

如何利用 MATLAB Compiler 轉出 C++ library 予 VS2005 使用

目的：

利用 MATLAB Compiler 將 MATLAB 的函式或 M-Files 轉成 C++ library(*.dll)，以供給 VS2005 使用。此範例包含 5 個步驟，分別說明如下：

1. 設定 MATLAB Compiler。
2. 輸入欲轉出的 M-Files，並決定所產生 library(*.lib)的名稱。
3. 撰寫 matrixdriver.cpp，呼叫 MATLAB 轉出的 C++ library。
4. 欲加入 VS2005 裡使用前，可以先利用 MATLAB Compiler 轉出 *.exe 測試看是否可正常執行。
5. 將產生的 library 置入 VS2005 裡使用。
6. 將產出的 C++ shared library 移置到沒有安裝 MATLAB 的環境。

軟體及系統需求：

1. Window 作業系統 (XP(含)以上的版本)
2. Matlab R2010a
3. MATLAB Compiler 4.13
4. Microsoft Visual Studio 2005 SP1 (Prof. Edition) 或是您可以使用其他 MATLAB 支援的 C Compiler，支援情形請參考下述網頁：

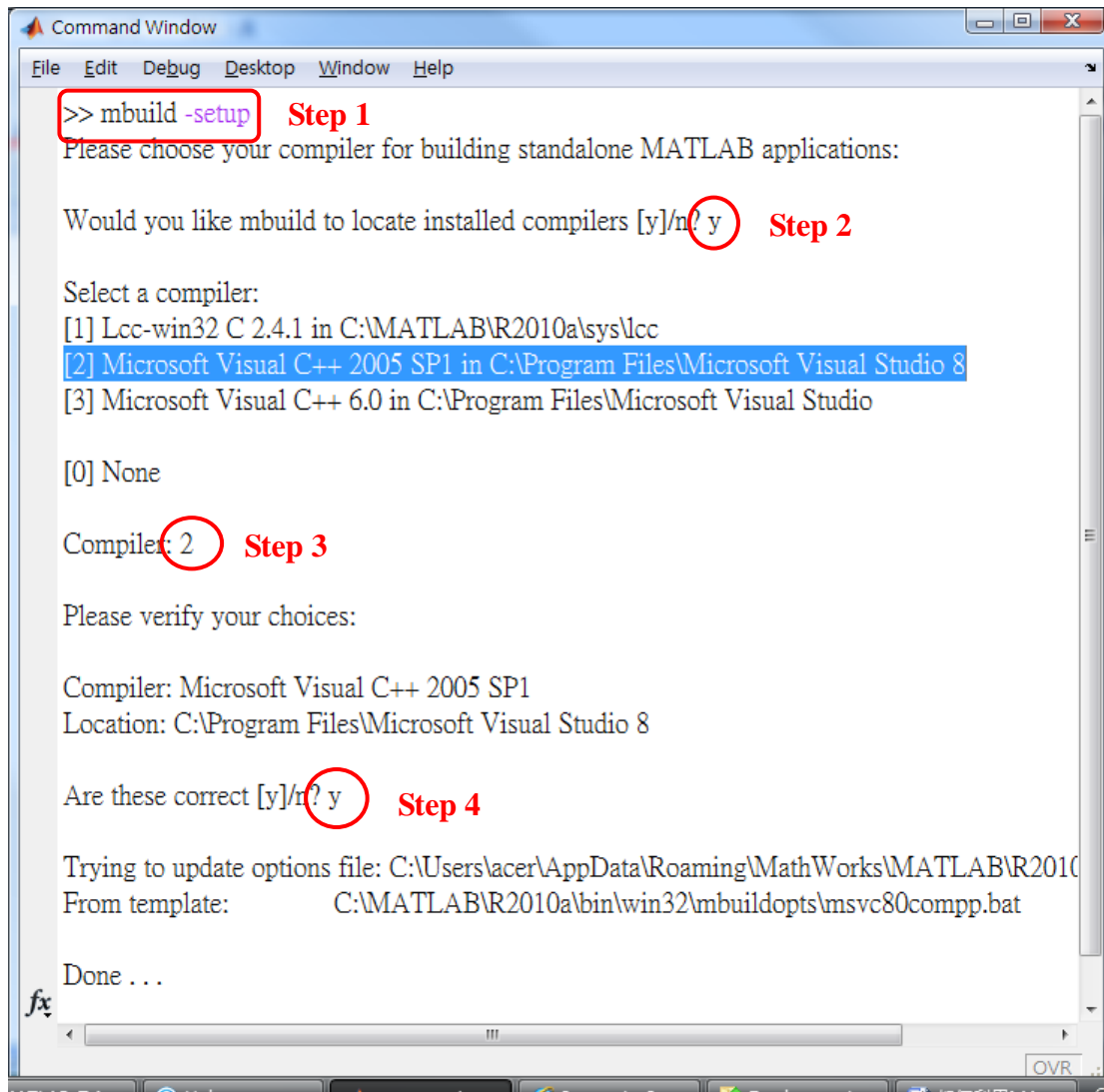
<http://www.mathworks.com/support/compilers/R2010a/index.html>

透過 MATLAB Compiler 將 M-file 轉成 C++ library(*.dll)以供給 VS2005 使用，步驟如下：

1. 設定 MATLAB Compiler

請在 MATLAB Command Window 下，執行下列指令，選擇所需要的 C Compiler，本範例請選擇 Microsoft Visual Studio 2005（依 Step1~4）。

