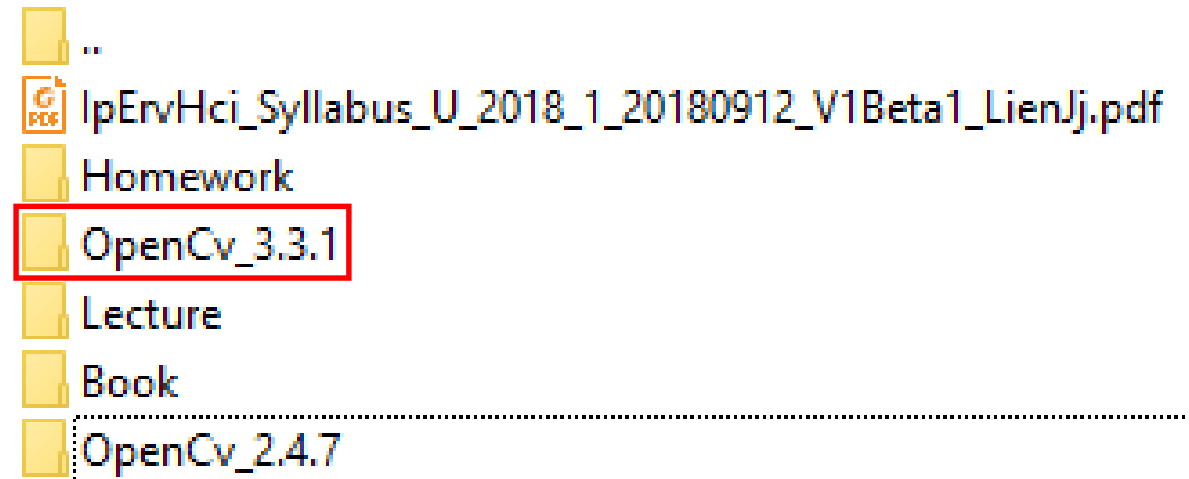


OpenCv & MFC 教學

一、OpenCv 設定教學

步驟1：下載 opencv

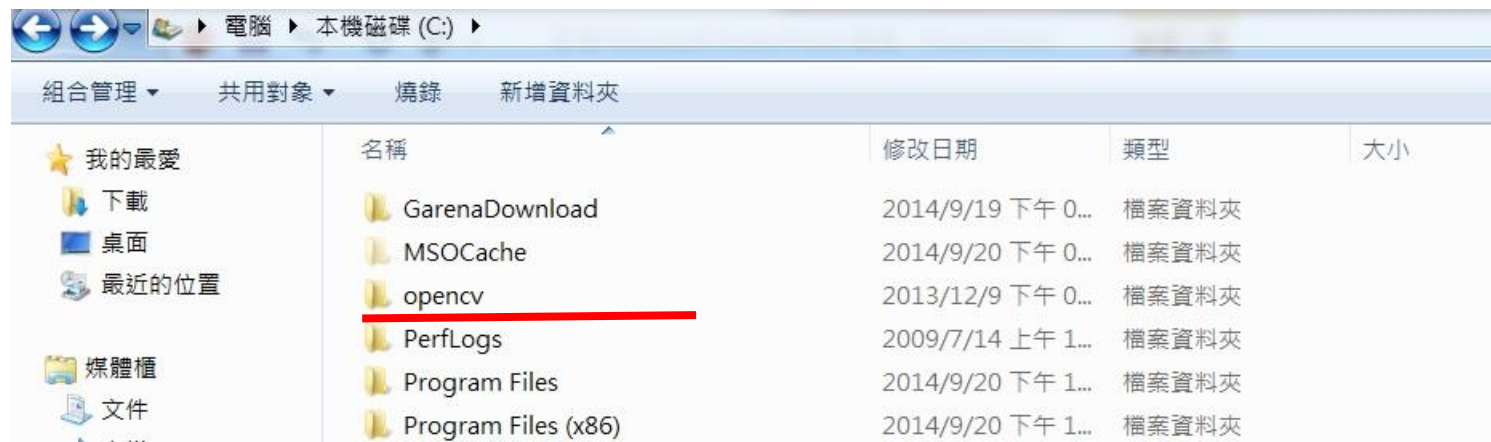
- 登入 FTP (140.116.154.1)
 - 帳號：opencv2018
 - 密碼：opencv2018
- 有下圖的檔案，並全部下載



Filename	Filesize	Filetype	Last modified
..			
opencv-3.3.1-vc14.exe	148,901,170	Application	10/5/2018 3:54:42 PM
PathEditor.exe	287,744	Application	10/5/2018 3:48:58 PM
OpenCv_Mfc_20160921_V2_Carol.pptx	3,360,273	Microsoft PowerPo...	10/5/2018 3:48:57 PM

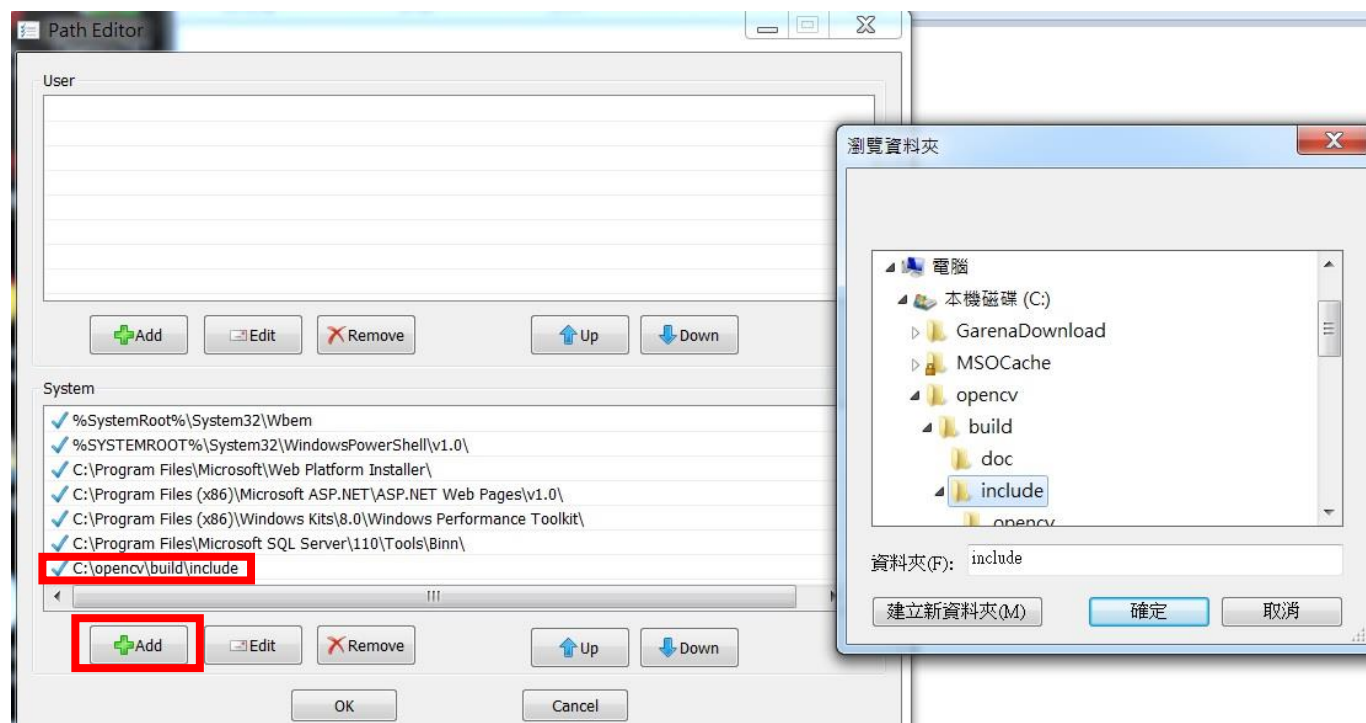
步驟2：安裝 opencv

- 打開 opencv-3.3.1-vc14.exe後會產生 opencv 資料夾。
- 將 opencv 資料夾丟入C槽(或D槽下都行)，只是等等路徑要改而已。



