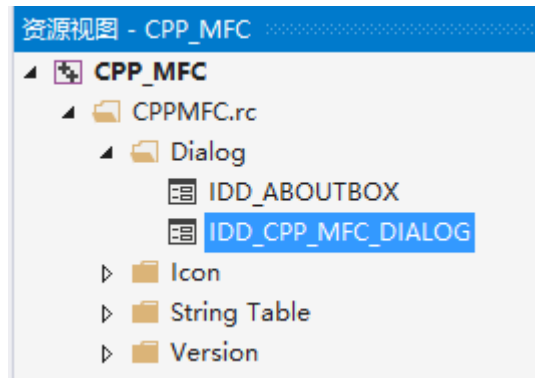
### 2、应用程序类型选择“基于对话框”，然后点击“完成”



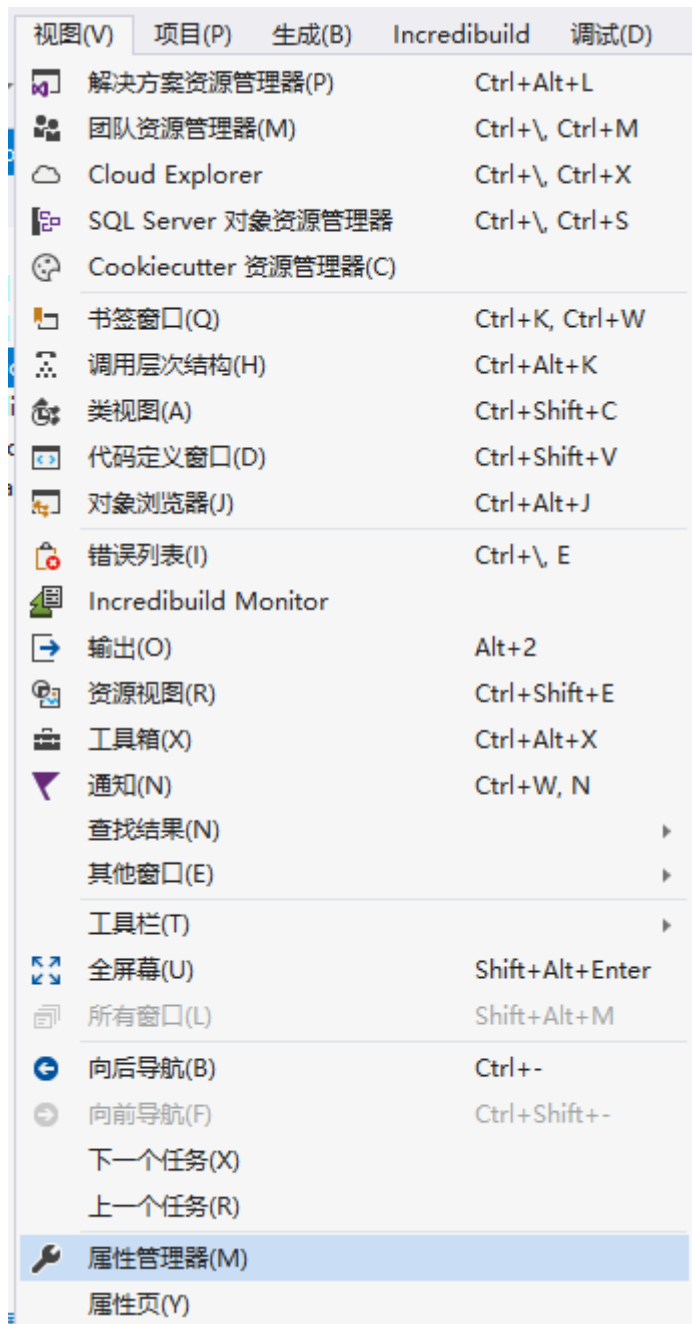
3、在新建的工程中双击 CPPMFC.rc（名字可能有变化，主要看后缀）文件



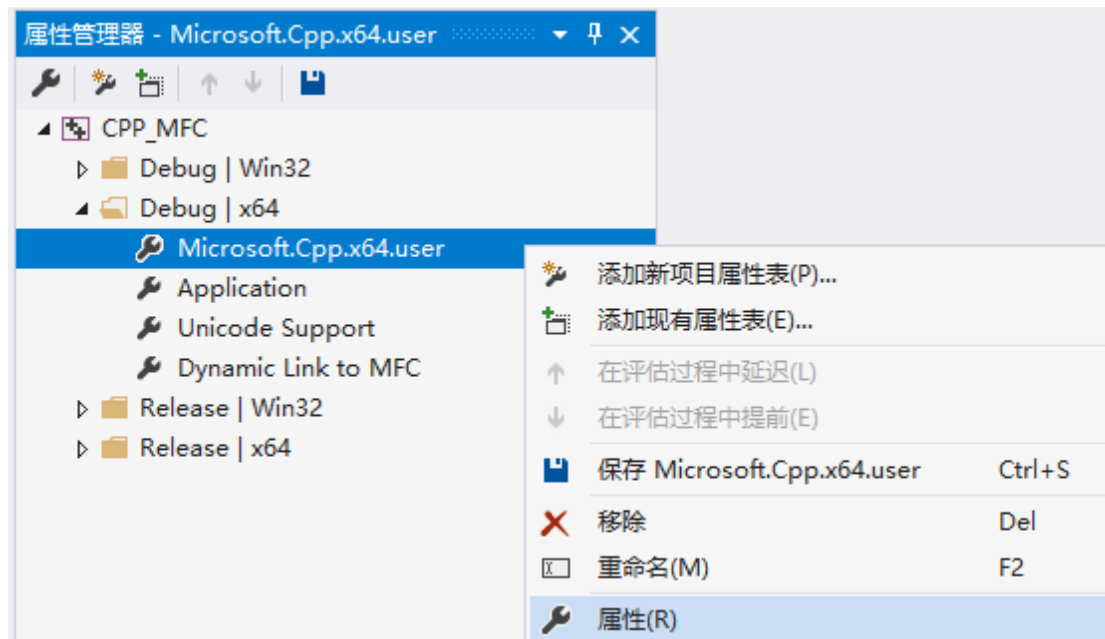
4、双击 Dialog 文件夹下的 IDD\_CPP\_MFC\_DIALOG 文件可以打开 MFC 的界面



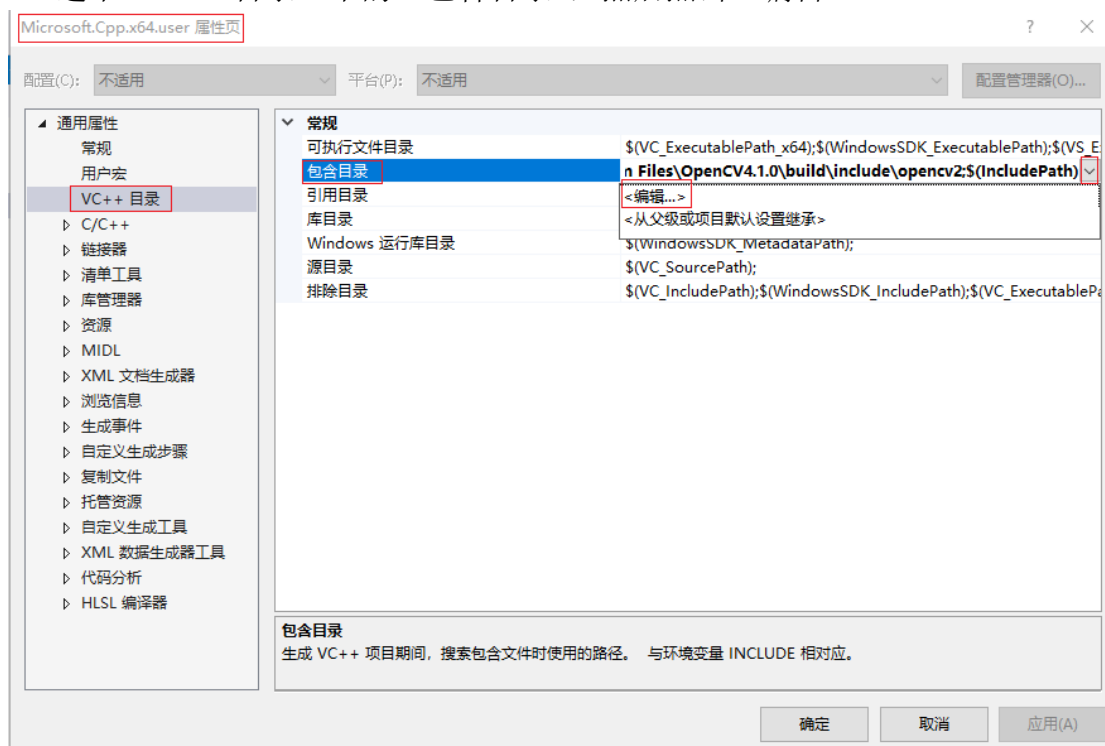
5、点击“视图(V)”菜单中的“属性管理器(M)”，下面开始在 C++中配置 Halcon，本文中的方法配置后每次新建一个 C++工程就不再需要重新配置了，网络上很多配置方法每次新建工程后需要重新配置



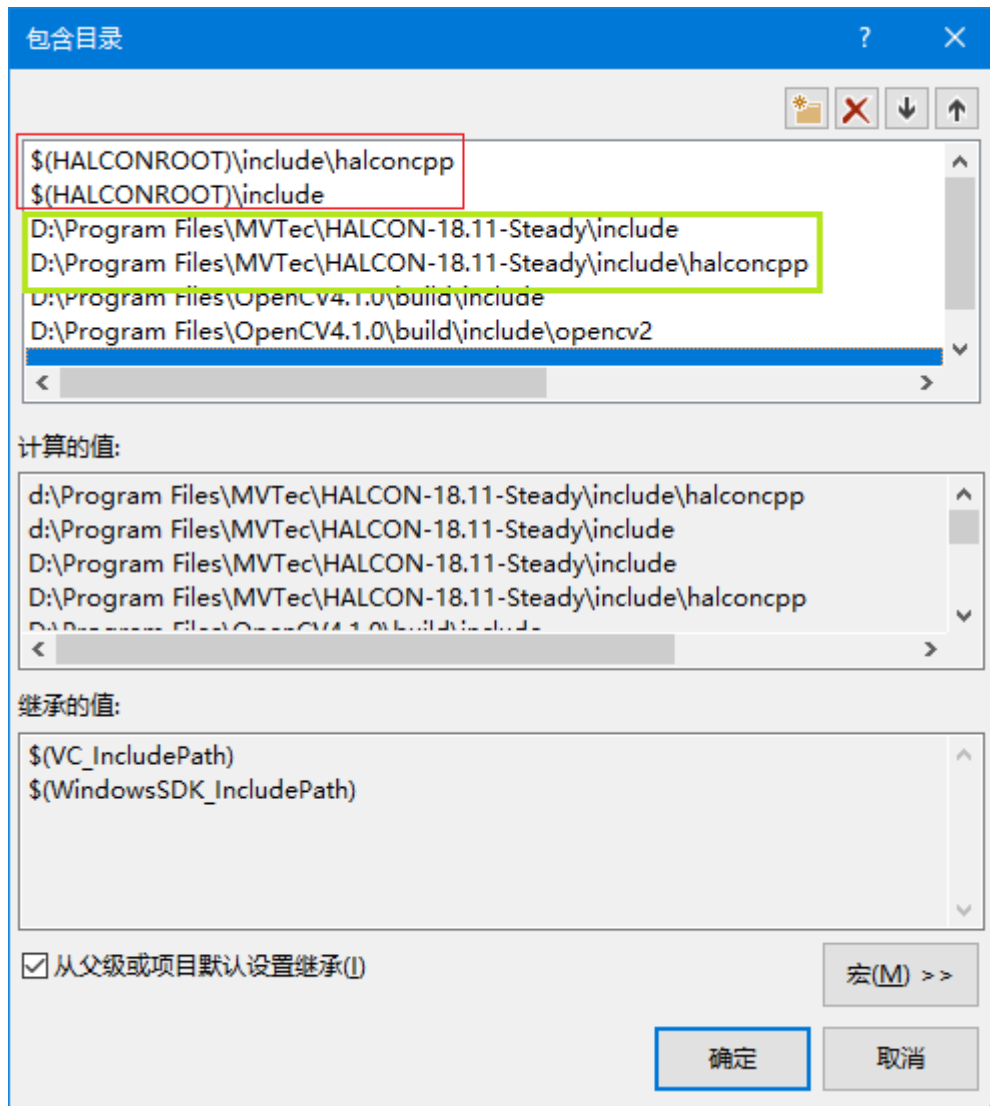
6、点击 Debug|x64 下的 Microsoft.cpp.x64.user，右键点击“属性”，此处依据个人实际情况配置 x86 或者 x64，只需要配置 Debug 类型就可以了，Release 类型会自动配置好