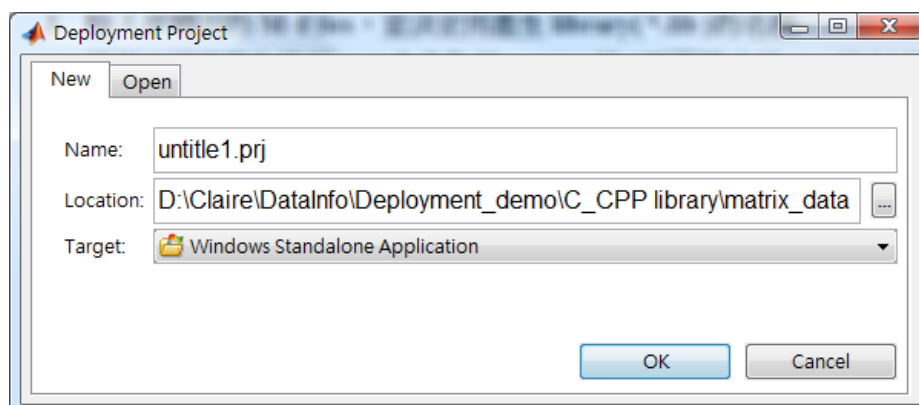
```
>> mbuild -setup
```



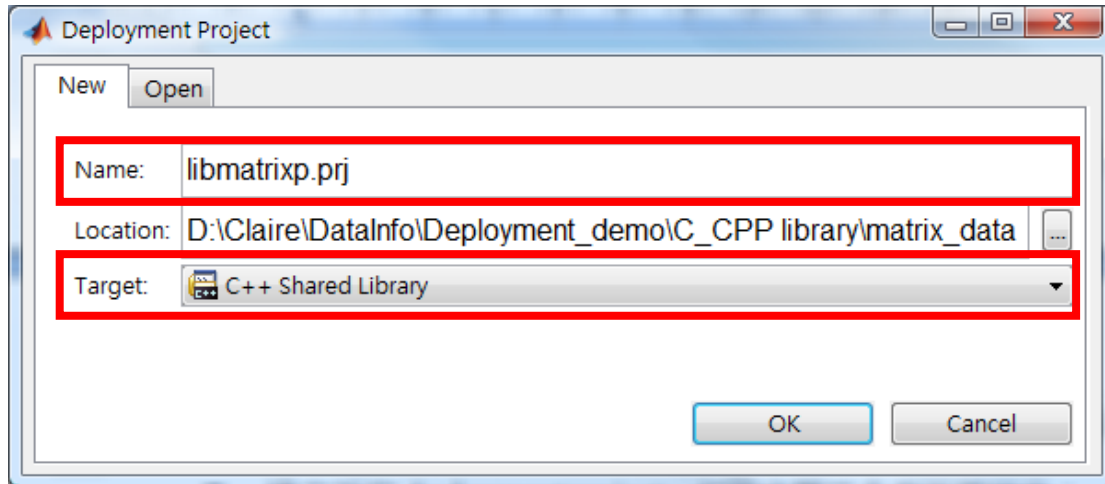
2. 輸入欲轉出的 M -Files，並決定所產生 library(*.lib)的名稱

範例中將欲產生的 library 命名為 libmatrixp.lib；而要轉成 library 的 M-Files 則為 addmatrix.m，multiplymatrix.m 與 eigmatrix.m。使用 Deployment Tool 的功能請參考下列說明：

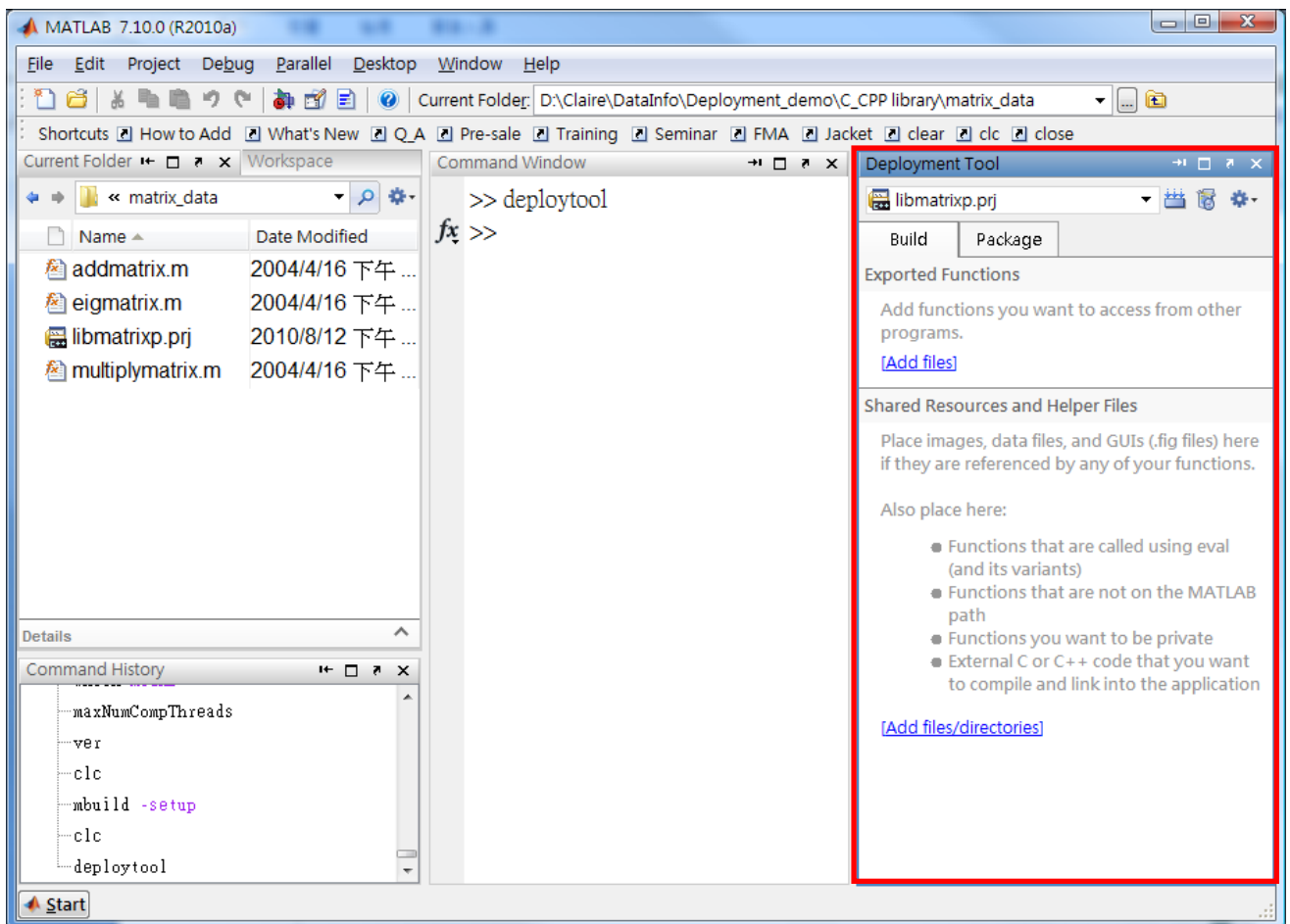
- 在 MATLAB Command Window 裡鍵入 deploytool 開啟 Deployment Tool 的介面。產生視窗如下圖所示：