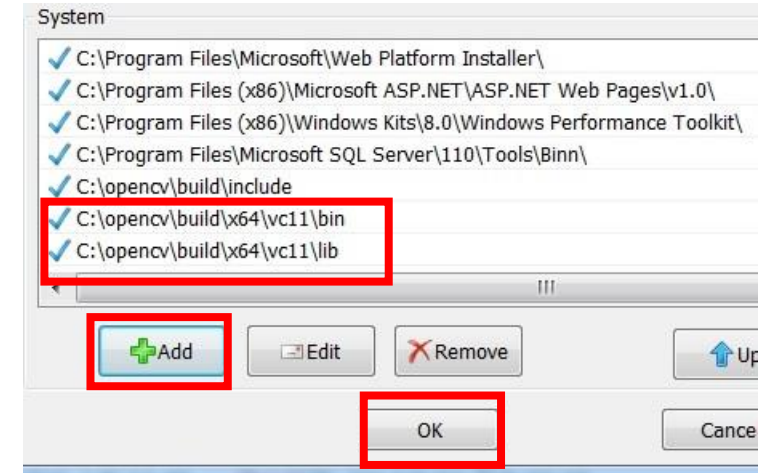
步驟3-1：設定路徑

- 右鍵系統管理員打開 PathEditor.exe
- 底下Add加入→ C:\opencv\build\include



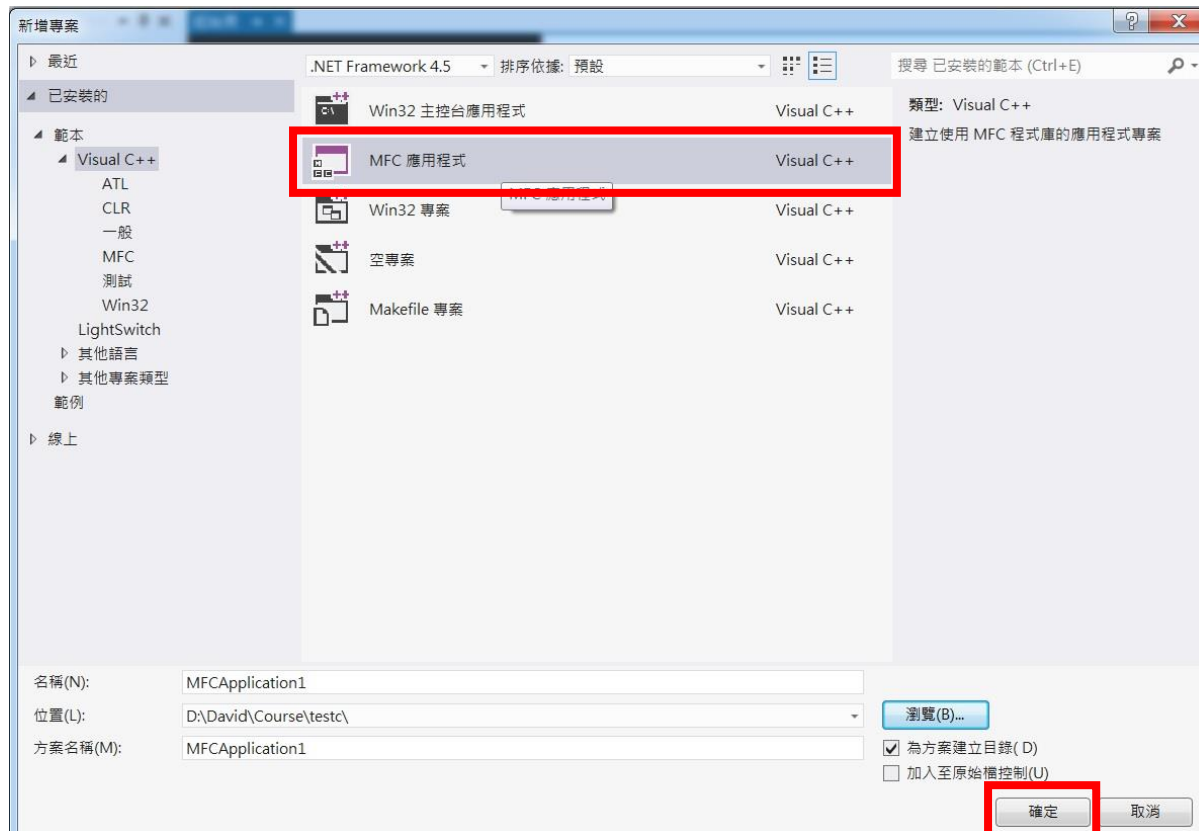
步驟3-2：設定路徑

- 電腦是 64 位元使用 visual 2015 底下 Add 加入
 - C:\opencv\build\x64\vc14\bin
 - C:\opencv\build\x64\vc14\lib
- 電腦是 32 位元使用 visual 2015 底下 Add 加入
 - C:\opencv\build\x86\vc14\bin
 - C:\opencv\build\x86\vc14\lib