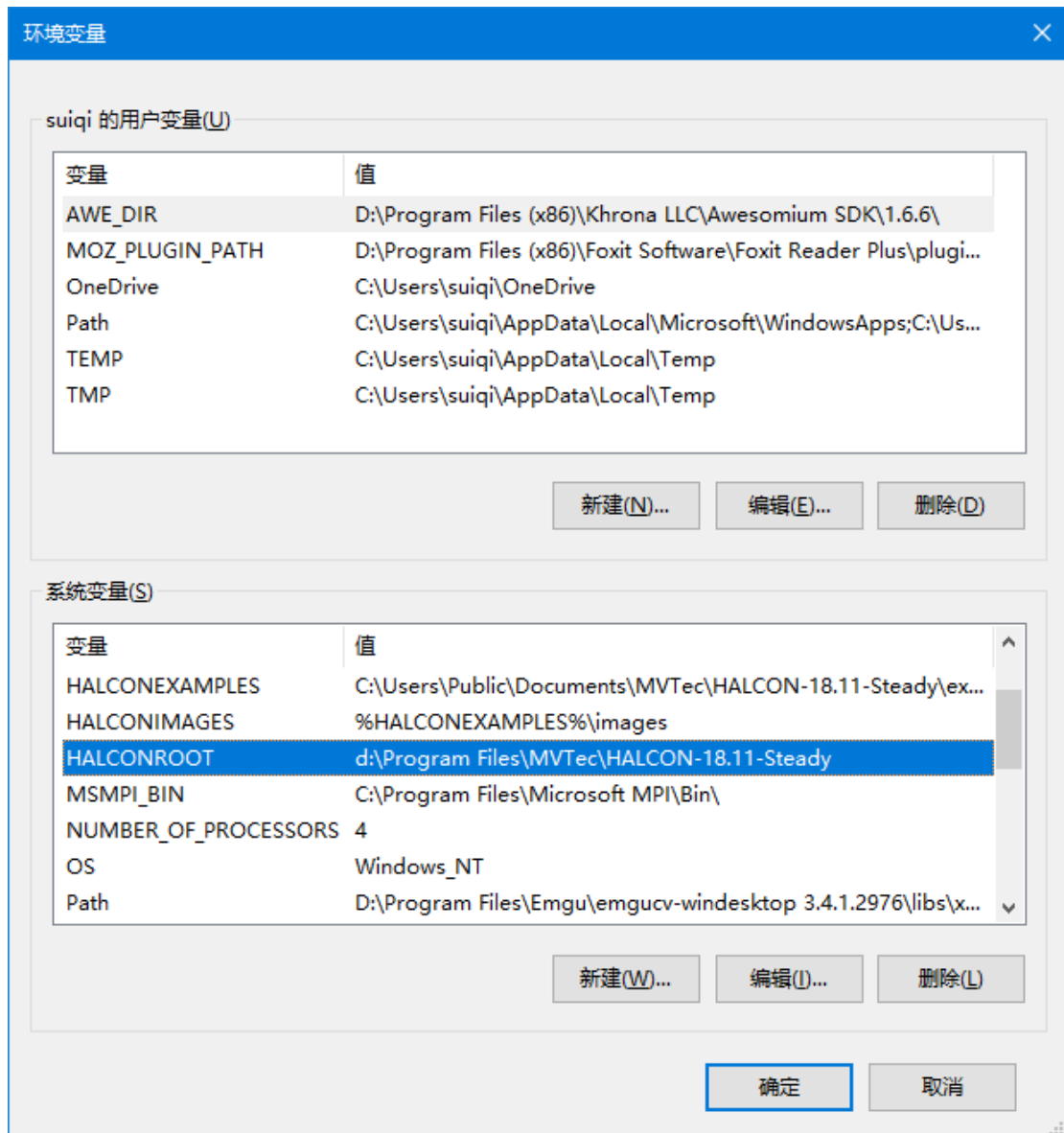
7、选中“VC++目录”中的“包含目录”，然后点击“编辑”



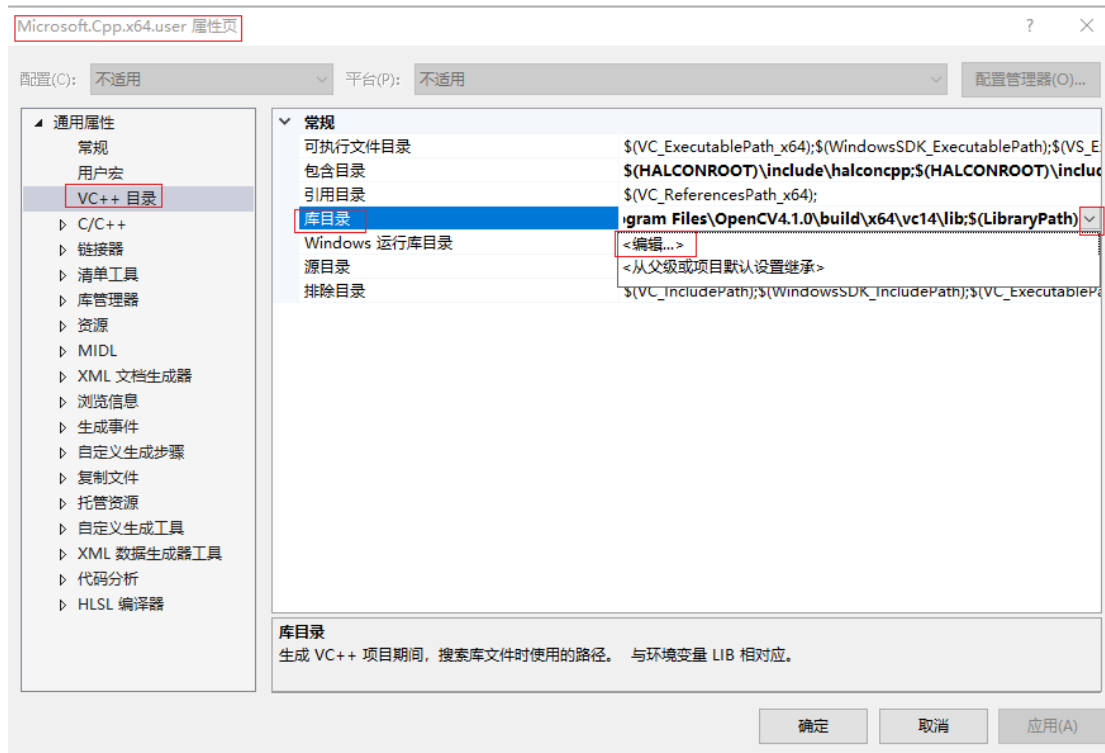
8、如下图中红色框或者绿色框标出的那样子，添加 Halcon 安装路径下的 include 文件夹以及 include 文件夹下的 halconcpp 文件夹