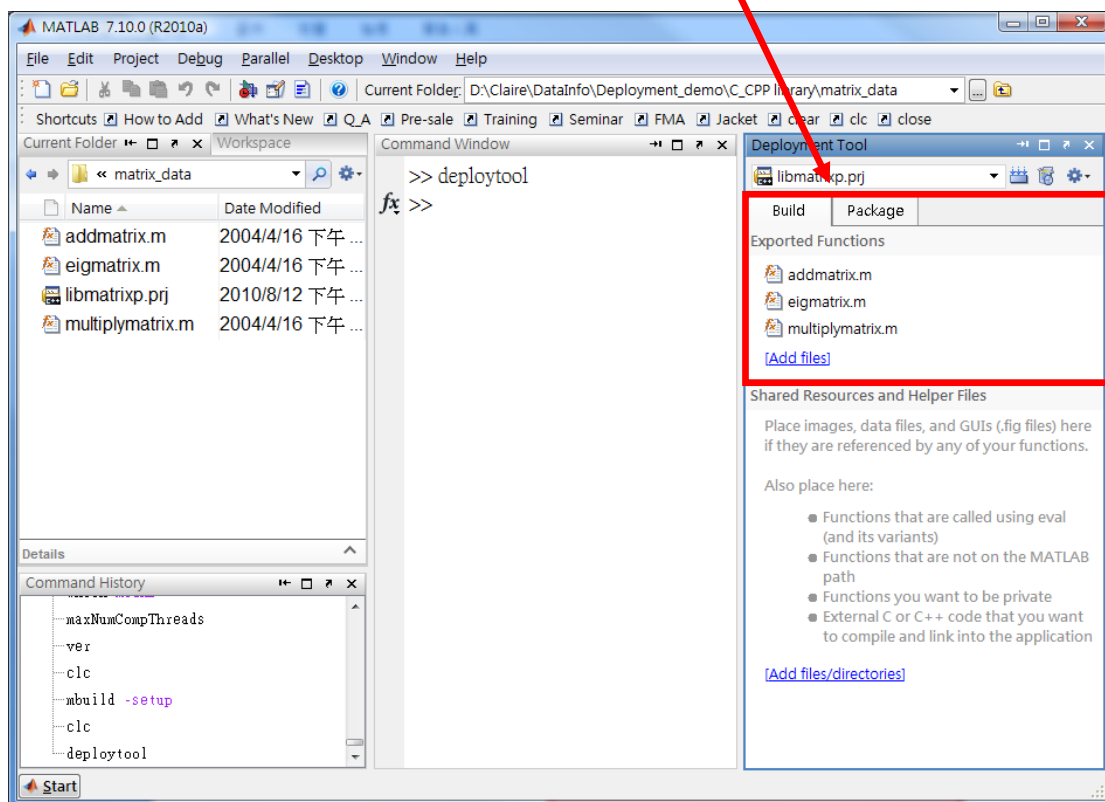
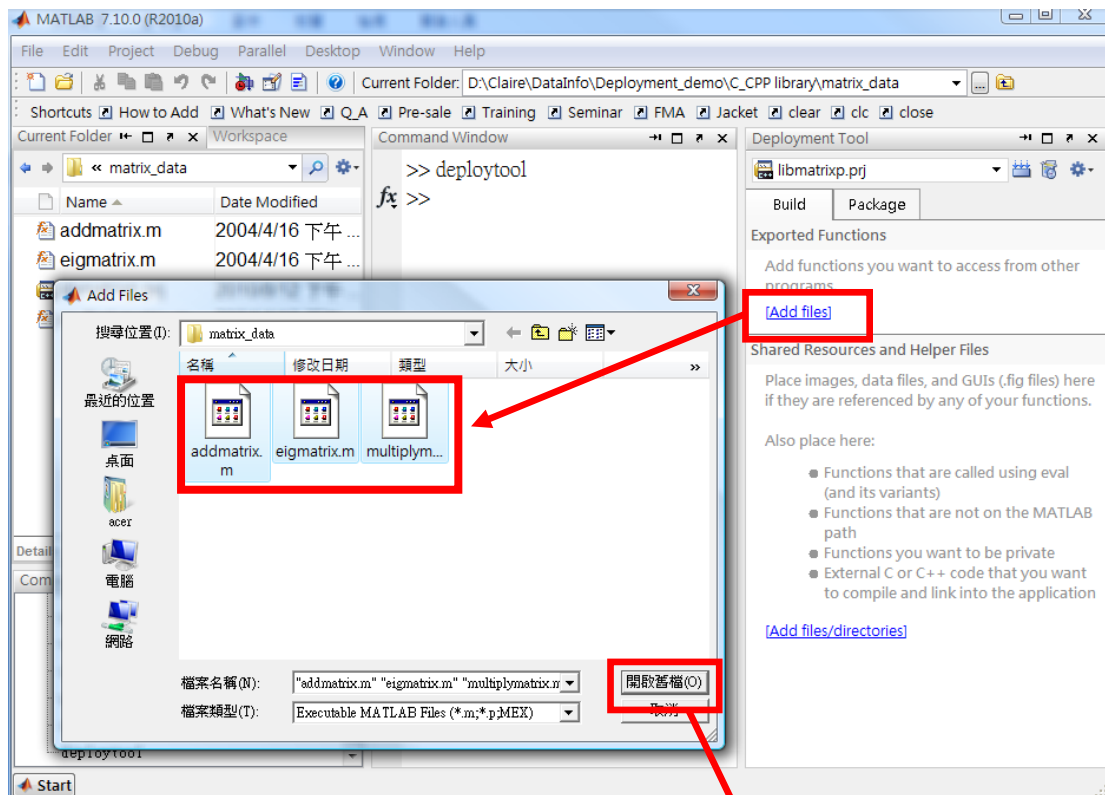
- 建立新的 deployment project，如下步驟並參考下圖所示：
 - i. Name：libmatrixp.prj (此為 library 的名稱)。
 - ii. Target：C++ Shared Library。
 - iii. Location：產生 C++ shared library 後的放置路徑。