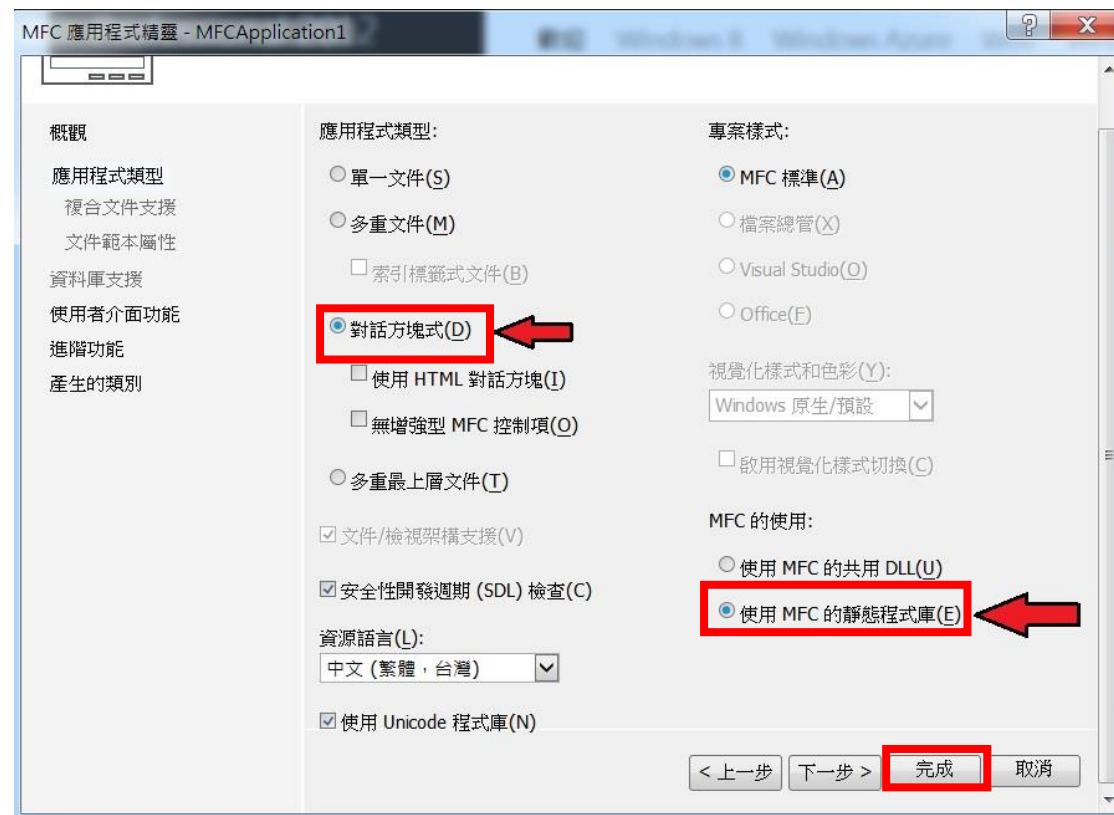
步驟4：MFC設定

- 先開啟新專案，選MFC應用程式



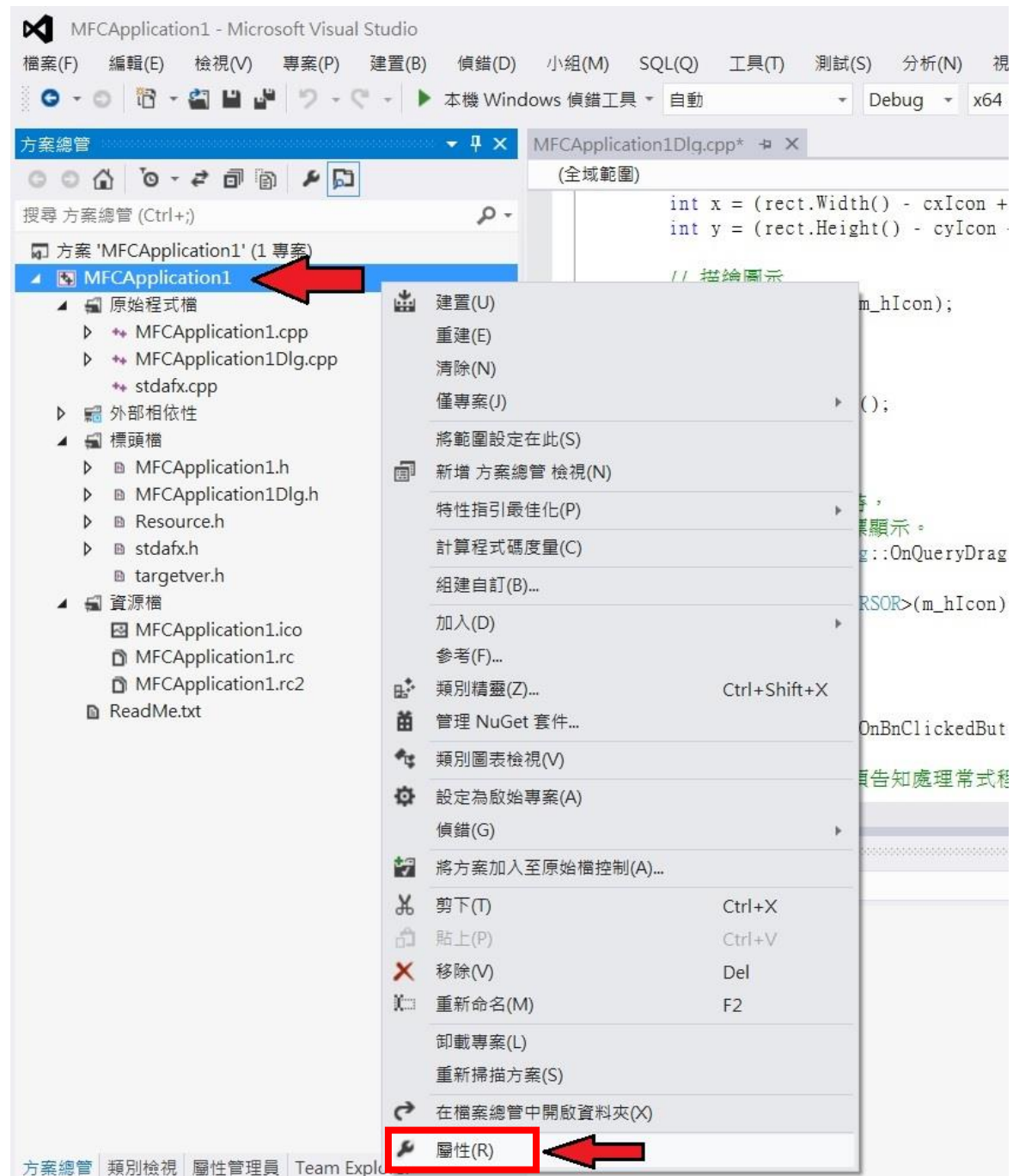
步驟4：MFC設定

- 下一步後點到下列頁面，選對話方塊式和使用MFC的靜態程式庫，最後按下完成即可。



步驟4：MFC 設定

- 點選專題名稱，右鍵點屬性

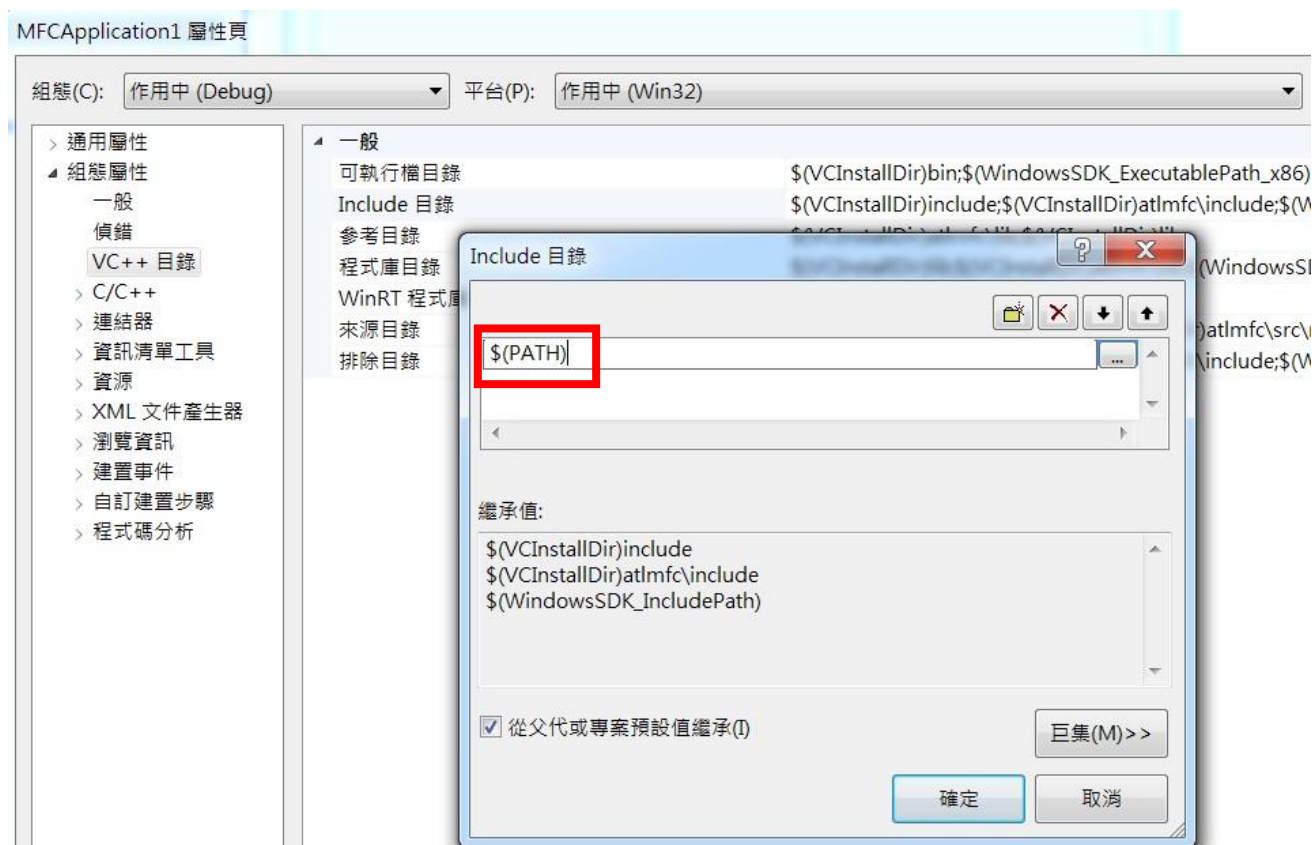
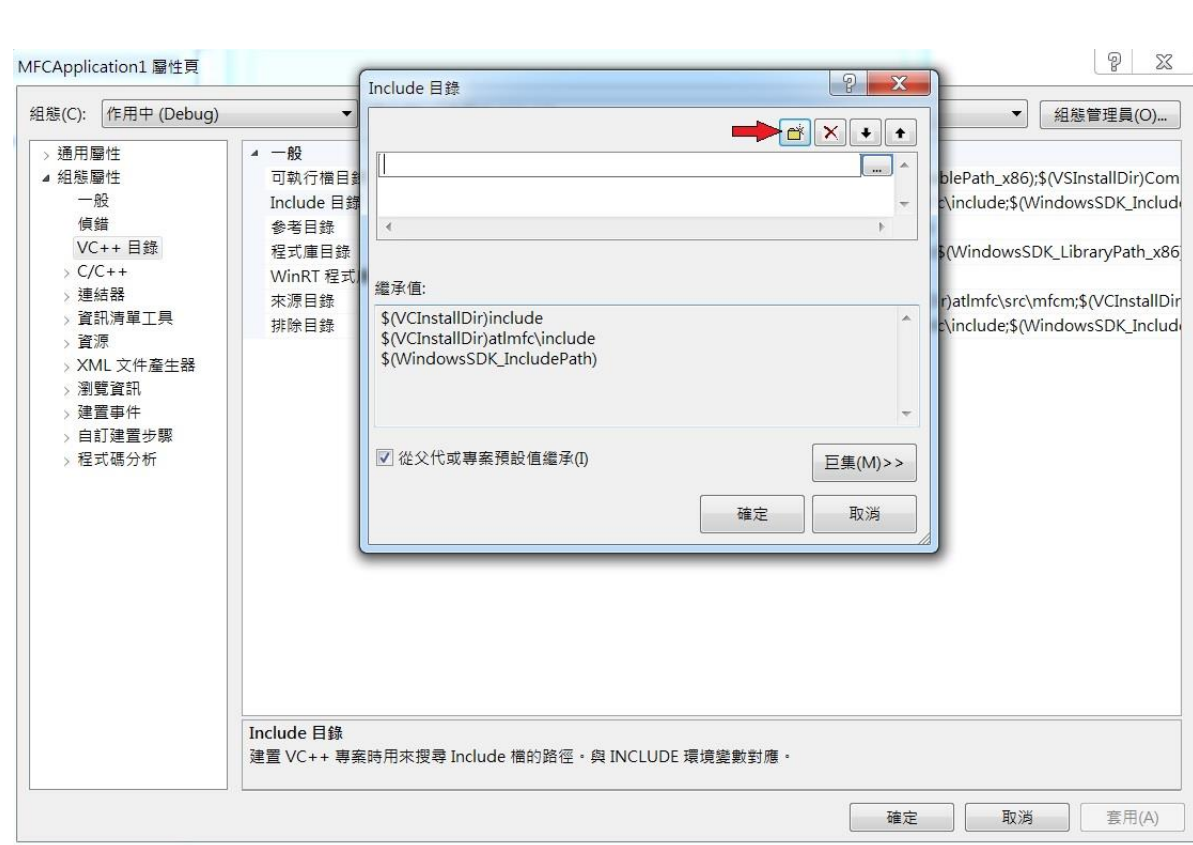