9、如上图红色框那样配置可以更具有通用性，因为安装 Halcon 时会自动在系统变量中添加其安装路径

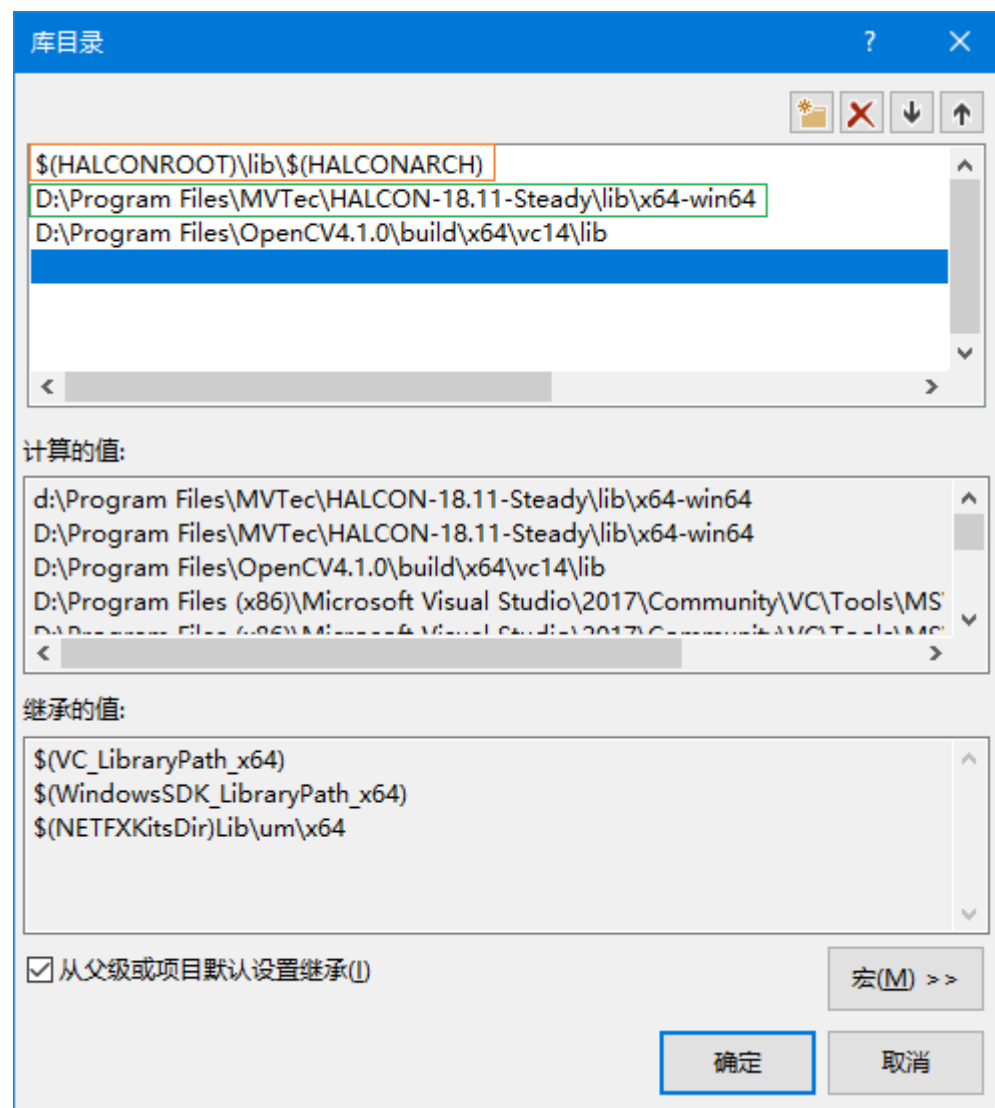


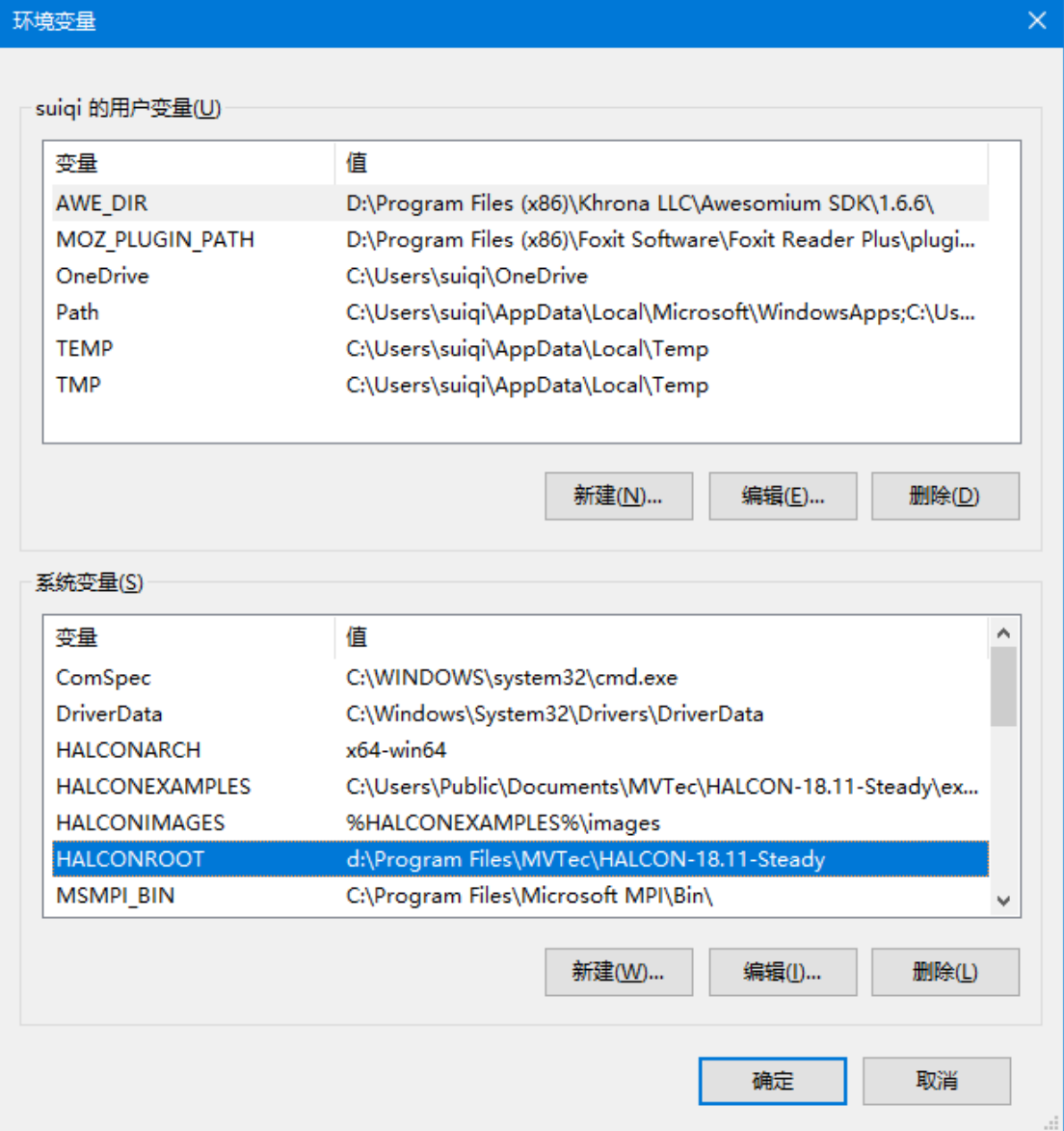
10、选中“VC++目录”下的“库目录”，然后点击“编辑”

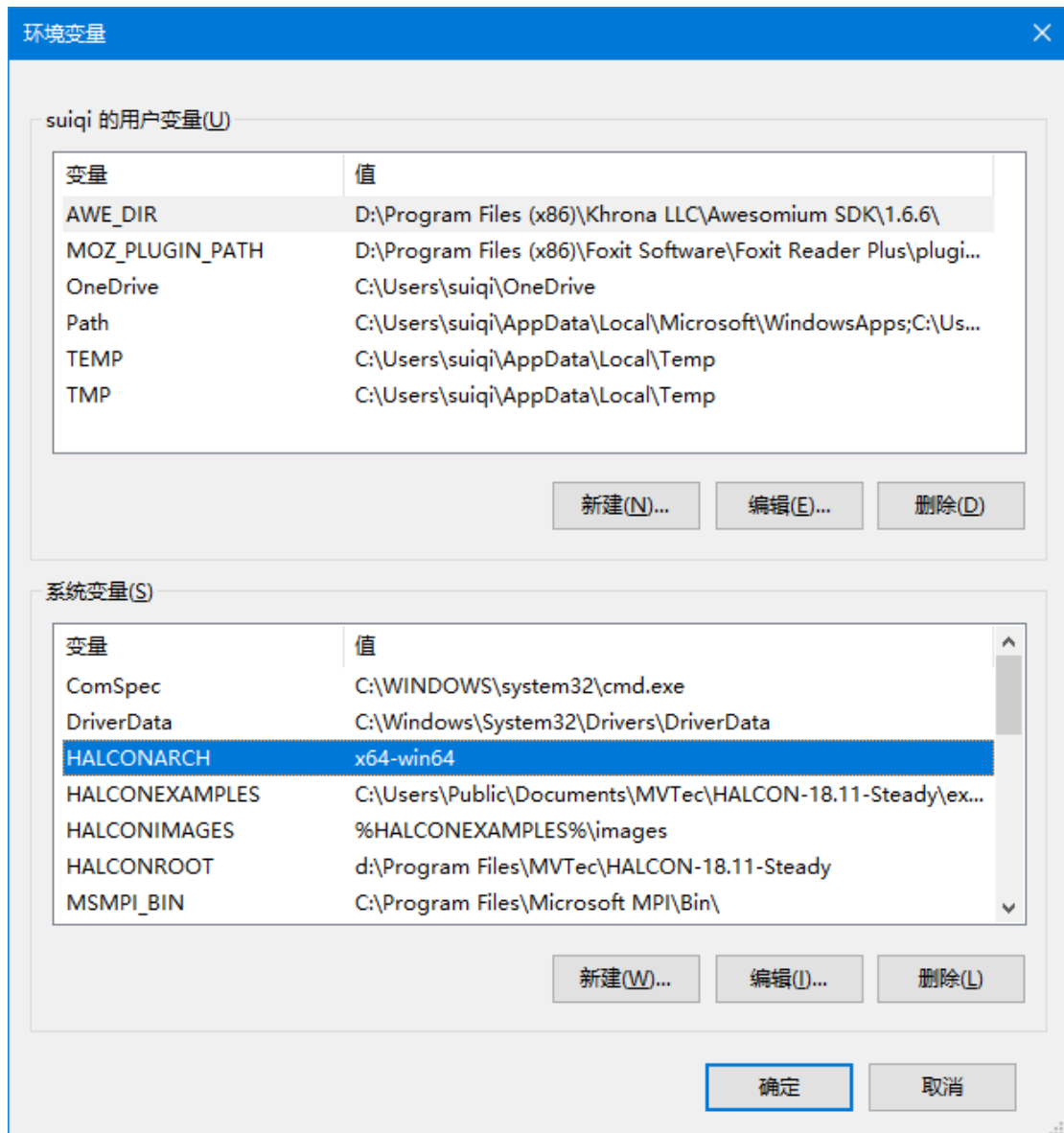


11、如下图红色框或者绿色框标出的那样，添加 Halcon 安装目录下的 lib 文件夹中的 x64-win64，此处依据需要选择版本，如果前面选择配置的是 Win32，那么此处做相应改变，红色框的配置同样具有通用性，这样子配置也同样的因为 Halcon 安装时自动在系统环境变量中添加了其路径及版本信息