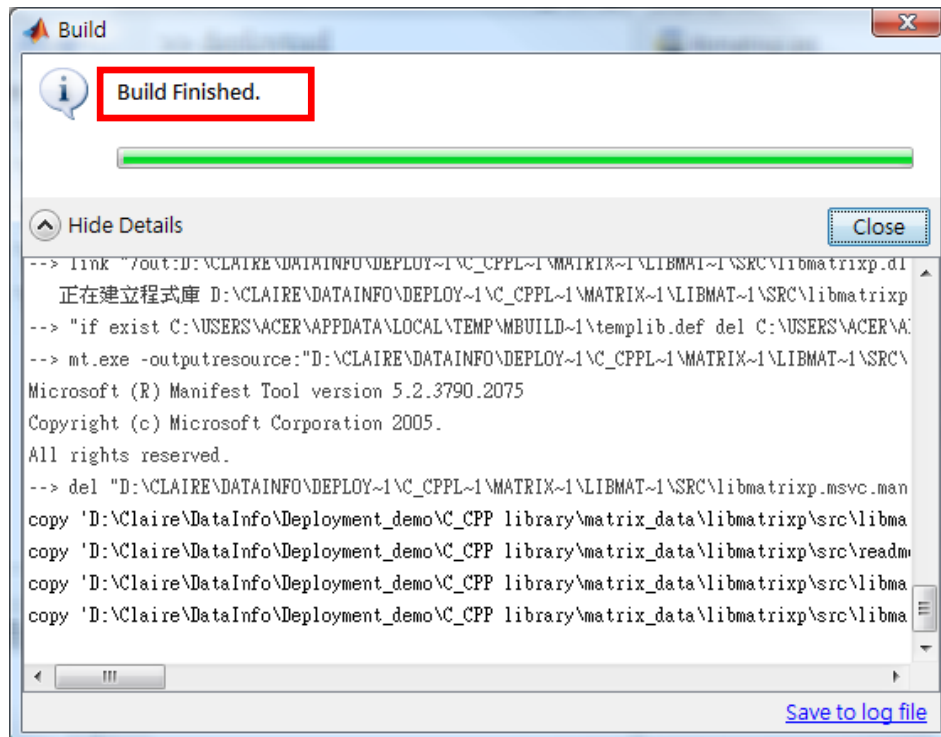
- iv. 按下 OK，產生新的 Project 如下圖所示。



- v. 將 M-file 加入此 Project，於 Build 處的 Exported functions，選擇 Add File，選取 addmatrix.m，multiplmatrix.m 與 eigmatrix.m 這 3 個檔案。



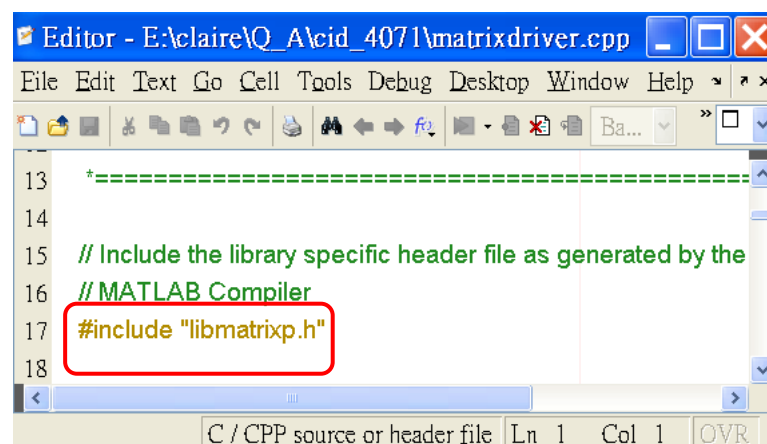
- vi. 按下 ，進行 C++ Shared Library 建置。建置完成後會在 Current Directory 中產生 libmatrixp(同 project name)的目錄，並於 Build 視窗顯示 Build Finished。而產生的 C++ Shared Library 會在 libmatrixp\distrib 目錄中。