- 打VC++目錄，點Include目錄，點編輯



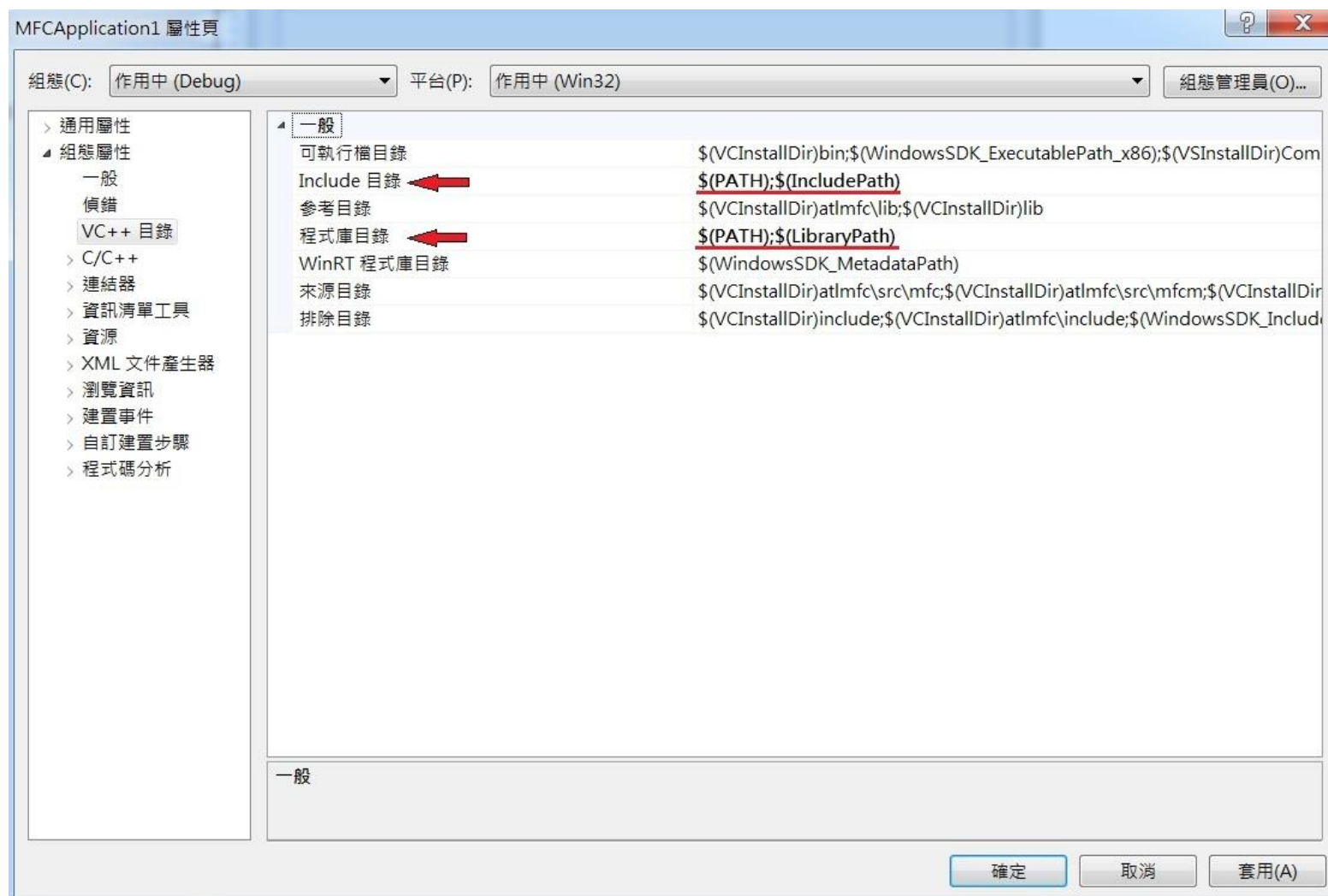
步驟4：MFC設定

- 在裡面新增一條，輸入 **\$(PATH)**。



步驟4：MFC設定

- 在程式庫目錄也加一條 $$(PATH)$ 最後應該會變成右圖這樣。



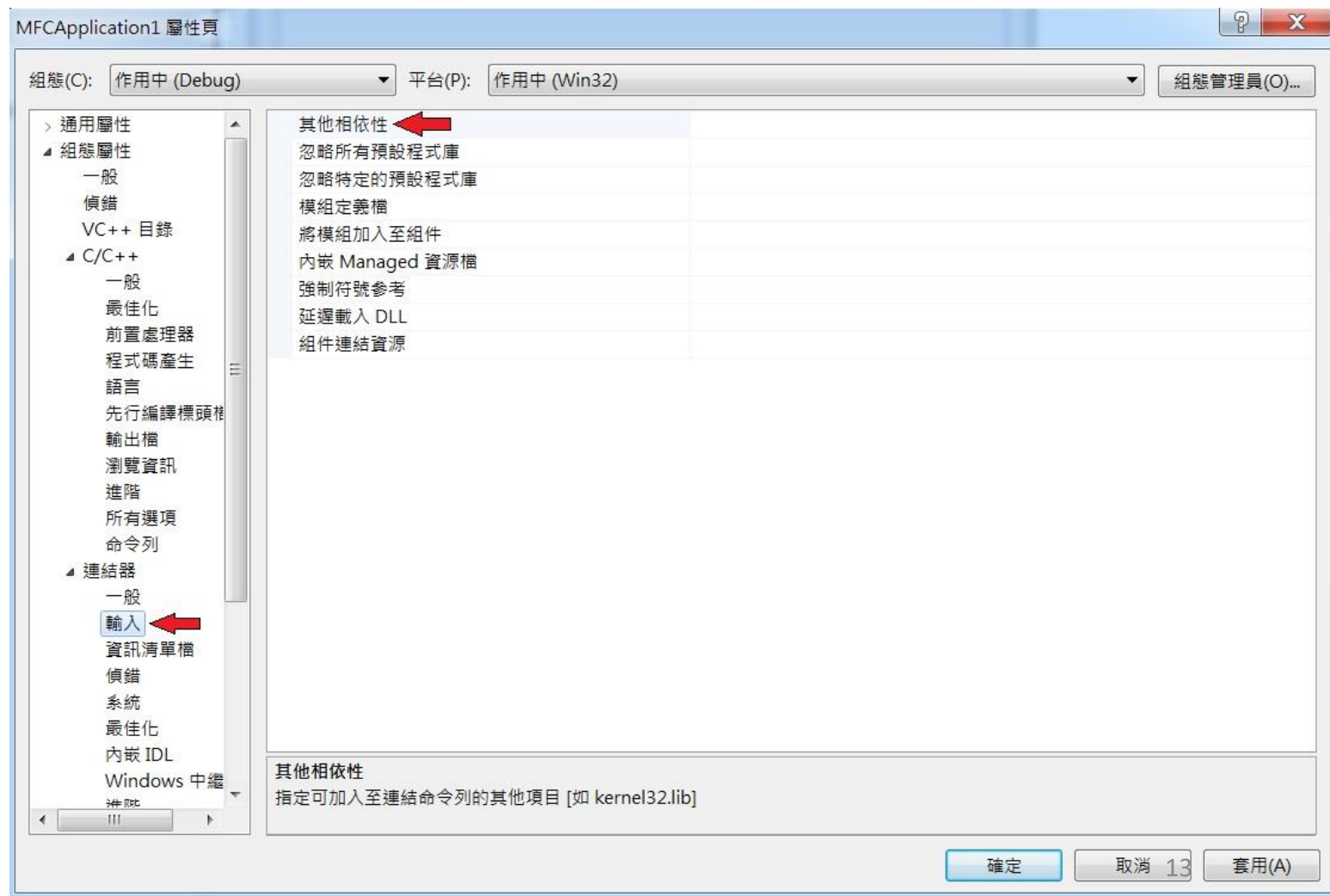
步驟4：MFC設定

- 最後點連結器

→ 輸入

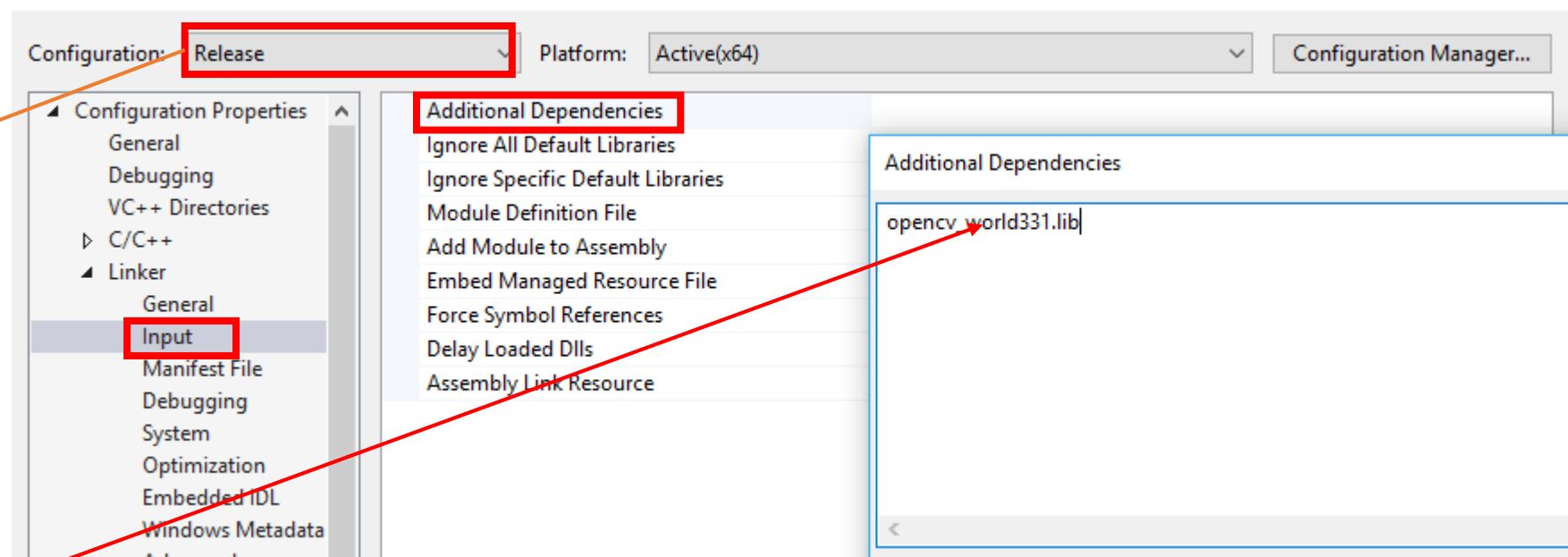
→ 其他相依性

→ 新增

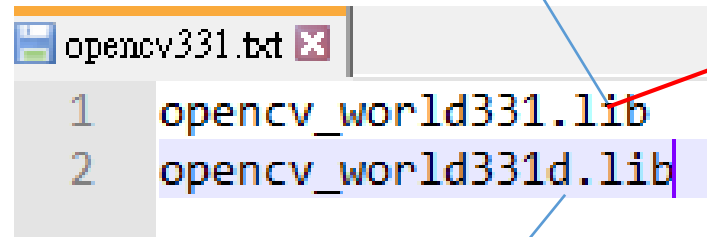


步驟4：MFC設定

- 打開opencv331.txt，把內容複製貼上。
- Only add opencv_world331.lib



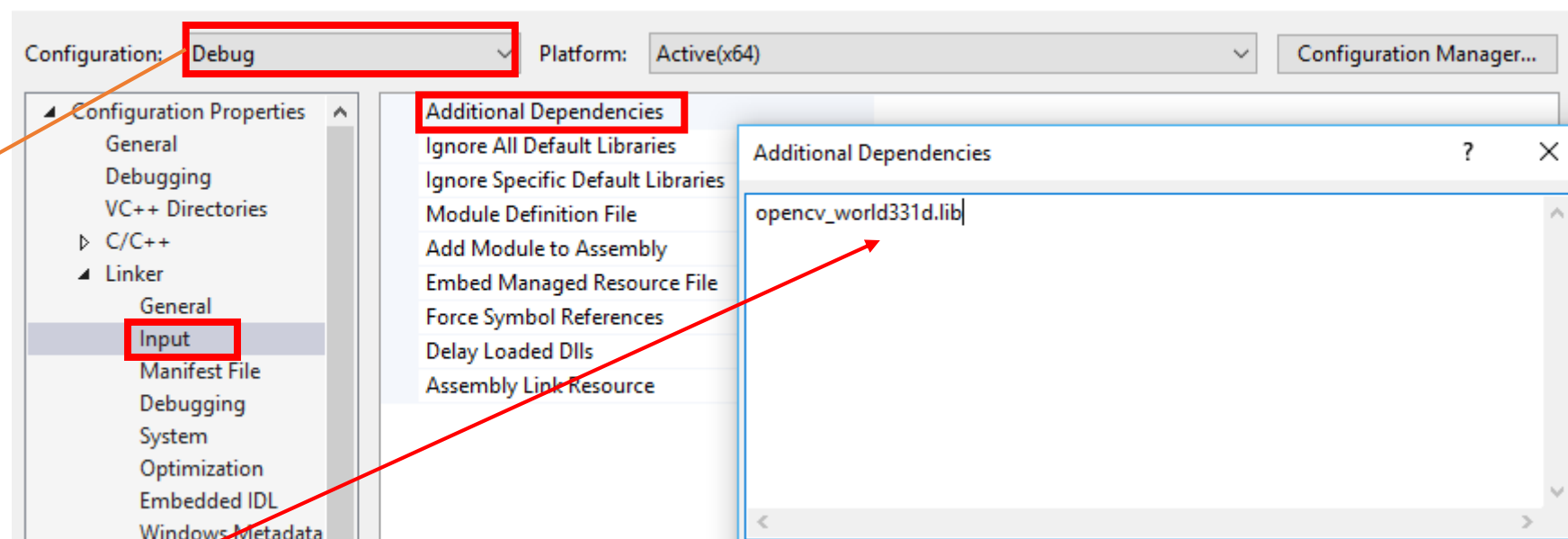
Libs for release mode