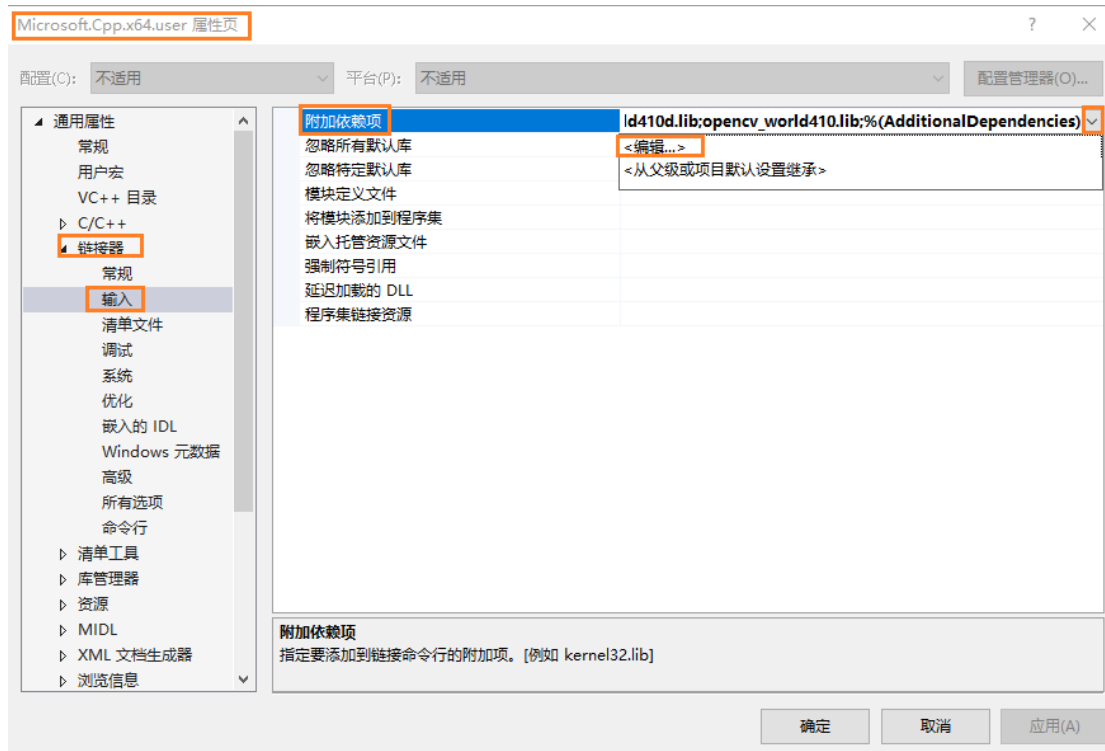


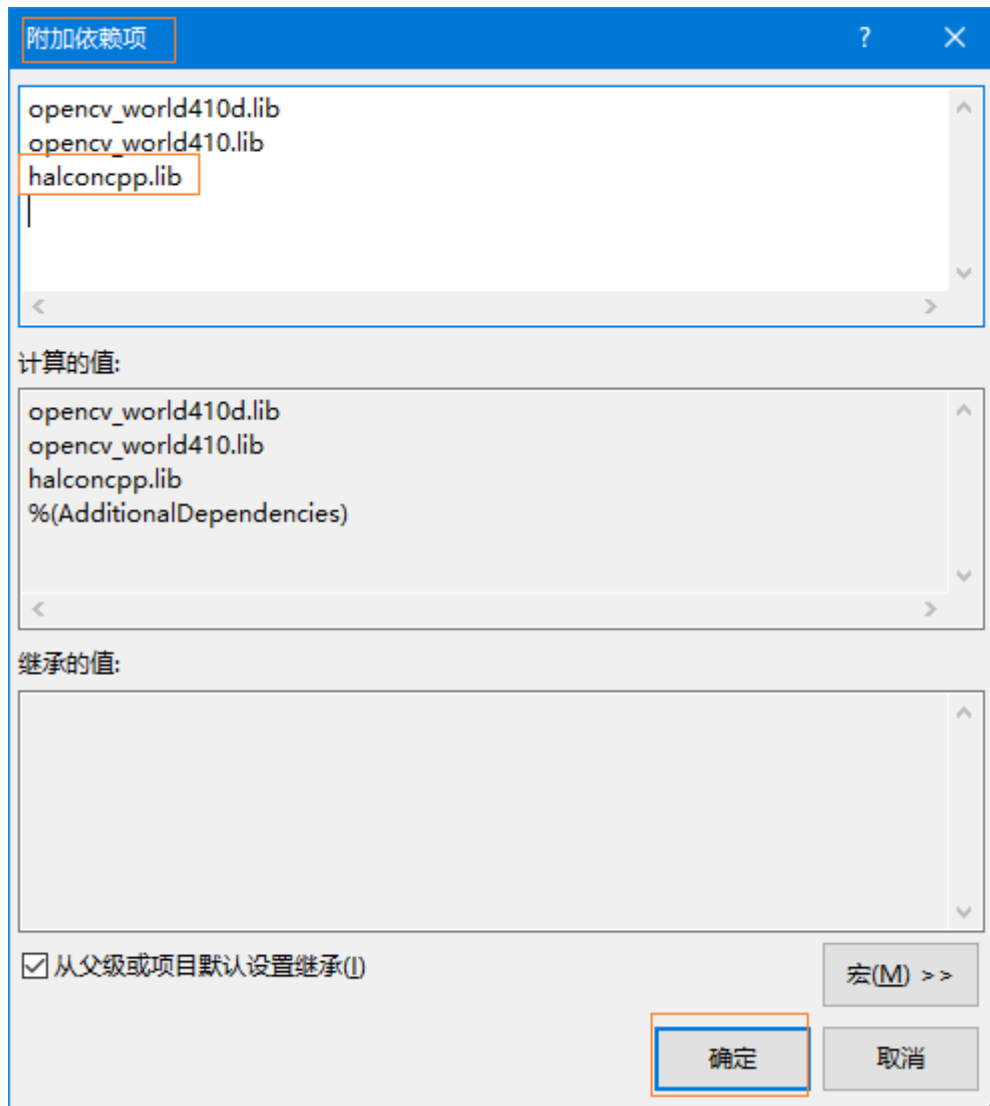




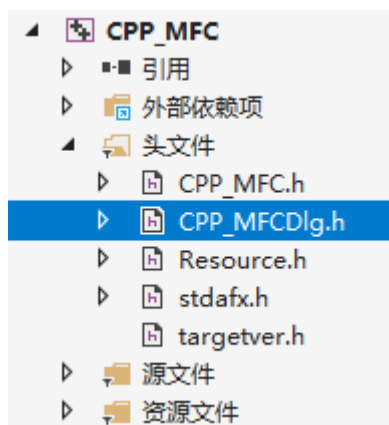


12、点击“链接器”下的“输入”，选中“附加依赖项”，然后点击“编辑”，添加 **halconcpp.lib**，然后点击“确定”，这样子在 C++ 中使用 Halcon 就配置完成了





13、打开 CPP\_MFC 下的 CPP\_MFCDlg.h 文件（后面是 Dlg.h，前面依据个人的工程名称变化）

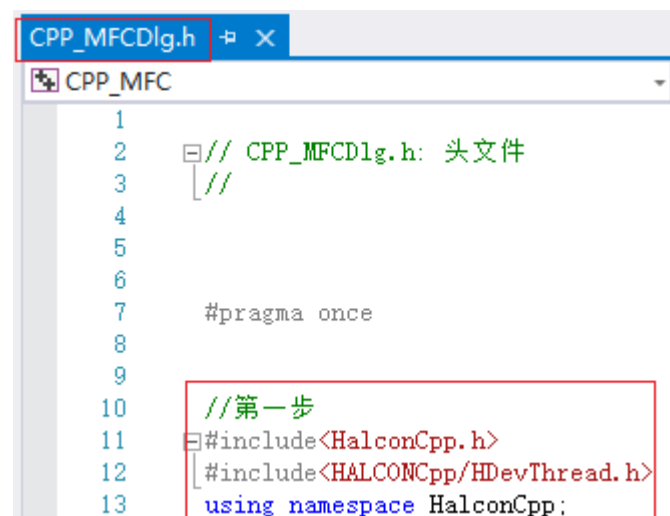


14、在 CPP\_MFCDlg.h 文件（后面是 Dlg.h，前面依据个人的工程名称变化）中添加代码

```
#include<HalconCpp.h>
#include<HALCONCpp/HDevThread.h>
```

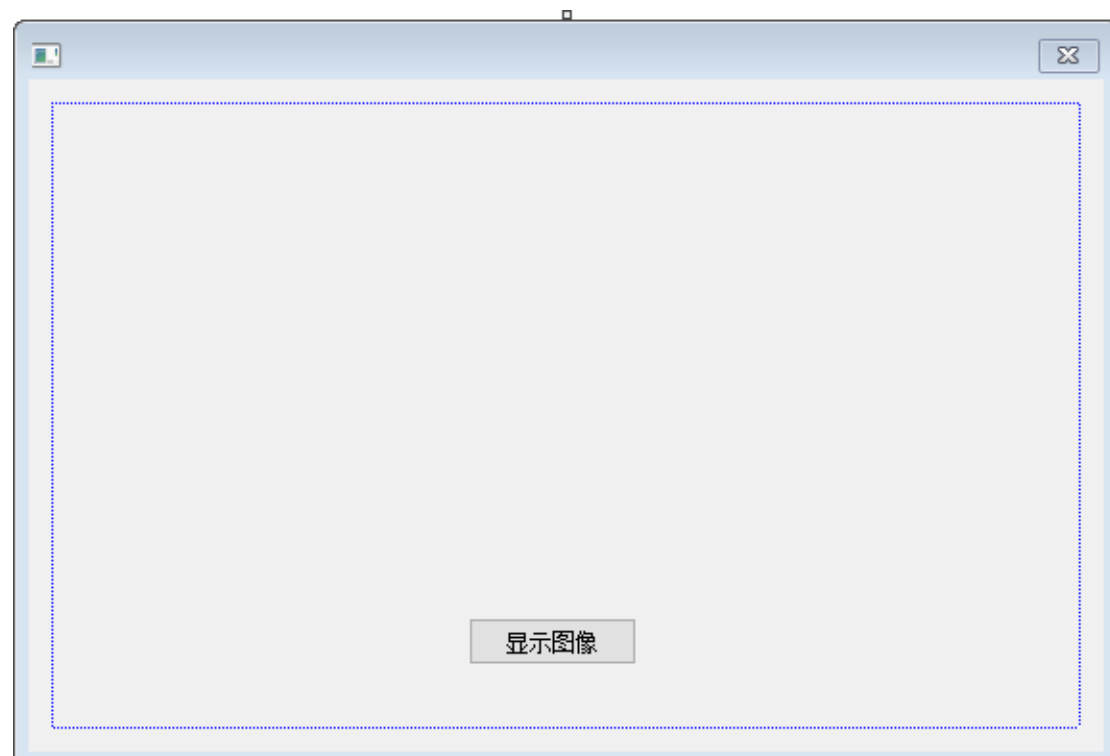
```
using namespace HalconCpp;
```

如下图所示



此处网络上有的网友写的是 `using namespace halcon;`，可能之前的版本是这样子使用吧，最新的 Halcon 这样子书写是不正确的

15、在第 4 步中打开了 MFC 的界面，先将界面上的无关控件删掉，然后点击工具箱添加 button 控件，然后点击该控件，右键属性，在 Caption 中将其名字改成“显示图像”，（按照此步骤继续后面显示的图像不在 MFC 窗口中，如果需要显示在 MFC 窗口中直接转到第 22 步）如下图所示：



16、双击“显示图像”，在 CPP\_MFCDlg.cpp(后面是 Dlg.cpp，前面依据个人的工程名称变化)中 Button 控件的事件函数（本步骤中的双击动作会自动产生）中添加 Halcon 生成的 CPP 代码中的 action 函数中的内容，如下图所示：

```

155
156
157 //第二步
158 void CCppMFCDlg::OnBnClickedButton1()
159 {
160     // TODO: 在此添加控件通知处理程序代码
161     // Local iconic variables
162     HObject ho_Image;
163
164     // Local control variables
165     HTuple hv_Width, hv_Height, hv_WindowHandle;
166
167     //第三步
168     /*if (HDevWindowStack::IsOpen())
169         CloseWindow(HDevWindowStack::Pop());*/
170     ReadImage(&ho_Image, "D:/Visual studio 2017/Projects/Halcon-Config/res/2.jpg");
171     GetImageSize(ho_Image, &hv_Width, &hv_Height);
172     SetWindowAttr("background_color", "black");
173     OpenWindow(0, 0, hv_Width, hv_Height, 0, "visible", "", &hv_WindowHandle);
174     HDevWindowStack::Push(hv_WindowHandle);
175     if (HDevWindowStack::IsOpen())
176         SetPart(HDevWindowStack::GetActive(), 0, 0, hv_Height - 1, hv_Width - 1);
177     if (HDevWindowStack::IsOpen())
178         DispObj(ho_Image, HDevWindowStack::GetActive());
179
180 }