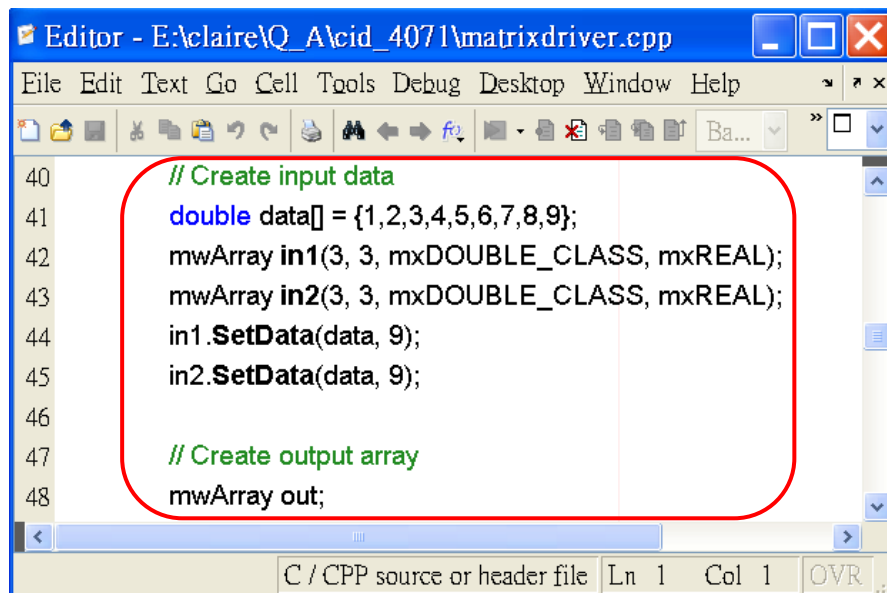
3. 撰寫 matrixdriver.cpp，產生之值傳入 libmatrixp.lib，計算結果則傳入 libmatrixp.lib

以下先簡單解釋此 C 架構：

- a. include 「libmatirxp.h」：libmatrixp.h 則是依據轉出的 Library 名稱不同而改變。



b. input 及 output 的宣告並給值：



```

40 // Create input data
41 double data[] = {1,2,3,4,5,6,7,8,9};
42 mxArray in1(3, 3, mxDOUBLE_CLASS, mxREAL);
43 mxArray in2(3, 3, mxDOUBLE_CLASS, mxREAL);
44 in1.SetData(data, 9);
45 in2.SetData(data, 9);
46
47 // Create output array
48 mxArray out;
  
```

c. 主程式架構如下，請參考 matrixdriver.cpp：

Creating Your MCR Instance → `if(!mclInitializeApplication(NULL,0))`

Create MATLAB Input Data → `mxArray in1(3, 3, mxDOUBLE_CLASS, mxREAL);`

Call The Library → `if(!libmatrixpInitialize())...`

Run the function1 of Libriary → `addmatrix(1, out, in1, in2);`

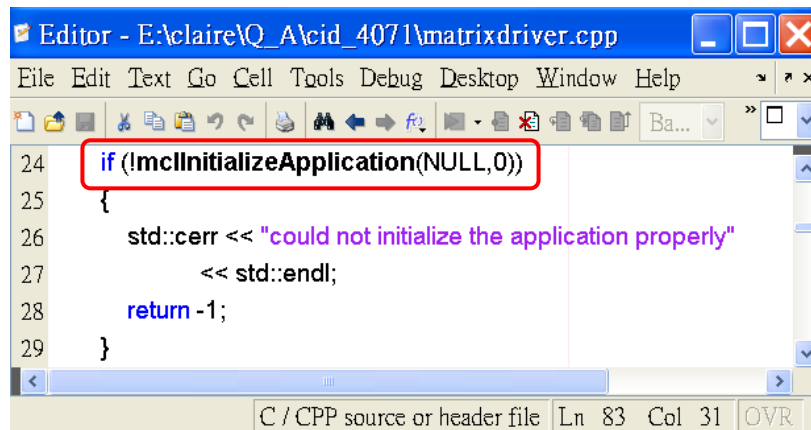
Display the Output1 → `std::cout << out << std::endl;`

Call the library termination routine → `libmatrixpTerminate();`

Turn off the MCR instance → `mclTerminateApplication();`

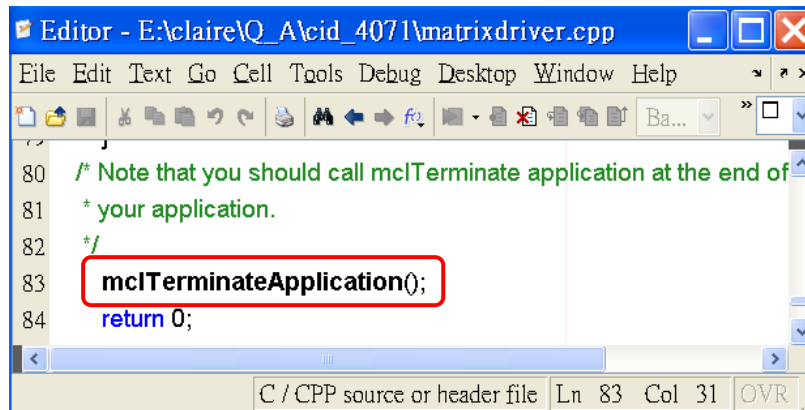
.....