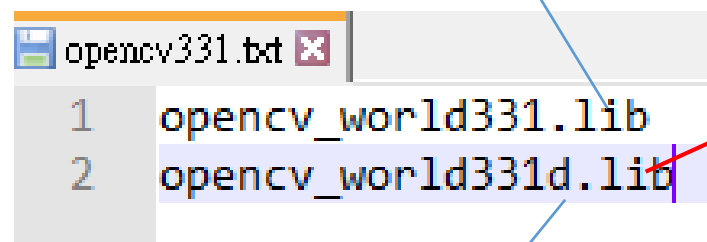
Libs for debug mode

步驟4：MFC設定

- 打開opencv331.txt，把內容複製貼上。
- Only add **opencv_world331d.lib**



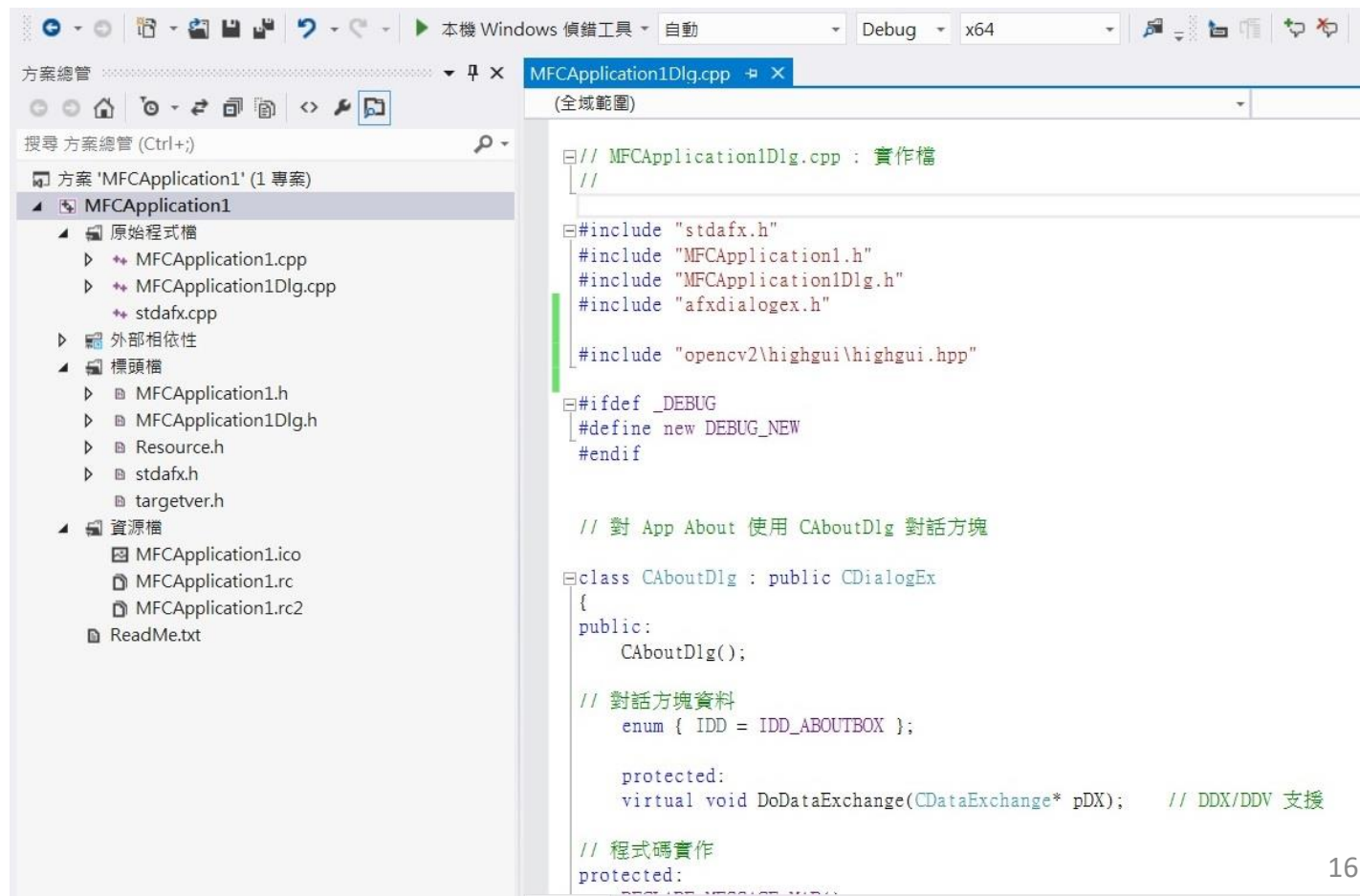
Libs for release mode



Libs for debug mode

步驟5：設定完了


- 終於可以開始操作啦 >0<



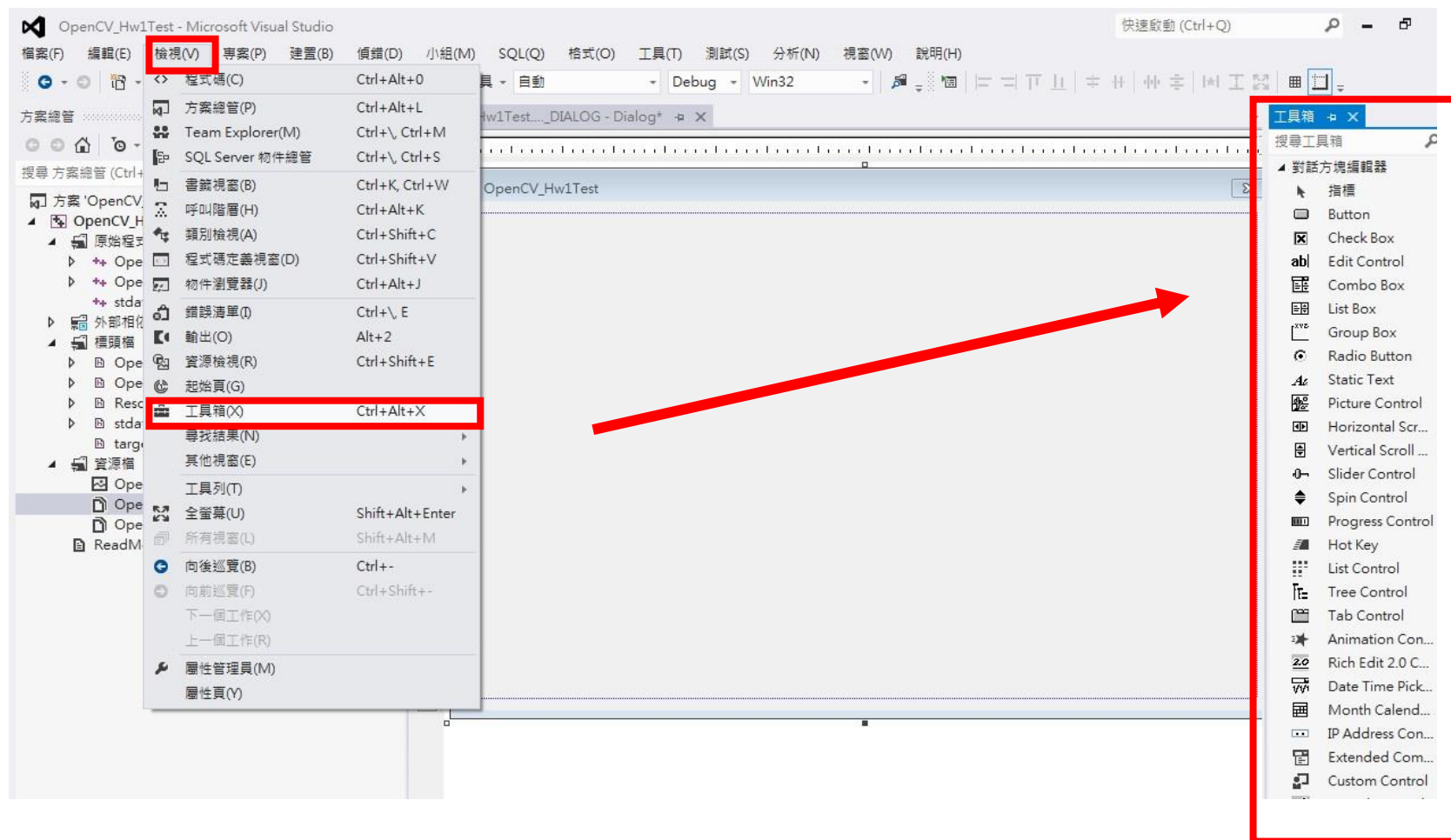
二、MFC 教學

打開工具箱

- 檢視 → 工具箱

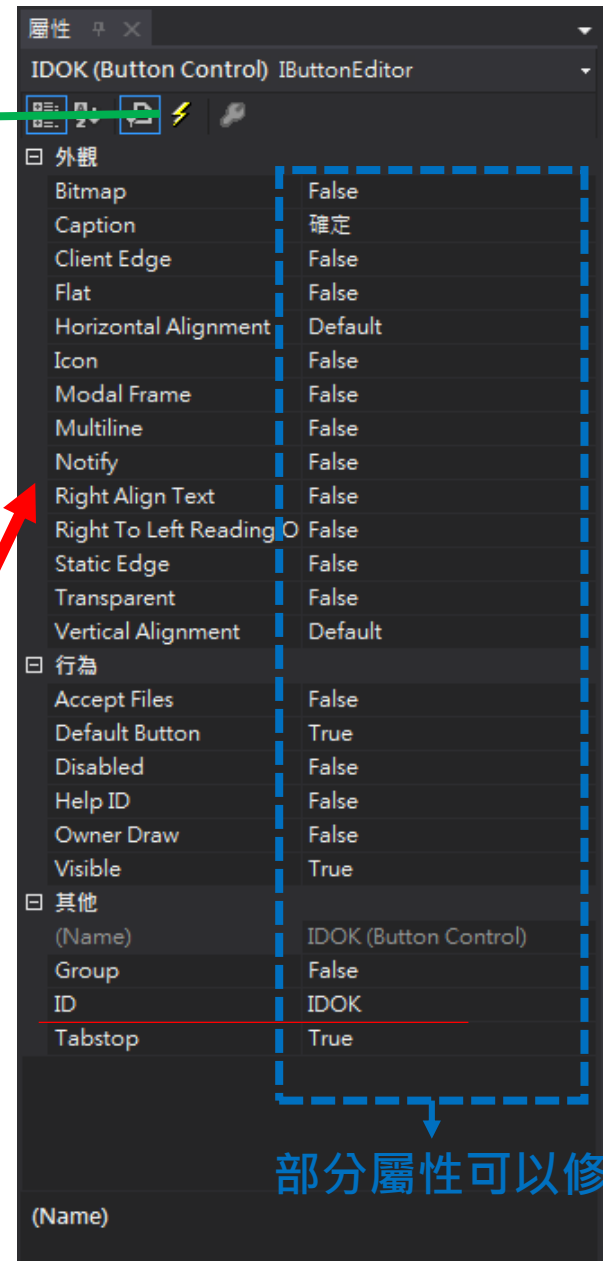
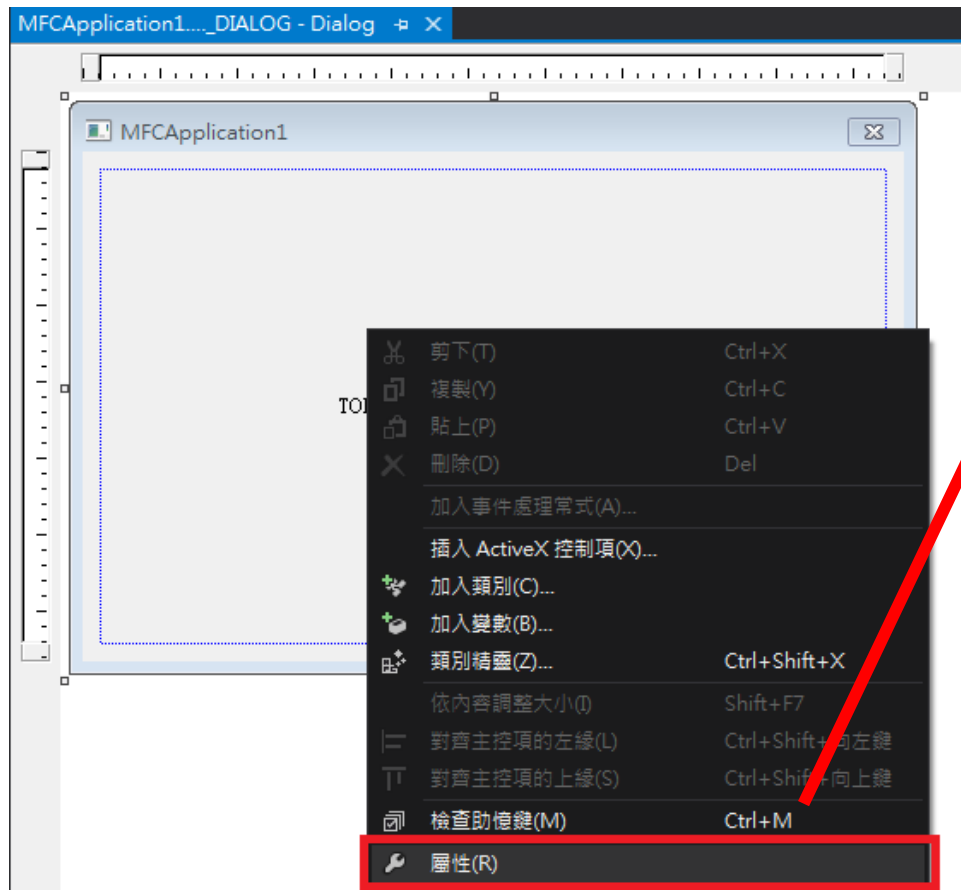
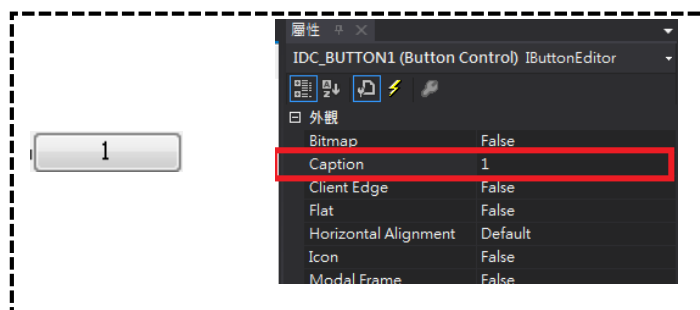
- 按下 針的圖案()
可以將工具箱固定在畫面上。