```

下图所示就是 Halcon 生成的 C++ 源程序中的 action 函数

```

// Main procedure
void action()
{

    // Local iconic variables
    HObject ho_Image;

    // Local control variables
    HTuple hv_Width, hv_Height, hv_WindowHandle;

    if (HDevWindowStack::IsOpen())
        CloseWindow(HDevWindowStack::Pop());
    ReadImage(&ho_Image, "D:/Visual studio 2017/Projects/Halcon-Config/res/2.jpg");
    GetImageSize(ho_Image, &hv_Width, &hv_Height);
    SetWindowAttr("background_color", "black");
    OpenWindow(0, 0, hv_Width, hv_Height, 0, "visible", "", &hv_WindowHandle);
    HDevWindowStack::Push(hv_WindowHandle);
    if (HDevWindowStack::IsOpen())
        SetPart(HDevWindowStack::GetActive(), 0, 0, hv_Height-1, hv_Width-1);
    if (HDevWindowStack::IsOpen())
        DispObj(ho_Image, HDevWindowStack::GetActive());
}

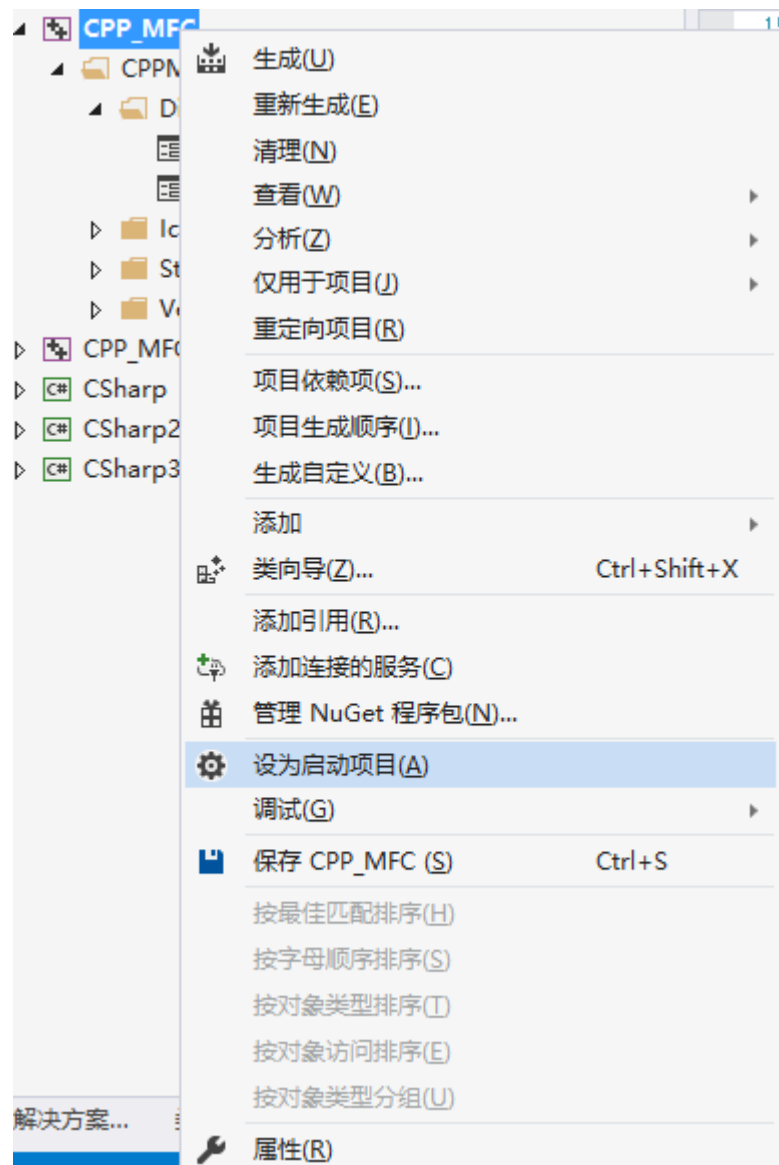
```

17、将上述添加的代码中的两句注释掉，否则会报错，报错原因暂时不清楚，如下图黑色框所示

```
155
156
157 //第二步
158 void CCppMFCDlg::OnBnClickedButton1()
159 {
160     // TODO: 在此添加控件通知处理程序代码
161     // Local iconic variables
162     HObject ho_Image;
163
164     // Local control variables
165     HTuple hv_Width, hv_Height, hv_WindowHandle;
166
167     //第三步
168     /*if (HDevWindowStack::IsOpen())
169     {
170         CloseWindow(HDevWindowStack::Pop());*/
170     ReadImage(&ho_Image, "D:/Visual studio 2017/Projects/Halcon-Config/res/2.jpg");
171     GetImageSize(ho_Image, &hv_Width, &hv_Height);
172     SetWindowAttr("background_color", "black");
173     OpenWindow(0, 0, hv_Width, hv_Height, 0, "visible", "", &hv_WindowHandle);
174     HDevWindowStack::Push(hv_WindowHandle);
175     if (HDevWindowStack::IsOpen())
176     {
177         SetPart(HDevWindowStack::GetActive(), 0, 0, hv_Height - 1, hv_Width - 1);
178     }
179     if (HDevWindowStack::IsOpen())
180     {
181         DispObj(ho_Image, HDevWindowStack::GetActive());
182     }
183 }
```

18、右键单击项目名称，点击“设为启动项目(A)”

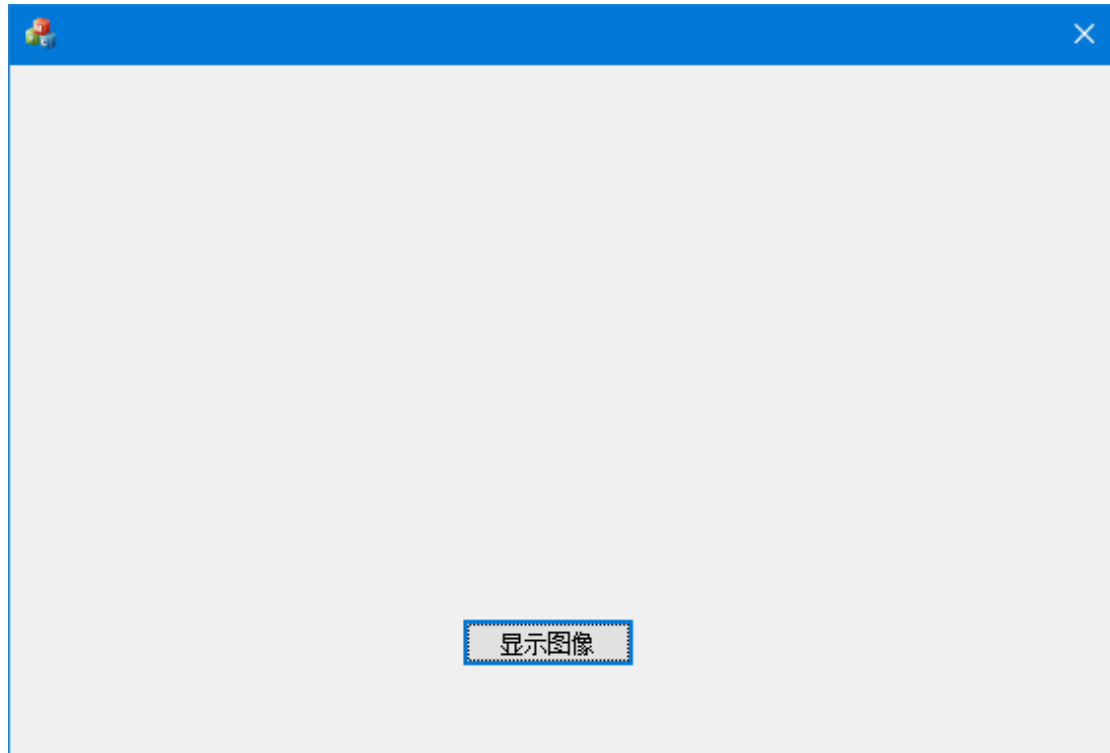




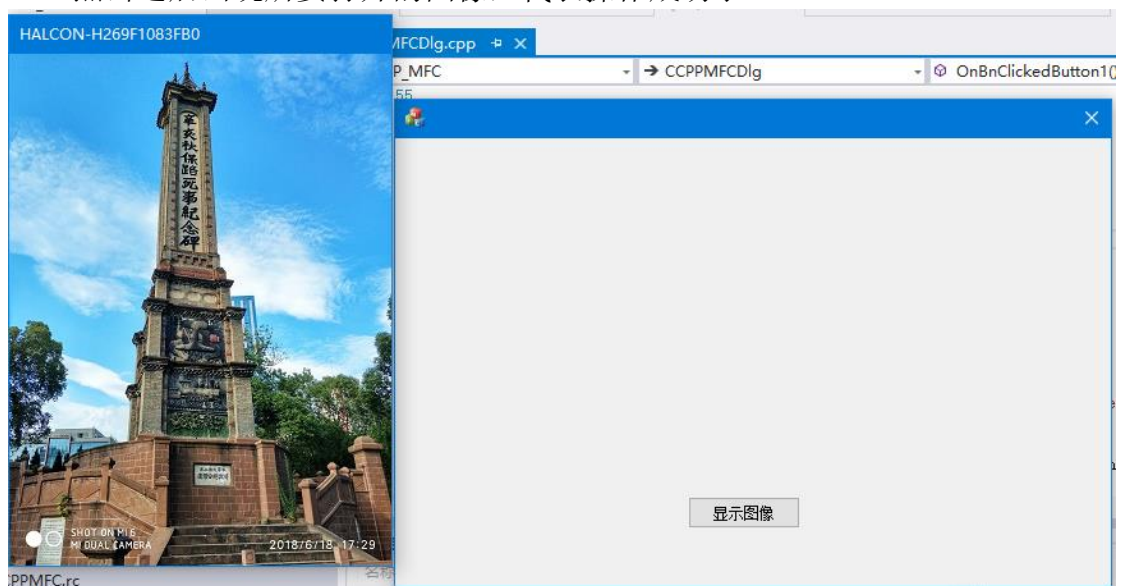
19、本文中操作之后结果如下图所示，点击“本地 Windows 调试器”



20、在下面的图中点击“显示图像”



21、点击之后出现所要打开的图像，代表操作成功了

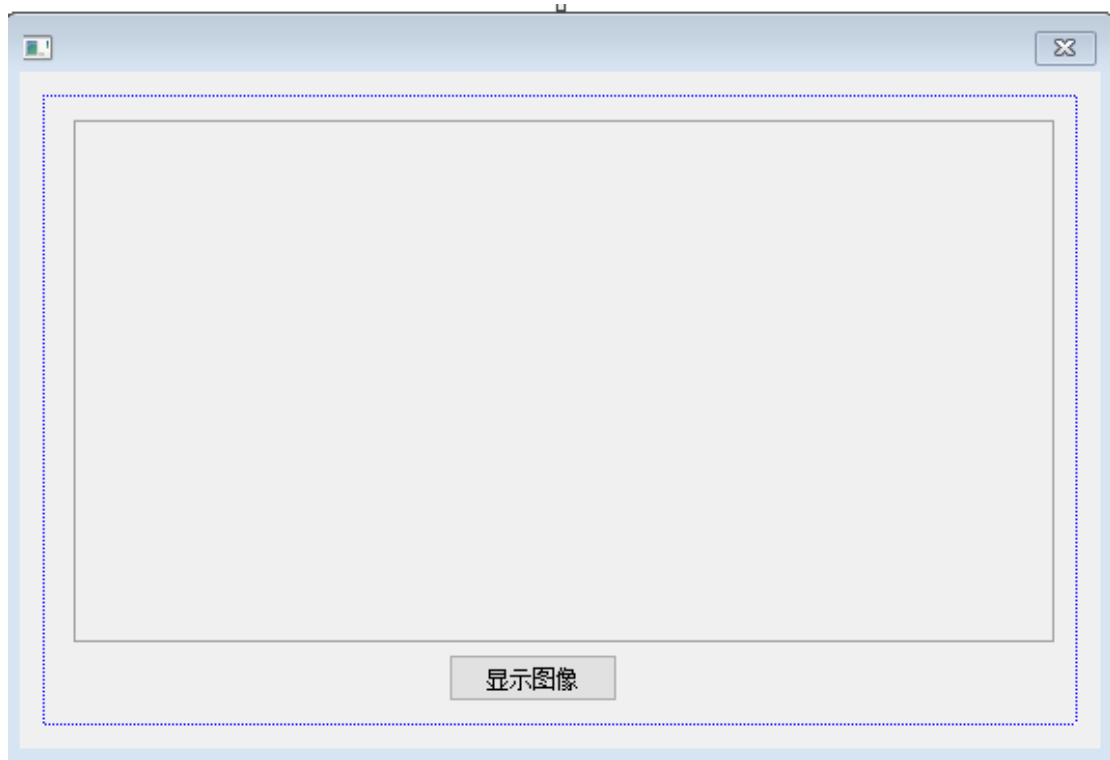


## Halcon 及 MFC 使用的进阶

上述操作时图像显示在一个单独的窗口中，并不能显示在 MFC 窗口中，有必要进行额外的工作将其显示到 MFC 窗口中，如果需要显示到 MFC 窗口中，前面的步骤与基础部分前 14 步一致，下面是另外的操作