粗體字部分 → 固定寫法，以後如果要在撰寫新 C++ language 去呼叫 MATLAB 轉出的 Library，都是這樣寫。



```

24 if (!mclInitializeApplication(NULL,0))
25 {
26     std::cerr << "could not initialize the application properly"
27         << std::endl;
28     return -1;
29 }
  
```



其它的 → 會依使用的狀況而改變，詳細說明如下：

i. Create MATLAB Input Data

`mwArray in1(3, 3, mxDOUBLE_CLASS, mxREAL);` → MATLAB `mw` 專用指令，產生一個 3*3 的矩陣空間，此矩陣空間要依據資料指定，本範例的資料為 `data`，參考步驟 b。

`in1.SetData(data, 9);` → 將 `data` 設定給 `in1` 的位置上，其中 9 是上面 3*3 的結果。

ii. Call The Library

`if (!libmatrxpInitialize()){ }` → 藍色的部分即是 User 命名的 Library name，要注意大小寫。Initialize 固定寫法。

iii. Run the function1 of Library

`addmatrix(1, out, in1, in2);` → 藍色部分為 M-file 名稱；1 代表 Output 的數量；`out` 代表 output 變數名稱；接著在寫 input 名稱下去。

iv. Call the library termination routine

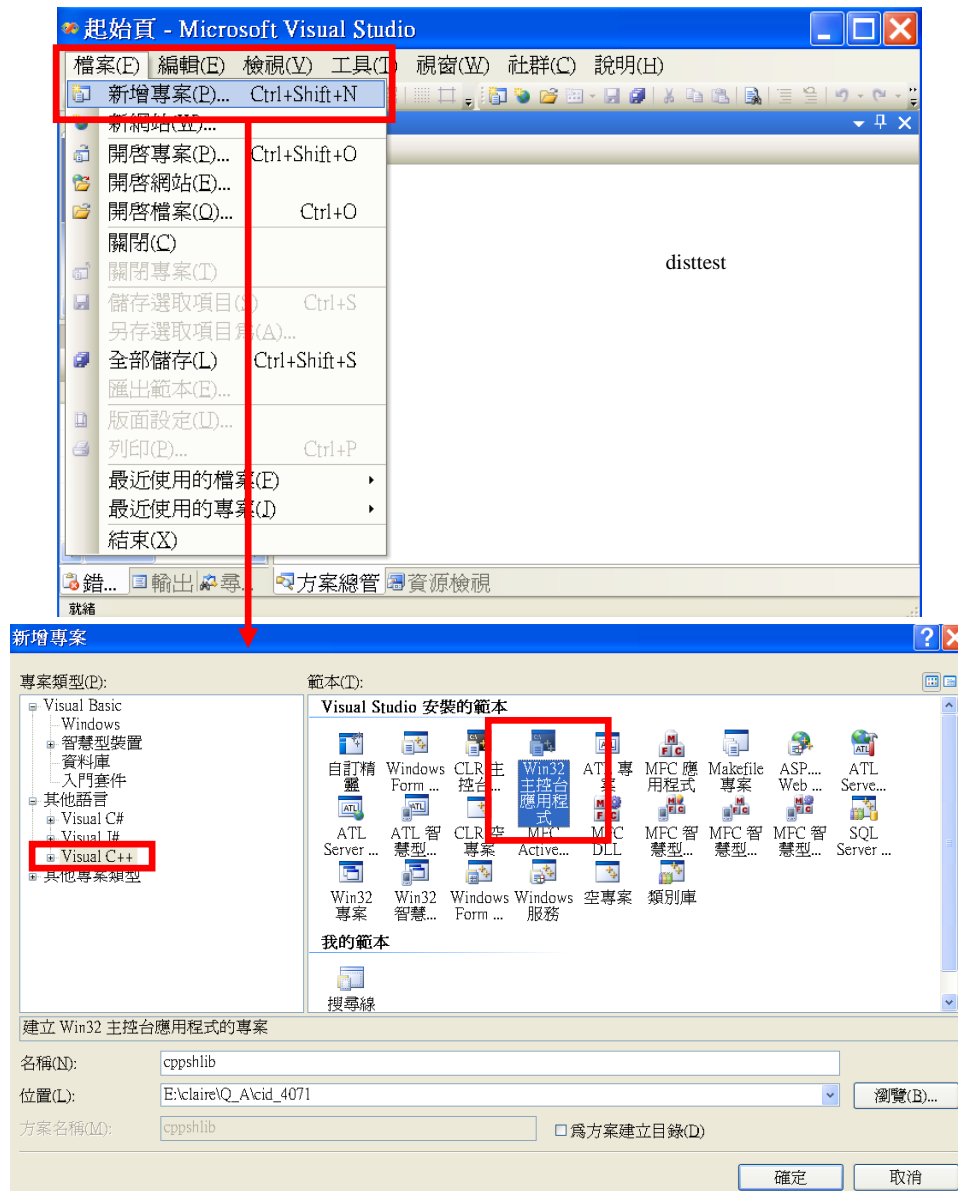
`libmatrxpTerminate();` → 藍色的部分即是 User 命名的 Library name，要注意大小寫。Terminate 固定寫法。

4. 欲加入 Microsoft Visual Studio 2005 裡使用前，可以先利用 MATLAB Compiler 轉出 *.exe 測試看是否可正常執行。請將 `matrixdriver.cpp` 與 `libmatrxp.lib` 放置在同一路徑下，於 MATLAB Command Window 底下鍵入指令，之後會產生 `matrixdriver.exe` 檔案。