- 將想要新增的元件拖曳至 DIALOG 畫面即可。



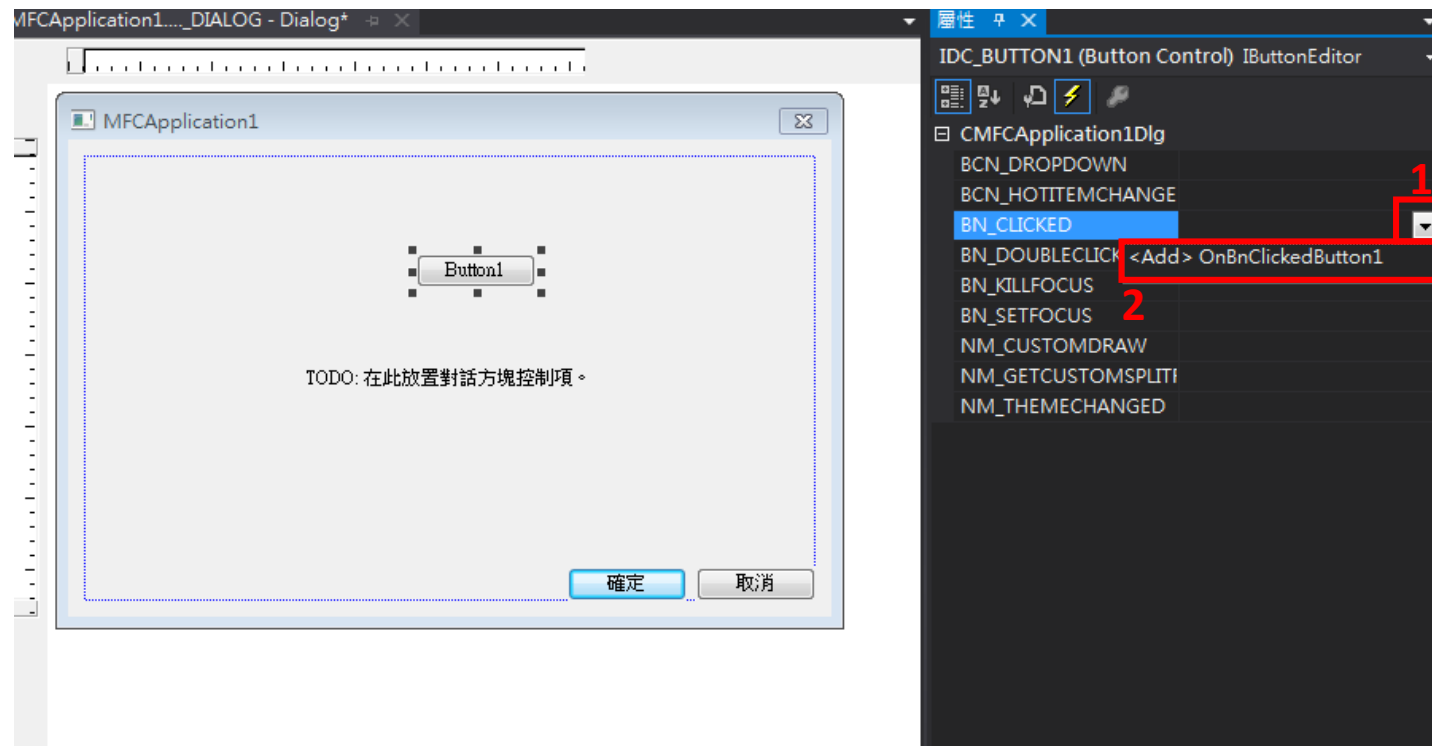
屬性

- DIALOG 畫面右鍵，選擇屬性
- 元件可以這這裡修改屬性、新增事件等。
- 例如：更改按鈕的Caption欄位，可以看到按鈕上的字改變了。→



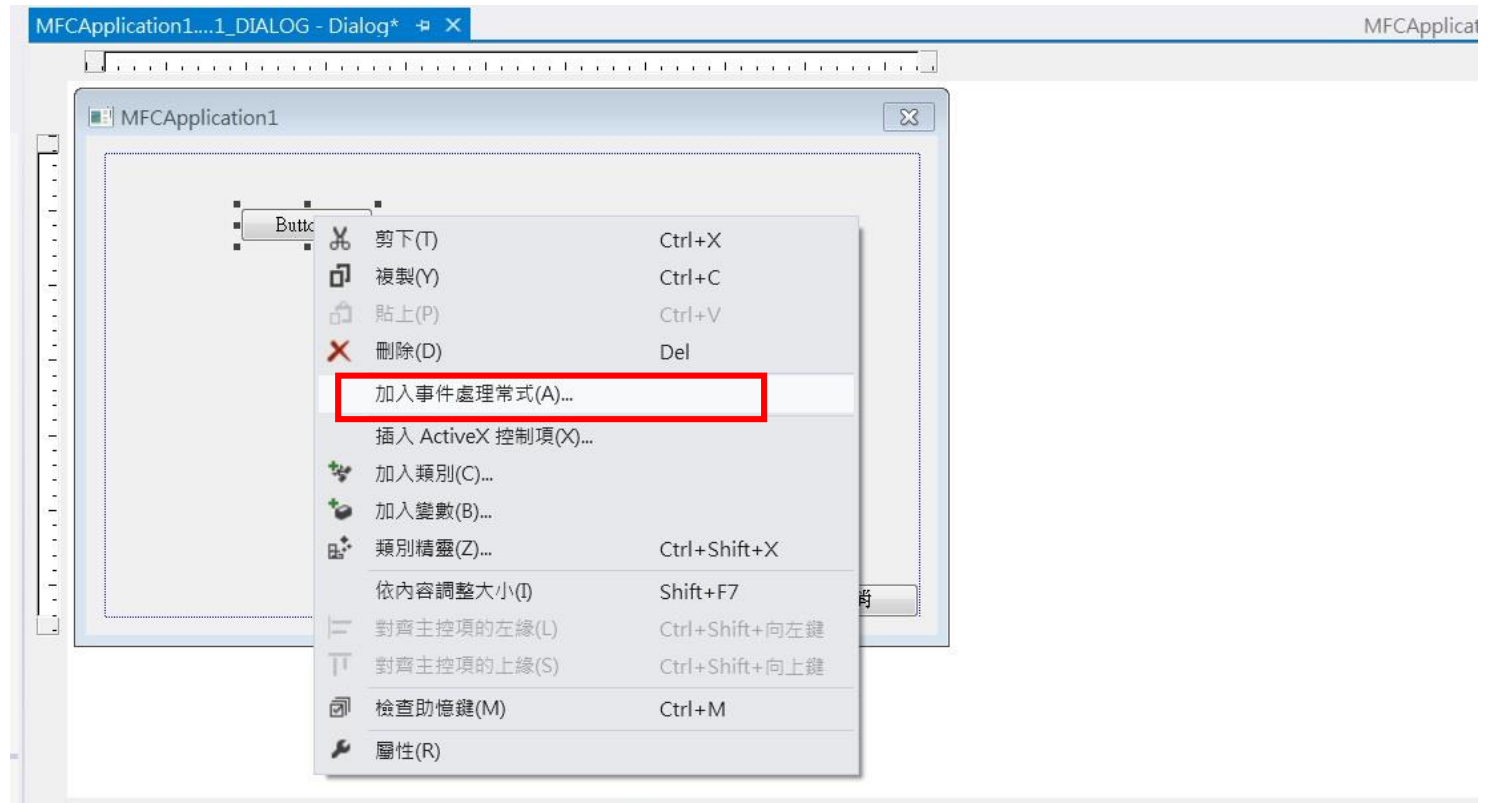
MFC：加入滑鼠點擊事件(BN_CLICKED)