22、（接基础部分前 14 步）在第 4 步中打开了 MFC 的界面，先将界面上的无关控件删掉，然后点击工具箱添加 Button 控件，然后点击该控件，右键属性，在 Caption 中将其名字改成“显示图像”，然后添加 Picture Control 控件，如下图所

示：



23、双击“显示图像”，在 `CPP_MFCDlg.cpp`(后面是 `Dlg.cpp`，前面依据个人的工程名称变化) 中 `Button` 控件的事件函数(本步骤中的双击动作会自动产生)中添加 Halcon 生成的 C++ 代码中的 `action` 函数中的内容，如下图所示：

```

void CCPPMFCPlusDlg::OnBnClickedButton1()
{
    // TODO: 在此添加控件通知处理程序代码

    //第二步
    // Local iconic variables
    HObject ho_Image;

    // Local control variables
    HTuple hv_Width, hv_Height, hv_WindowHandle;

    //第三步
    /*if (HDevWindowStack::IsOpen())
        CloseWindow(HDevWindowStack::Pop());*/

    ReadImage(&ho_Image, "D:/Visual studio 2017/Projects/Halcon-Config/res/2.jpg");
    GetImageSize(ho_Image, &hv_Width, &hv_Height);

    //第四步, 其中IDC_STATIC为Picture Control的ID
    HWND hImgWnd = GetDlgItem(IDC_STATIC)->m_hWnd;

    SetWindowAttr("background_color", "black");

    //第五步
    //修改后的语句, 将原来的0改成(LONG)hImgWnd
    OpenWindow(0, 0, hv_Width, hv_Height, (LONG)hImgWnd, "visible", "",
    &hv_WindowHandle);
    //原来的语句
    //OpenWindow(0, 0, hv_Width, hv_Height, 0, "visible", "", &hv_WindowHandle);

    HDevWindowStack::Push(hv_WindowHandle);
    if (HDevWindowStack::IsOpen())
        SetPart(HDevWindowStack::GetActive(), 0, 0, hv_Height - 1, hv_Width - 1);
    if (HDevWindowStack::IsOpen())
        DispObj(ho_Image, HDevWindowStack::GetActive());
}

```

下图所示就是 Halcon 生成的 CPP 源程序中的 action 函数

```

// Main procedure
void action()
{
    // Local iconic variables
    HObject ho_Image;

    // Local control variables
    HTuple hv_Width, hv_Height, hv_WindowHandle;

    if (HDevWindowStack::IsOpen())
        CloseWindow(HDevWindowStack::Pop());
    ReadImage(&ho_Image, "D:/Visual studio 2017/Projects/Halcon-Config/res/2.jpg");
    GetImageSize(ho_Image, &hv_Width, &hv_Height);
    SetWindowAttr("background_color", "black");
    OpenWindow(0,0,hv_Width,hv_Height,0,"visible","",&hv_WindowHandle);
    HDevWindowStack::Push(hv_WindowHandle);
    if (HDevWindowStack::IsOpen())
        SetPart(HDevWindowStack::GetActive(),0, 0, hv_Height-1, hv_Width-1);
    if (HDevWindowStack::IsOpen())
        DispObj(ho_Image, HDevWindowStack::GetActive());
}

```

24、将下图中黑色框框出的语句注释掉，否则会报错，图中第三步

```

void CCPPMFCPlusDlg::OnBnClickedButton1()
{
    // TODO: 在此添加控件通知处理程序代码

    //第二步
    // Local iconic variables
    HObject ho_Image;

    // Local control variables
    HTuple hv_Width, hv_Height, hv_WindowHandle;

    //第三步
    /*if (HDevWindowStack::IsOpen())
        CloseWindow(HDevWindowStack::Pop());*/

    ReadImage(&ho_Image, "D:/Visual studio 2017/Projects/Halcon-Config/res/2.jpg");
    GetImageSize(ho_Image, &hv_Width, &hv_Height);

    //第四步, 其中IDC_STATIC为Picture Control的ID
    HWND hImgWnd = GetDlgItem(IDC_STATIC)->m_hWnd;

    SetWindowAttr("background_color", "black");

    //第五步
    //修改后的语句, 将原来的0改成(LONG)hImgWnd
    OpenWindow(0, 0, hv_Width, hv_Height, (LONG)hImgWnd, "visible", "",
    &hv_WindowHandle);
    //原来的语句
    //OpenWindow(0, 0, hv_Width, hv_Height, 0, "visible", "", &hv_WindowHandle);

    HDevWindowStack::Push(hv_WindowHandle);
    if (HDevWindowStack::IsOpen())
        SetPart(HDevWindowStack::GetActive(), 0, 0, hv_Height - 1, hv_Width - 1);
    if (HDevWindowStack::IsOpen())
        DispObj(ho_Image, HDevWindowStack::GetActive());
}

```

25、为了将图像显示到 MFC 窗体中的 Picture Control 控件中，需要得到其句柄，添加下图蓝色框框出的语句

```

//第四步, 其中IDC_STATIC为Picture Control的ID
HWND hImgWnd = GetDlgItem(IDC_STATIC)->m_hWnd;

```

如图中第四步

```

void CCPPMFCPlusDlg::OnBnClickedButton1()
{
    // TODO: 在此添加控件通知处理程序代码

    //第二步
    // Local iconic variables
    HObject ho_Image;

    // Local control variables
    HTuple hv_Width, hv_Height, hv_WindowHandle;

    //第三步
    /*if (HDevWindowStack::IsOpen())
        CloseWindow(HDevWindowStack::Pop());*/

    ReadImage(&ho_Image, "D:/Visual studio 2017/Projects/Halcon-Config/res/2.jpg");
    GetImageSize(ho_Image, &hv_Width, &hv_Height);

    //第四步, 其中IDC_STATIC为Picture Control的ID
    HWND hImgWnd = GetDlgItem(IDC_STATIC)->m_hWnd;

    SetWindowAttr("background_color", "black");

    //第五步
    //修改后的语句, 将原来的0改成(LONG)hImgWnd
    OpenWindow(0, 0, hv_Width, hv_Height, (LONG)hImgWnd, "visible", "",
    &hv_WindowHandle);
    //原来的语句
    //OpenWindow(0, 0, hv_Width, hv_Height, 0, "visible", "", &hv_WindowHandle);

    HDevWindowStack::Push(hv_WindowHandle);
    if (HDevWindowStack::IsOpen())
        SetPart(HDevWindowStack::GetActive(), 0, 0, hv_Height - 1, hv_Width - 1);
    if (HDevWindowStack::IsOpen())
        DispObj(ho_Image, HDevWindowStack::GetActive());
}

```

26、修改原先的打开窗口的语句，将其中的第三个 0 改成 (LONG)hImgWnd，如图中黄色线和黄色框标注的地方

```

void CCPPMFCPlusDlg::OnBnClickedButton1()
{
    // TODO: 在此添加控件通知处理程序代码

    //第二步
    // Local iconic variables
    HObject ho_Image;

    // Local control variables
    HTuple hv_Width, hv_Height, hv_WindowHandle;

    //第三步
    /*if (HDevWindowStack::IsOpen())
        CloseWindow(HDevWindowStack::Pop());*/

    ReadImage(&ho_Image, "D:/Visual studio 2017/Projects/Halcon-Config/res/2.jpg");
    GetImageSize(ho_Image, &hv_Width, &hv_Height);

    //第四步, 其中IDC_STATIC为Picture Control的ID
    HWND hImgWnd = GetDlgItem(IDC_STATIC)->m_hWnd;

    SetWindowAttr("background_color", "black");

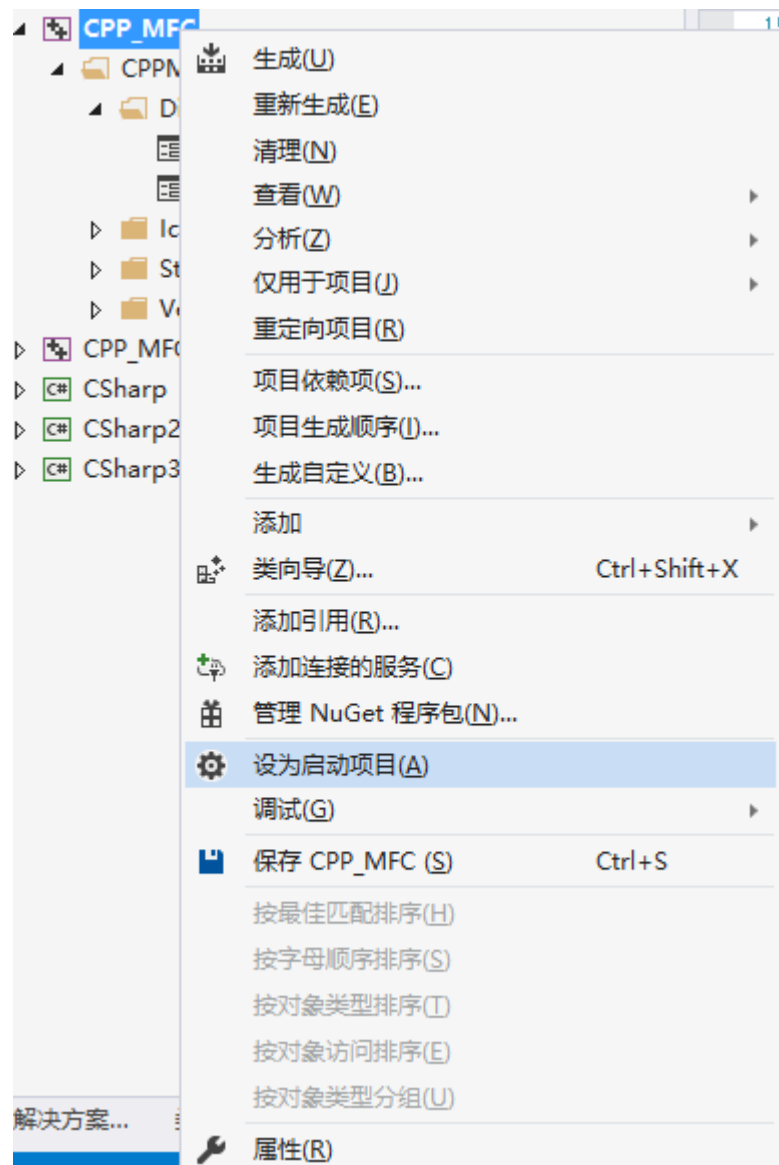
    //第五步
    //修改后的语句, 将原来的0改成(LONG)hImgWnd
    OpenWindow(0, 0, hv_Width, hv_Height, (LONG)hImgWnd, "visible", "",
    &hv_WindowHandle);
    //原来的语句
    //OpenWindow(0, 0, hv_Width, hv_Height, 0, "visible", "", &hv_WindowHandle);

    HDevWindowStack::Push(hv_WindowHandle);
    if (HDevWindowStack::IsOpen())
        SetPart(HDevWindowStack::GetActive(), 0, 0, hv_Height - 1, hv_Width - 1);
    if (HDevWindowStack::IsOpen())
        DispObj(ho_Image, HDevWindowStack::GetActive());
}

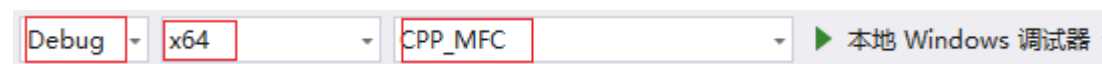
```