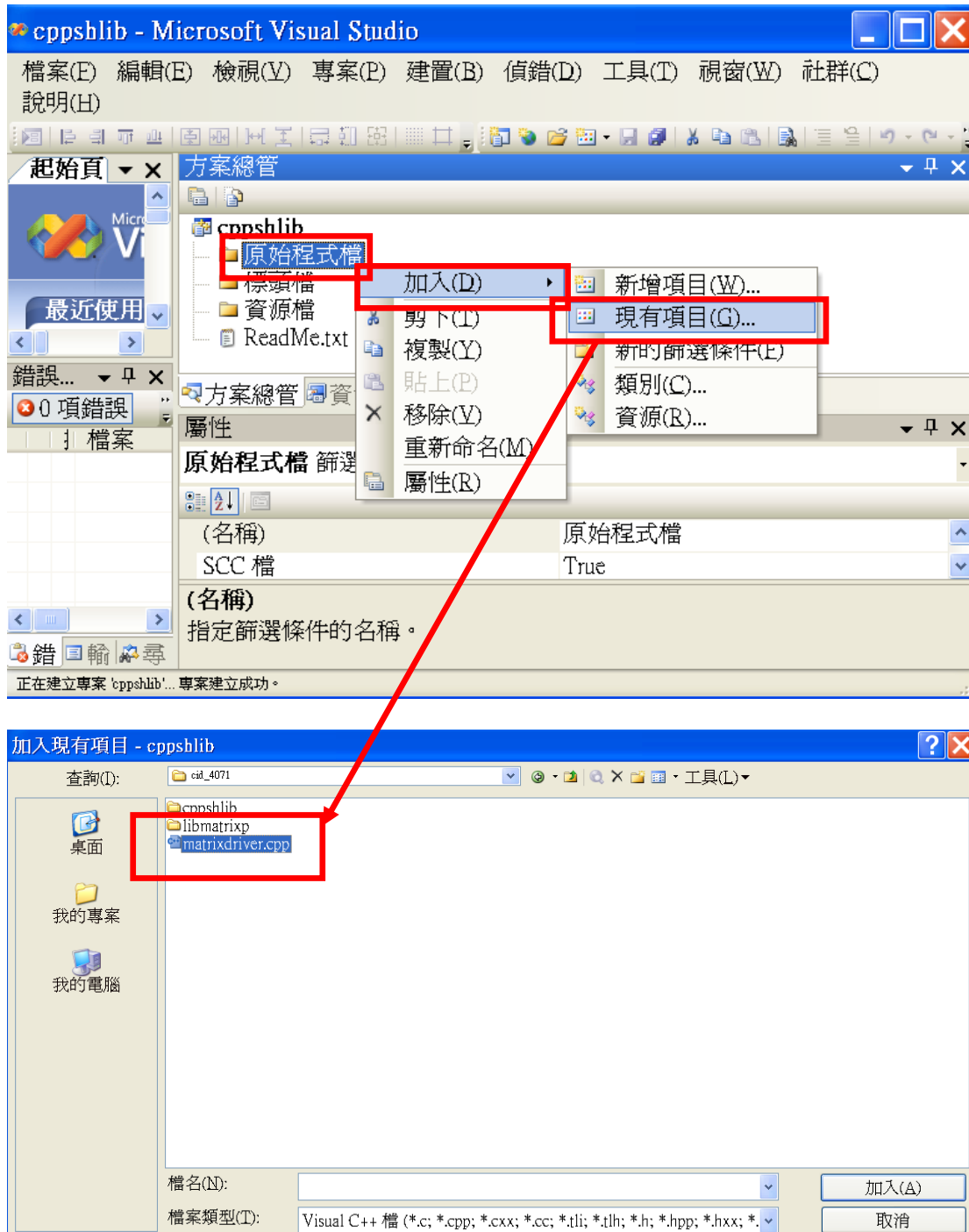
```
>>mbuild matrixdriver.cpp libmatrxp.lib
```

5. 將產生的 library 置入 Microsoft Visual Studio 2005 裡使用

- a. 打開 Microsoft Visual Studio 2005，選檔案 → 新增專案 → win32 主控台應用程式，輸入 Project name(範例命名為 `cppshlib`)，按下 OK 按鈕。

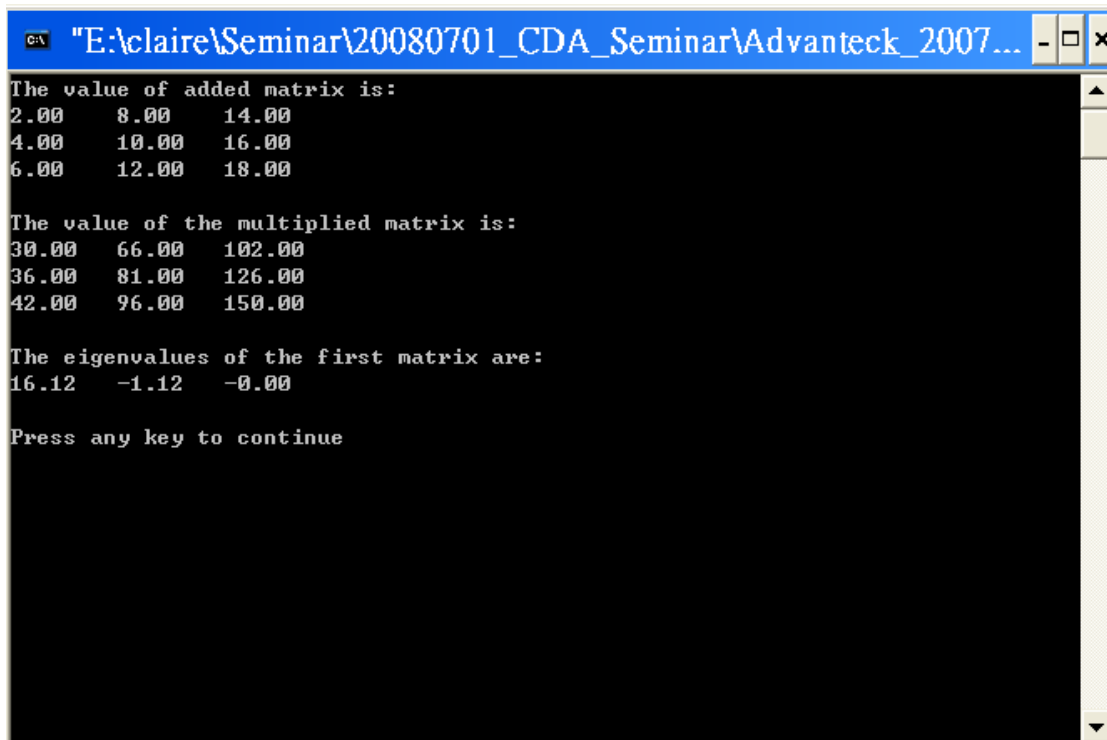


- b. 開啟方案總管
- i. 將 matrixdriver.cpp 加入專案中。



- ii. 標頭檔: 加入 libmatrixp.h, 此檔案位於 libmatrixp\distrib 目錄中。
- iii. 資源檔: 加入 libmatrixp.lib (此檔案位於 libmatrixp\distrib 目錄中) 以及 mclmcrtr.lib (此檔案位於 <MATLABROOT>\extern\lib\win32\microsoft\, <MATLABROOT>代表您安裝 MATLAB 的路徑)。