- (法一)對元件滑鼠左鍵雙擊。
- (法二)於屬性的控制項事件對 BN_CLICKED 欄位新增函數，如右圖。



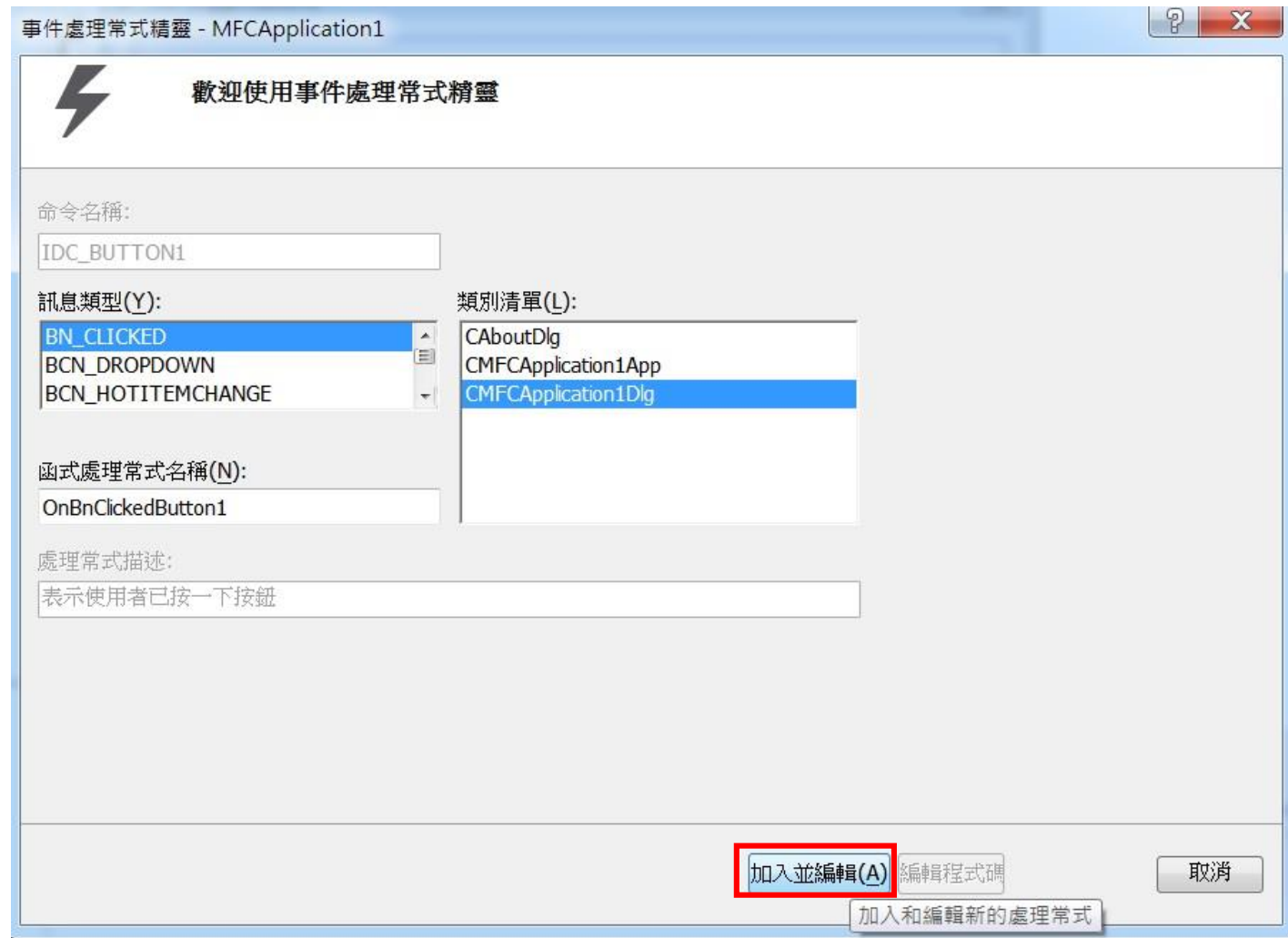
MFC：加入滑鼠點擊事件(BN_CLICKED)

- (法三)對元件點右鍵，
加入事件處理常式，
如右圖。



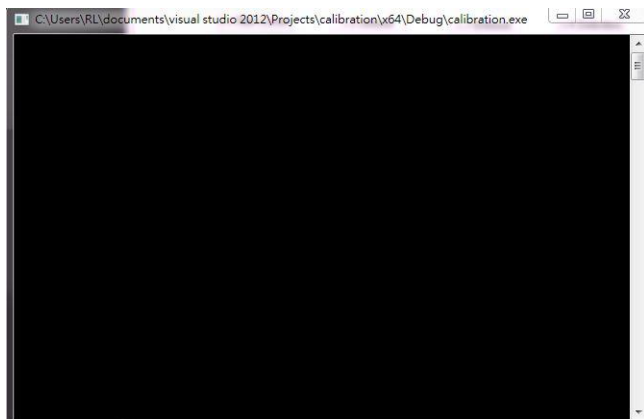
MFC：加入滑鼠點擊事件(BN_CLICKED)

- 預設為：
BN_CLICKED
C[專案名]Dlg
- 直接按下加入並編輯就行了。
- 其餘不同事件也是依照上述方法二或方法三來新增，有興趣者可以自行研究。



MFC：開DOS視窗

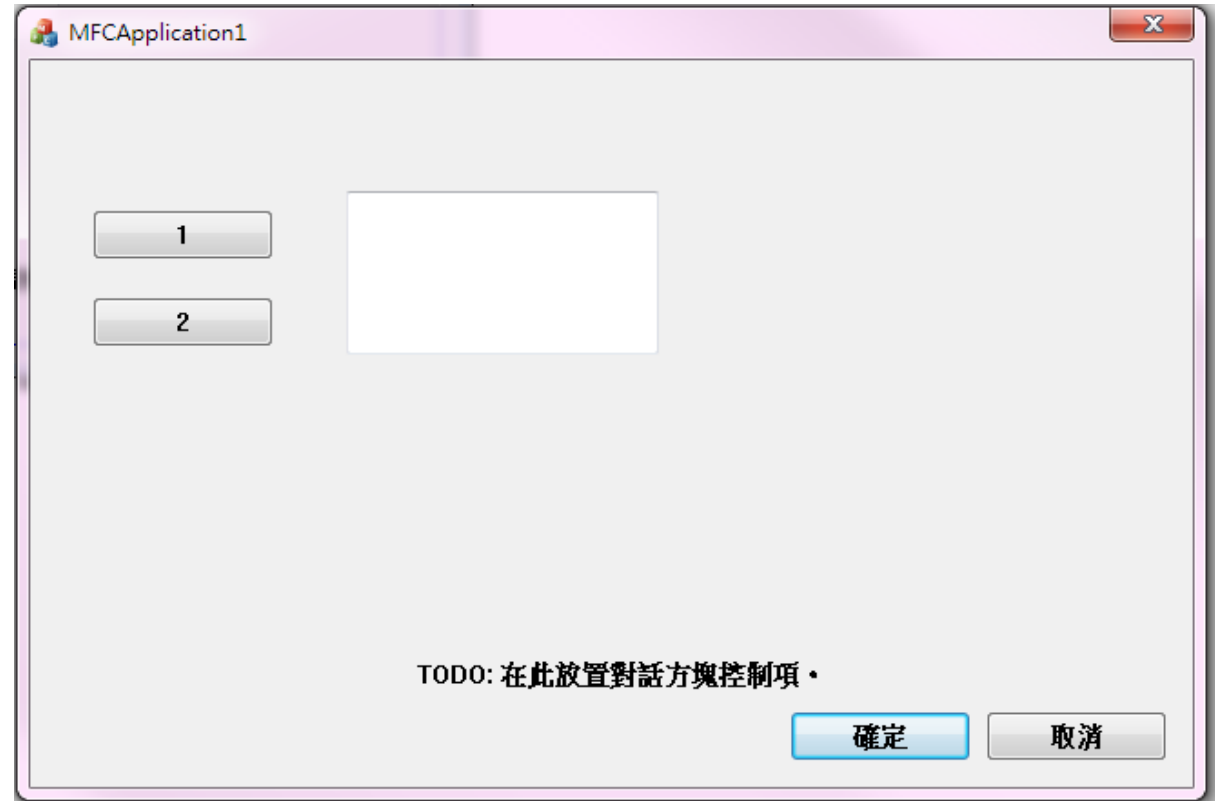
- 在BOOL CMFCApplication1Dlg::OnInitDialog()底下
- 加入AllocConsole();
- freopen ("CONOUT\$", "w", stdout);
- 就可以開黑窗出來了



```
100 {
101     pSysMenu->AppendMenu(MF_SEPARATOR);
102     pSysMenu->AppendMenu(MF_STRING, IDM_ABOUTBOX, strAbout);
103 }
104
105
106 // 設定此對話方塊的圖示。當應用程式的主視窗不是對話方塊時，
107 // 框架會自動從事此作業
108 SetIcon(m_hIcon, TRUE); // 設定大圖示
109 SetIcon(m_hIcon, FALSE); // 設定小圖示
110
111 // TODO: 在此加入額外的初始設定
112 AllocConsole();
113 freopen ("CONOUT$", "w", stdout );
114
115 return TRUE; // 傳回 TRUE，除非您對控制項設定焦點
116 }
117
118 void CMFCApplication1Dlg::OnSysCommand(UINT nID, LPARAM lParam)
119 {
120     if ((nID & 0xFFFF) == IDM_ABOUTBOX)
121     {
122         CAboutDlg dlgAbout;
123         dlgAbout.DoModal();
124     }
```

MFC：簡易範例

- 按下不同按鈕，更改 Edit Control 裡面的字。
- (步驟一)利用工具箱將 Button 以及 Edit Control 元件拖曳至 DIALOG畫面，介面如右：



MFC：簡易範例

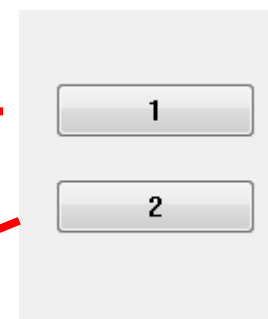
- 加入滑鼠點擊事件，並於兩個按鈕的滑鼠事件函數分別加入下方程式碼，如圖。

```
GetDlgItem(IDC_EDIT1)->SetWindowTextW(L"1");  
GetDlgItem(IDC_EDIT1)->SetWindowTextW(L"2");
```

此為Edit Control 元件的 ID，可於“屬性”查看

其他	
(Name)	IDC_EDIT1 (Edit Control)
Group	False
ID	IDC_EDIT1
Tabstop	True

```
void CMFCApplication1Dlg::OnBnClickedButton1()  
{  
    // TODO: 在此加入控制項告知處理常式程式碼  
    GetDlgItem(IDC_EDIT1)->SetWindowTextW(L"1");  
}  
  
void CMFCApplication1Dlg::OnBnClickedButton2()  
{  
    // TODO: 在此加入控制項告知處理常式程式碼  
    GetDlgItem(IDC_EDIT1)->SetWindowTextW(L"2");  
}
```