27、右键单击项目名称，点击“设为启动项目(A)”

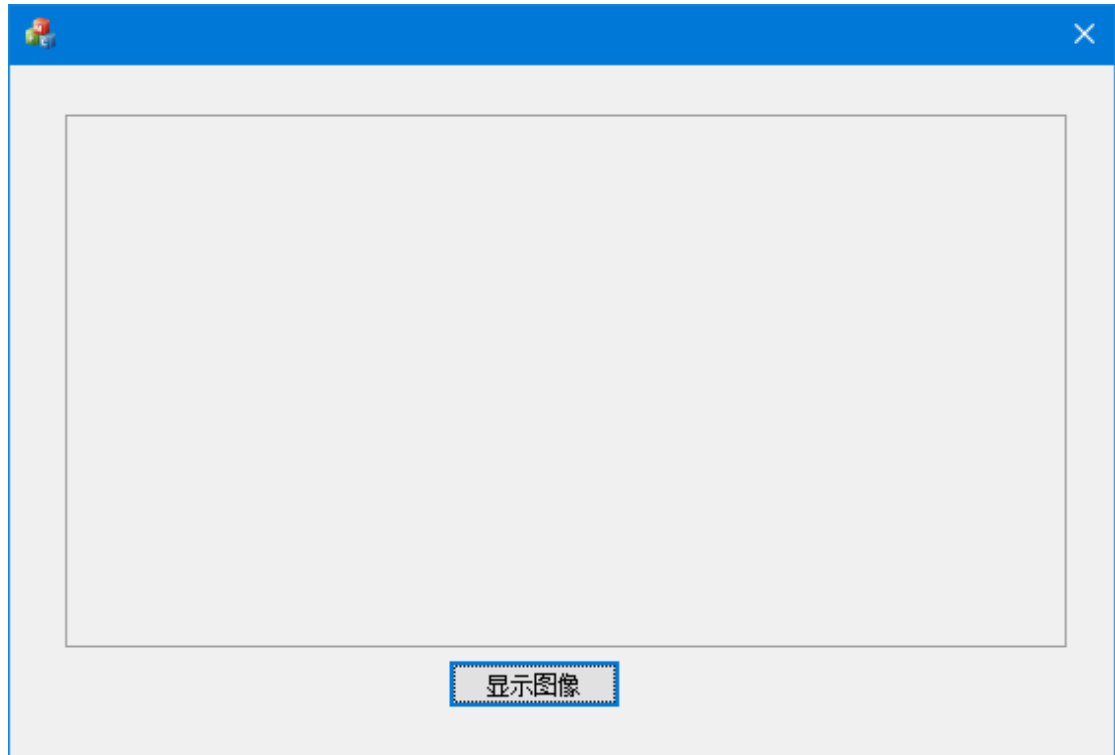




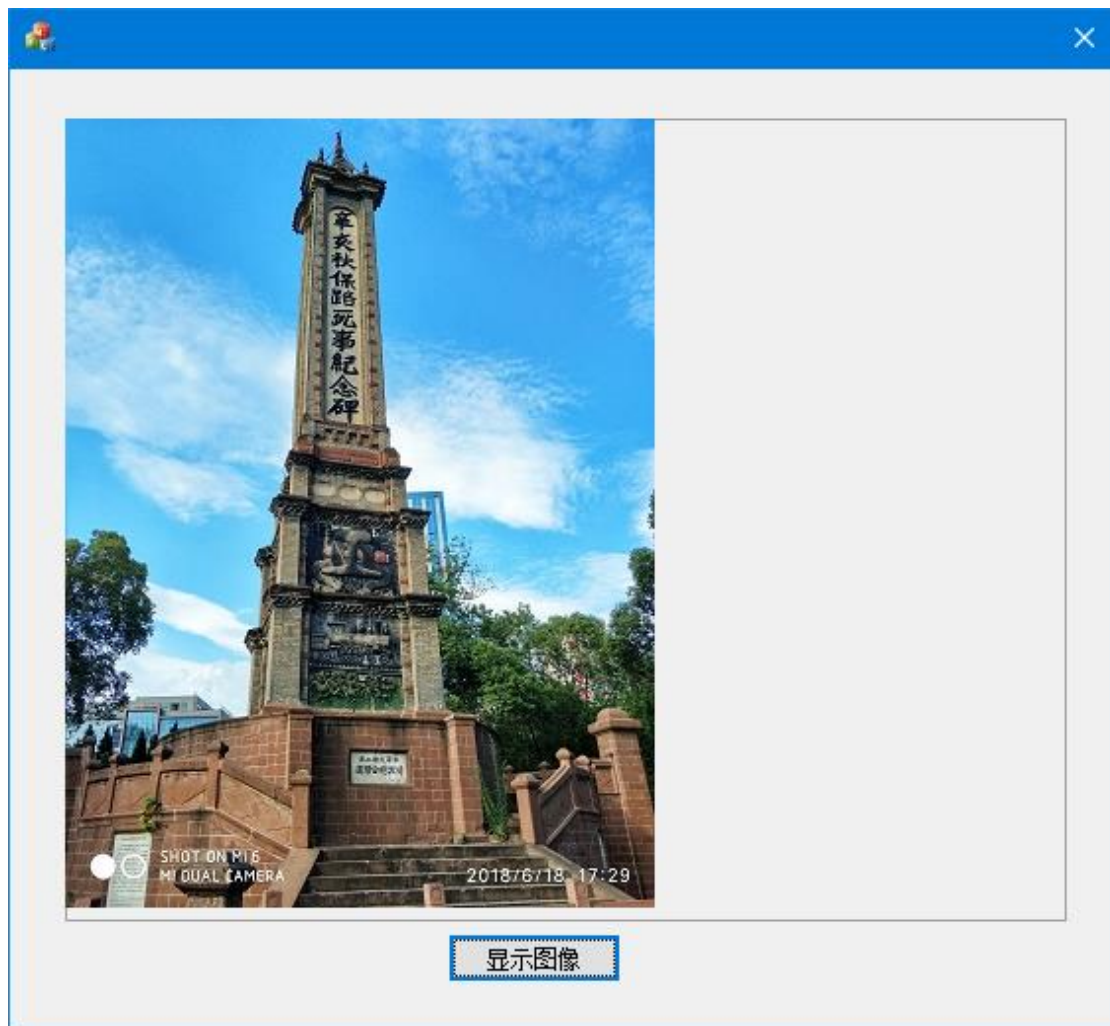
28、本文中操作之后结果如下图所示，点击“本地 Windows 调试器”



29、在下面的图中点击“显示图像”