- c. 點選專案 → cppshlib 屬性 → 組態屬性 → 加入路徑
 - i. C/C++ → 一般 → 其他 Include 目錄，加入下列路徑：
 - <MATLABROOT>\extern\include
 - MATLAB 轉出 C++ library 的目錄，如：
E:\claire\Q_A\cid_4071\libmatrixp\distrib
 - ii. 連結器 → 一般 → 其他程式庫目錄，加入下列路徑：
 - <MATLABROOT>\extern\lib\win32\microsoft
 - MATLAB 轉出 C++ library 的目錄，如：
E:\claire\Q_A\cid_4071\libmatrixp\distrib
- d. 存檔 → 編譯專案 → 建置專案。
- e. 執行之前，必須將 MATLAB 產生的 libmatrixp.dll 複製到 bebug 資料夾中，與 exe 執行檔放在一起才能順利執行。執行後的結果如下：



```

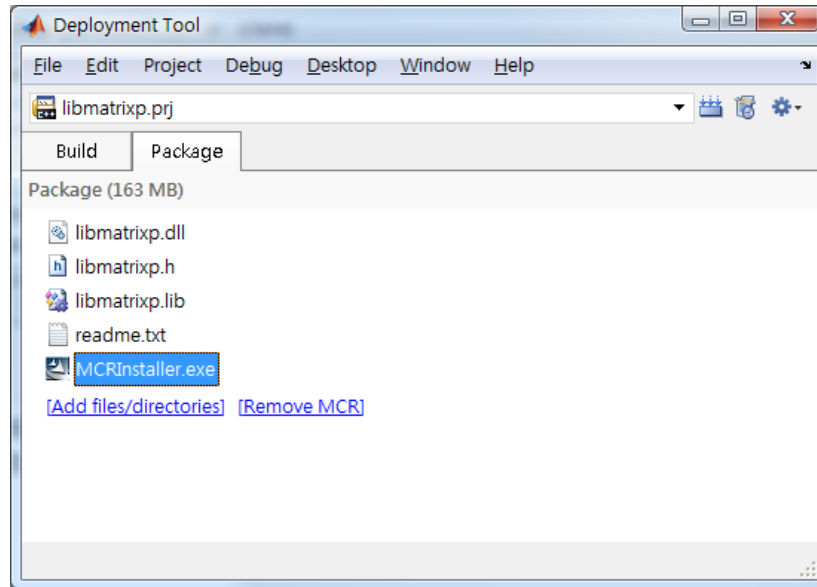
C:\ "E:\claire\Seminar\20080701_CDA_Seminar\Advantech_2007...
The value of added matrix is:
2.00  8.00  14.00
4.00  10.00  16.00
6.00  12.00  18.00

The value of the multiplied matrix is:
30.00  66.00  102.00
36.00  81.00  126.00
42.00  96.00  150.00

The eigenvalues of the first matrix are:
16.12  -1.12  -0.00


Press any key to continue
  
```

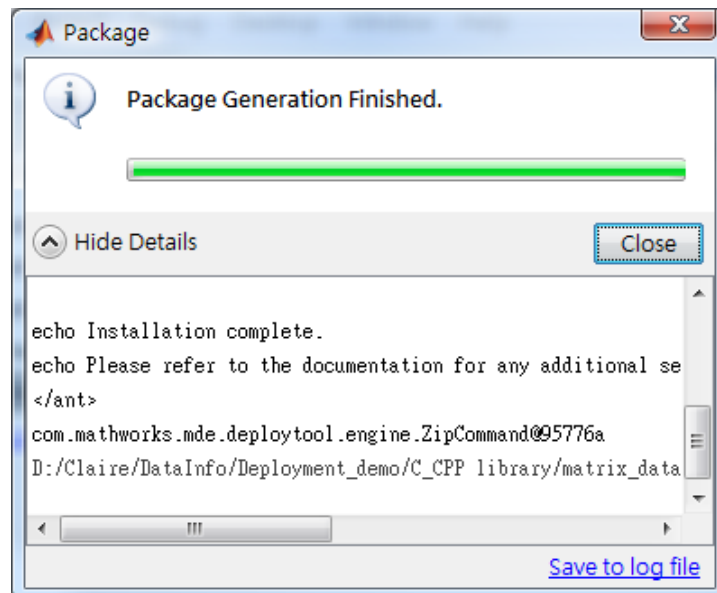
6. 將產出的 C++ shared library 移置到沒有安裝 MATLAB 的電腦中：
 - a. 在 Deploy Tool 視窗中，選擇 Package，將要發佈的檔案，透過[Add files/directories]或是[Add MCR]整理在一起，如下圖所示：



NOTE：[Add files/directories]的功能可將說明文件或是其他附件與 C++ shared library 被打包在一起。

[Add MCR]的功能會將 MCRInstaller.exe 與 C++ shared library 被打包在一起，請將 **MCRInstaller.exe** 安裝於沒有安裝 **MATLAB** 的電腦上，僅需安裝一次即可。

- b. 點選  按鈕作打包的動作，選擇檔案儲存的路徑，完成後會出現 Package 視窗，並顯示打包完成，如下圖所示。



- c. 將打包完成的檔案，移置到其他電腦，安裝完 MCRInstaller.exe 後，按照第 5 步驟，即可在 VS2005 中使用 C++ shared library。