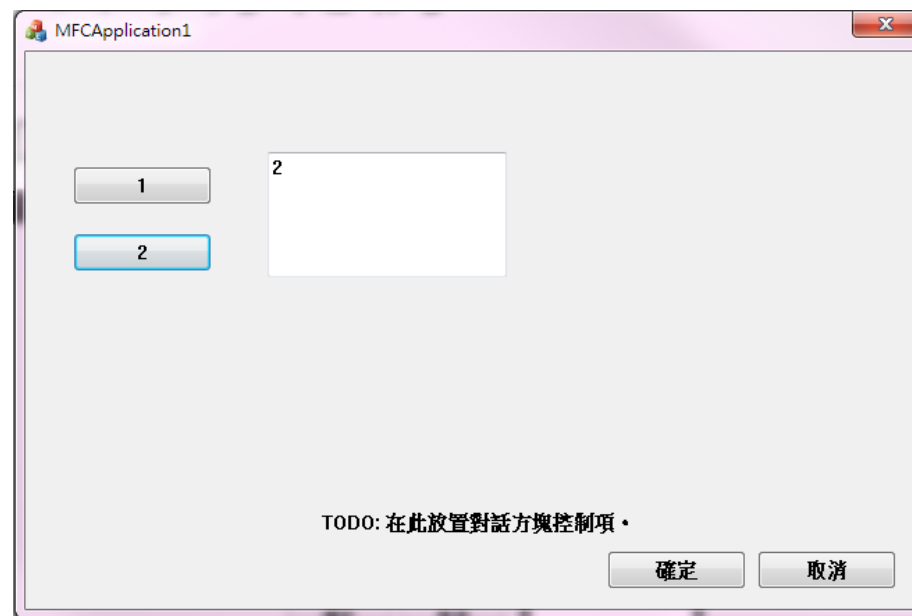


MFC：簡易範例

- 結果如圖：



(按下按鈕1)



(按下按鈕2)

Q&A

EX問題1：

- 範例 code 無法開啟 "highgui.h"



The screenshot shows the 'Output' window in Visual Studio. The 'Show output from (S):' dropdown is set to 'Build'. The output text is as follows:

```
1>----- 已開始建置：專案：MFCApplication1，組態：Debug x64 -----  
1> MFCApplication1Dlg.cpp  
1>MFCApplication1Dlg.cpp(9): fatal error C1083: 無法開啟包含檔案: 'highgui.h': No such file or directory  
===== 建置：0 成功、1 失敗、0 最新、0 略過 =====
```

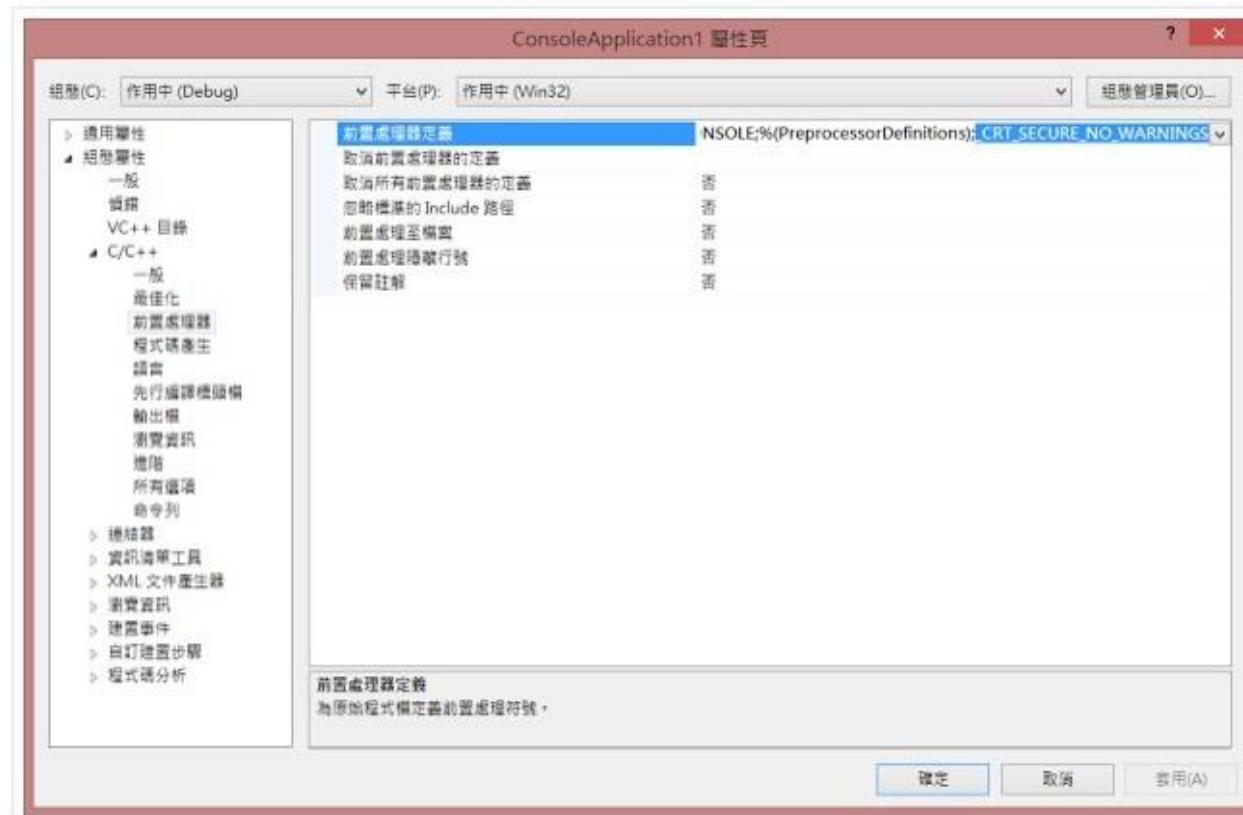
- 解法
 將 `#include "highgui.h"`
 改成 `#include "opencv2\highgui\highgui.hpp"`

EX問題2：

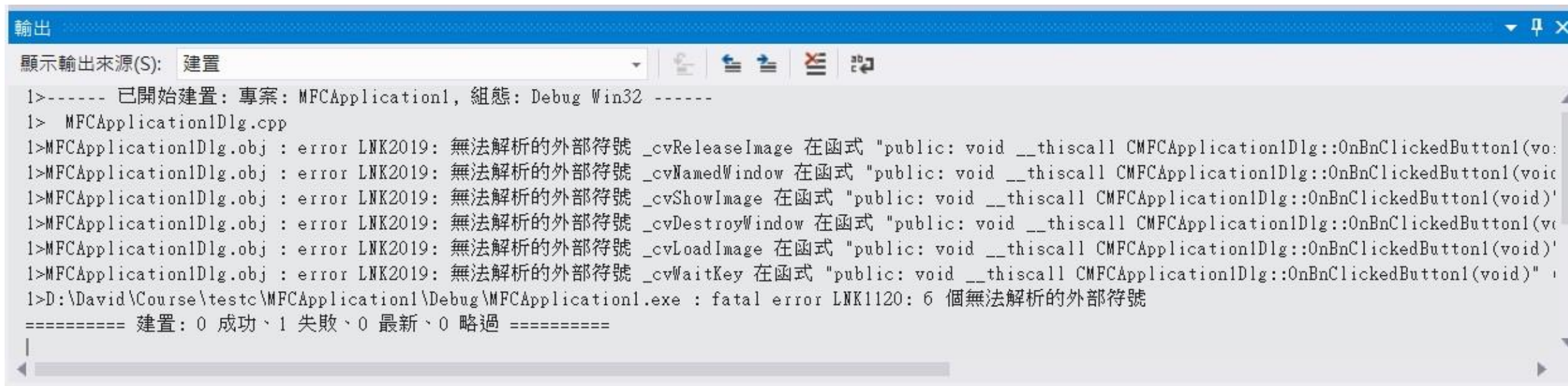
CODE

```
error C4996: 'scanf': This function or variable may be unsafe. Consider using scanf_s instead. To disable deprecation, use _CRT_SECURE_NO_WARNINGS. See online help for details.
```

- 解法：方案總管點右鍵→ 屬性
 - 組態屬性
 - C/C++ → 前置處理器
 - 於「前置處理器定義」後面加上 `_CRT_SECURE_NO_WARNINGS`
 - 確定



EX問題3：無法解析的外部符號



The screenshot shows a Visual Studio output window titled "輸出" (Output). The "顯示輸出來源(S):" (Show output from) dropdown is set to "建置" (Build). The output text is as follows:

```
1>----- 已開始建置: 專案: MFCAApplication1, 組態: Debug Win32 -----
1> MFCAApplication1Dlg.cpp
1>MFCAApplication1Dlg.obj : error LNK2019: 無法解析的外部符號 _cvReleaseImage 在函式 "public: void __thiscall CMFCAApplication1Dlg::OnBnClickedButton1(vo:
1>MFCAApplication1Dlg.obj : error LNK2019: 無法解析的外部符號 _cvNamedWindow 在函式 "public: void __thiscall CMFCAApplication1Dlg::OnBnClickedButton1(voic
1>MFCAApplication1Dlg.obj : error LNK2019: 無法解析的外部符號 _cvShowImage 在函式 "public: void __thiscall CMFCAApplication1Dlg::OnBnClickedButton1(void)'"
1>MFCAApplication1Dlg.obj : error LNK2019: 無法解析的外部符號 _cvDestroyWindow 在函式 "public: void __thiscall CMFCAApplication1Dlg::OnBnClickedButton1(vo
1>MFCAApplication1Dlg.obj : error LNK2019: 無法解析的外部符號 _cvLoadImage 在函式 "public: void __thiscall CMFCAApplication1Dlg::OnBnClickedButton1(void)'"
1>MFCAApplication1Dlg.obj : error LNK2019: 無法解析的外部符號 _cvWaitKey 在函式 "public: void __thiscall CMFCAApplication1Dlg::OnBnClickedButton1(void)" ,
1>D:\David\Course\testc\MFCAApplication1\Debug\MFCAApplication1.exe : fatal error LNK1120: 6 個無法解析的外部符號
===== 建置: 0 成功、1 失敗、0 最新、0 略過 =====
```

EX問題3：

- 解法：點開 **Win32**
→ **組態管理員**
→ **新增** → **x64** → **確定**