30、点击之后出现所要打开的图像，代表操作成功了

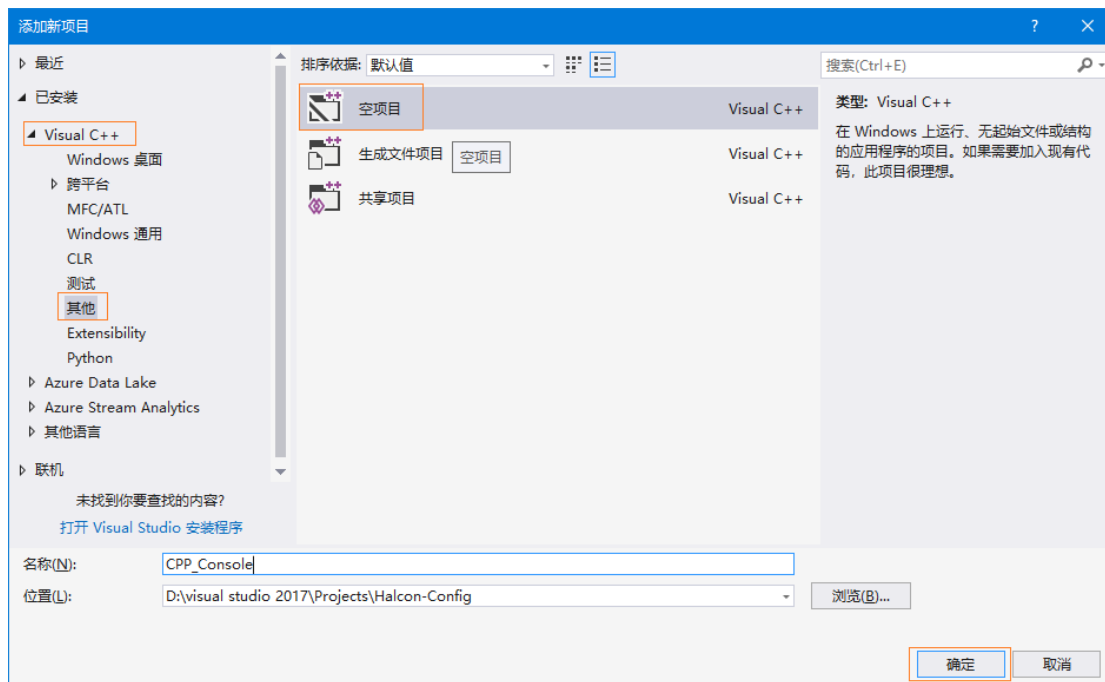


图像中的 Picture Control 暂时未找到怎么设置可以自适应图像大小添加滚动条

## Halcon 代码在 C++控制台程序中的应用

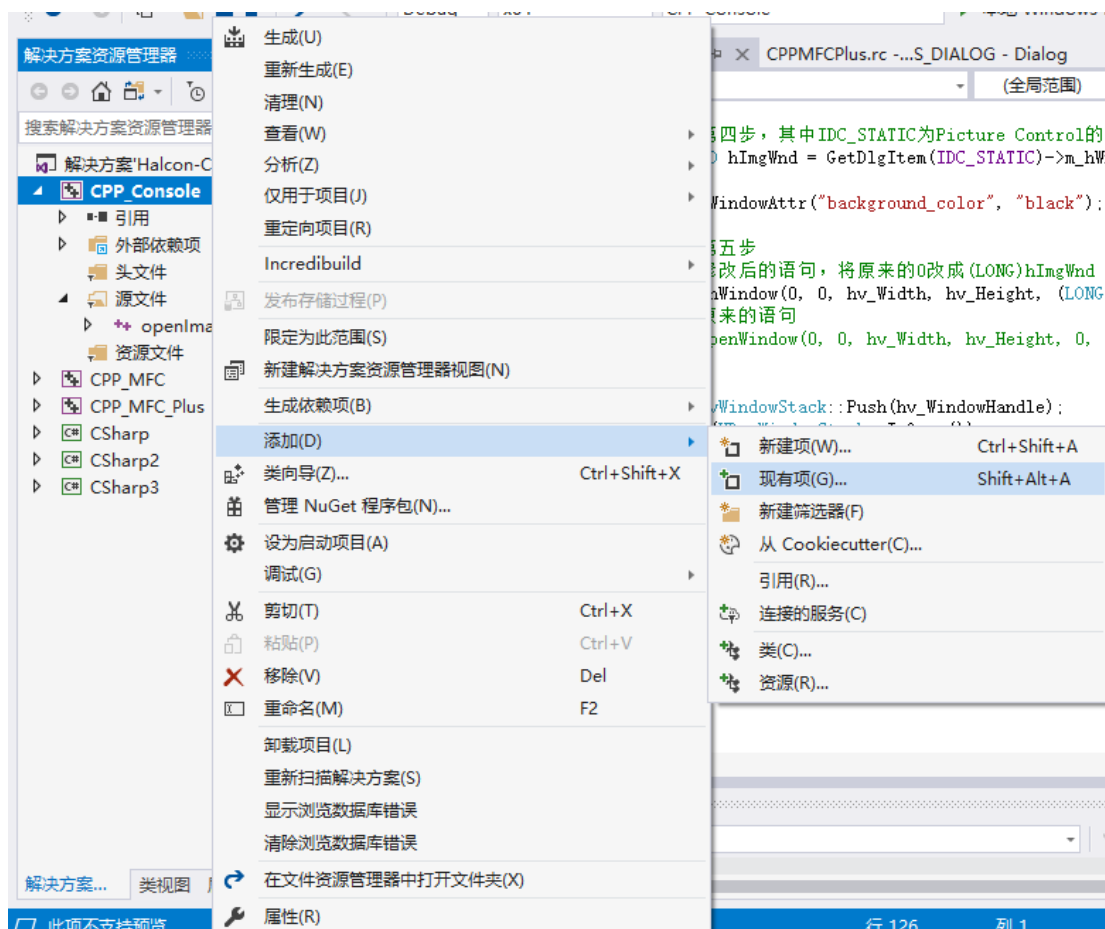
有时候不需要利用 MFC 窗口，只需要一个控制台程序即可，操作步骤如下

1)、如下图所示，新建一个 Visual C++空项目

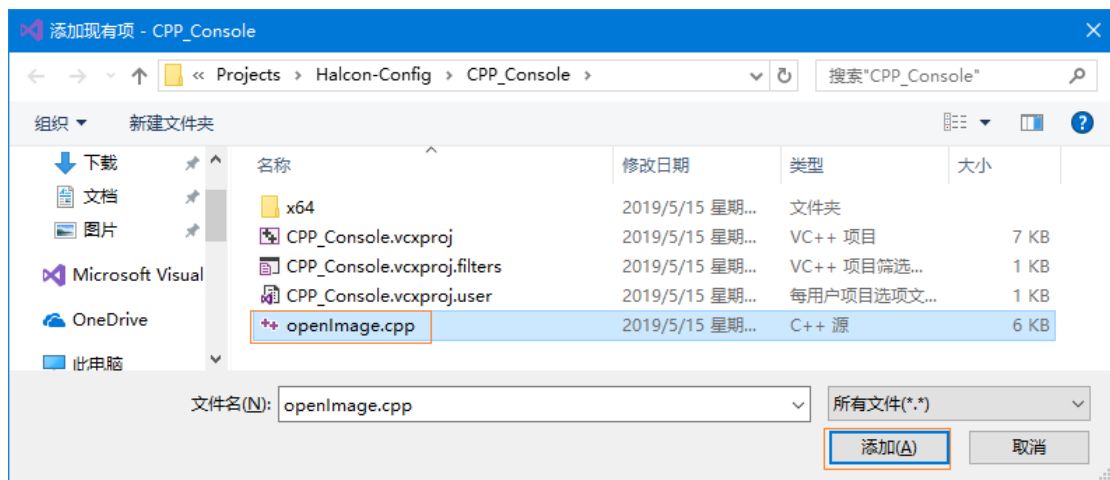


2)、Halcon 及 C++的配置参考前面的第 5 到第 12 步

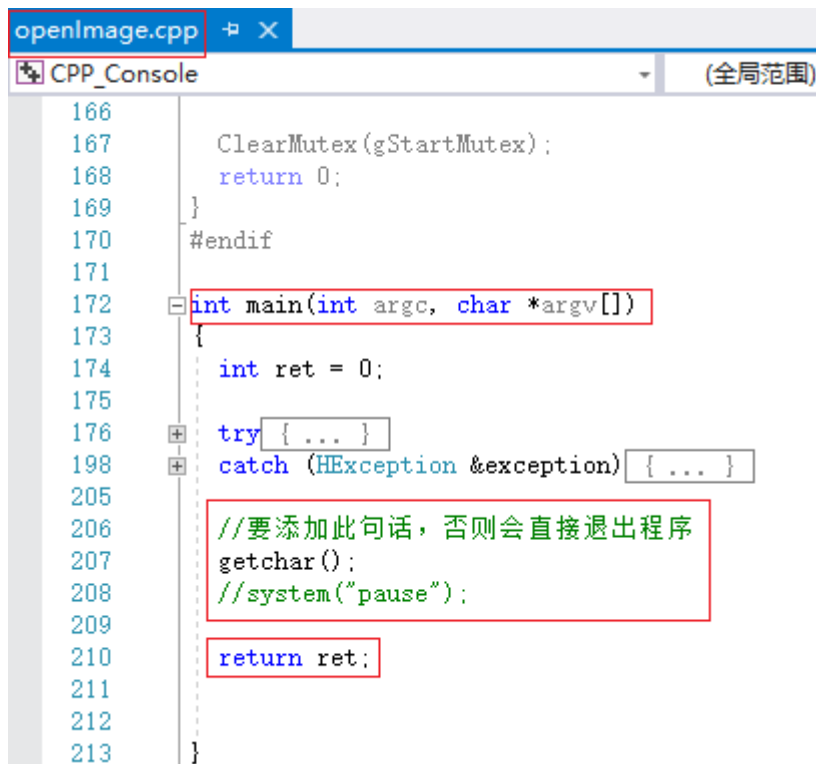
3)、右键单击工程名，然后点击“添加(D)”中的“现有项(G)”，选中 Halcon 导出的 C++源程序



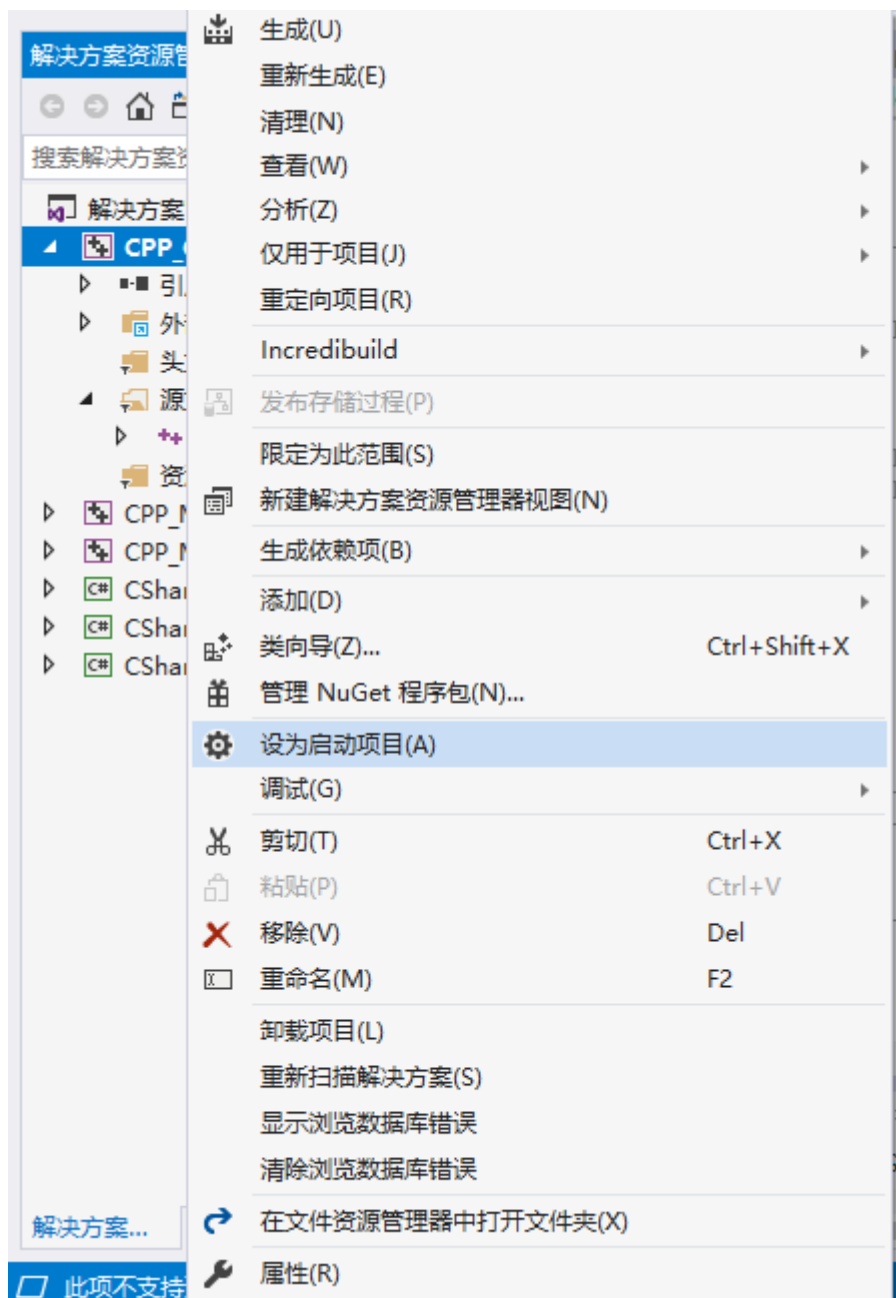
4)、如下图所示，选中源程序，点击“添加(A)”



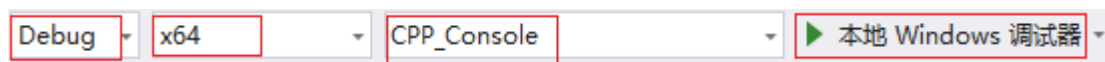
5)、找到导入的源程序，如下图所示在 main 函数最后面的语句 `return ret;` 前面添加代码 `getchar();` 或者 `system("pause");`，否则运行程序时会一闪而过



6)、右键单击项目名称，点击“设为启动项目(A)”



7)、本文中操作之后结果如下图所示，点击“本地 Windows 调试器”



8)、运行结果如下图所示



9)、在黑色的控制台上面输入任意键即可退出程序