

Classification	ISaGRAF Chinese FAQ-167						
Author	Chun Tsai	Version	1.3	Date	Feb. 2014	Page	1 / 31

## 在 ISaGRAF WinCE PAC 內開發屬於自己的 C-function 與 C-function Block?

### ● 應用說明

從以下的 ISaGRAF Driver (驅動) 版本起開始支持使用者自行開發、屬於自己專屬的 C-function 與 C-function block. 開發完後, 就可在 PC/ISaGRAF 上使用.

VP-2xW7/2xW6, VP-4137 :	1.53 版起
WP-5147/WP-5146 :	1.07 版起
WinPAC-8xx7 :	1.61 版起
XP-8xx7-ATOM-CE6 :	1.02 版起
XP-8xx7-CE6 :	1.41 版起

若你的 PAC 內的 ISaGRAF 驅動版本是比上方還早期的版本, 可參訪網站來取得最新的版本:  
<http://www.icpdas.com/en/download/show.php?num=368&nation=US&kind1=&model=&kw=isagraf>

### ● 檔案說明

本文件 “FAQ167\_chinese.pdf” (與 “FAQ167\_demo\_chinese.zip”) 可以從以下網址取得

<https://www.icpdas.com/en/faq/index.php?kind=280#751> > FAQ-167. 其中

“FAQ167\_demo\_chinese.zip” 包含了 “FAQ167\_chinese.pdf” 與幾個 EVC++ 4.0 和 VS2008 專案, 可用來編譯自定的 C-function 與 C-function block 的 DLL 檔案.

ICP DAS 標準出貨的 ISaGRAF PAC 沒有包含以下任何 “User\_c” 的 DLL. 若想在以下 PAC 內新增自定的 C-function 與 C-function block, 使用者就必需加入自己編譯產生的 DLL 檔案, 並和 ISaGRAF Driver (驅動) 放在 PAC 內的同一個路徑內. 然後將該 PAC 重新開機一次, 它就會支持使用者自定的 C-function 與 C-function block.

PAC 內要支持自定 C-function 與 C-function block 必須具備的 DLL 檔案:

XP-8xx7-CE6	\\System_disk\\ISaGRAF\\User_c_xpce6.dll
WP-8xx7	\\System_disk\\ISaGRAF\\User_c_wp8.dll
WP-5147	\\Micro_SD\\ISaGRAF\\User_c_wp5.dll
VP-25W7 / VP-4137	\\System_disk\\ISaGRAF\\User_c_vp.dll
XP-8xx7-ATOM-CE6	\\System_disk\\ISaGRAF\\User_c_xpce6_atom.dll

Classification	ISaGRAF Chinese FAQ-167						
Author	Chun Tsai	Version	1.3	Date	Feb. 2014	Page	2 / 31

## 1. 如何開發自定義的 C-function 與 C-function block

此 FAQ 提供了自定 C-function 與 C-function block 時, 所需的 DLL 檔案. 請先將檔案放在 PAC 中與 ISaGRAF Driver 相同的路徑內 (參考上一頁), 或使用以下專案來編譯產生 DLL 檔案.

例如: 本文件是以 XP-8347-CE6 來當範例操作. 需把 "User\_c\_xpce6.dll" 複製到 \System\_disk\ISaGRAF\ 的路徑下, 並將 PAC 重新開機.

### 1.1. 選擇欲開發的目標專案

在 FAQ167\_demo\_chinese.zip 中提供了以下專案, 可用來編譯 (參考 4.3 節- 步驟 2) 並產生 DLL 檔案.

例如: 本文件是針對 XP-8xx7-CE6 PAC 來說明 EVC++4.0 專案 (參考第 4 章). 使用者需把 "User\_c\_xpce6" 這個 EVC++ 4.0 專案資料夾複製到 PC 內. 請確定您的 PC 有安裝 EVC++ 4.0 軟體才可以編譯資料夾內的專案.

#### ◆ EVC++ 4.0 專案

1. User\_c\_vp : VP-2xW7/2xW6, VP-4137
2. User\_c\_wp5: WP-5147/WP-5146
3. User\_c\_wp8: WinPAC-8xx7
4. [User\\_c\\_xpce6 : XP-8xx7-CE6](#)
5. User\_c\_xpce6\_atom : XP-8xx7-ATOM-CE6

#### ◆ VS2008 專案

1. User\_c\_vp : VP-2xW7/2xW6, VP-4137
2. User\_c\_wp5: WP-5147/WP-5146
3. User\_c\_wp8: WinPAC-8xx7
4. [User\\_c\\_xpce6 : XP-8xx7-CE6](#)
5. User\_c\_xpce6\_atom : XP-8xx7-ATOM-CE6

另外, 此 FAQ-167 也提供了已完成的 ISaGRAF 範例程式.

(C-function 範例 - "by\_long"; C-function block 範例 - "long\_by"; ISaGRAF 專案 - "user\_c.pia").

您可在 1.2 節 先回存這些檔案, 並對照著第三章的說明來逐步開發屬於自己的 C-function 與 C-function Block.

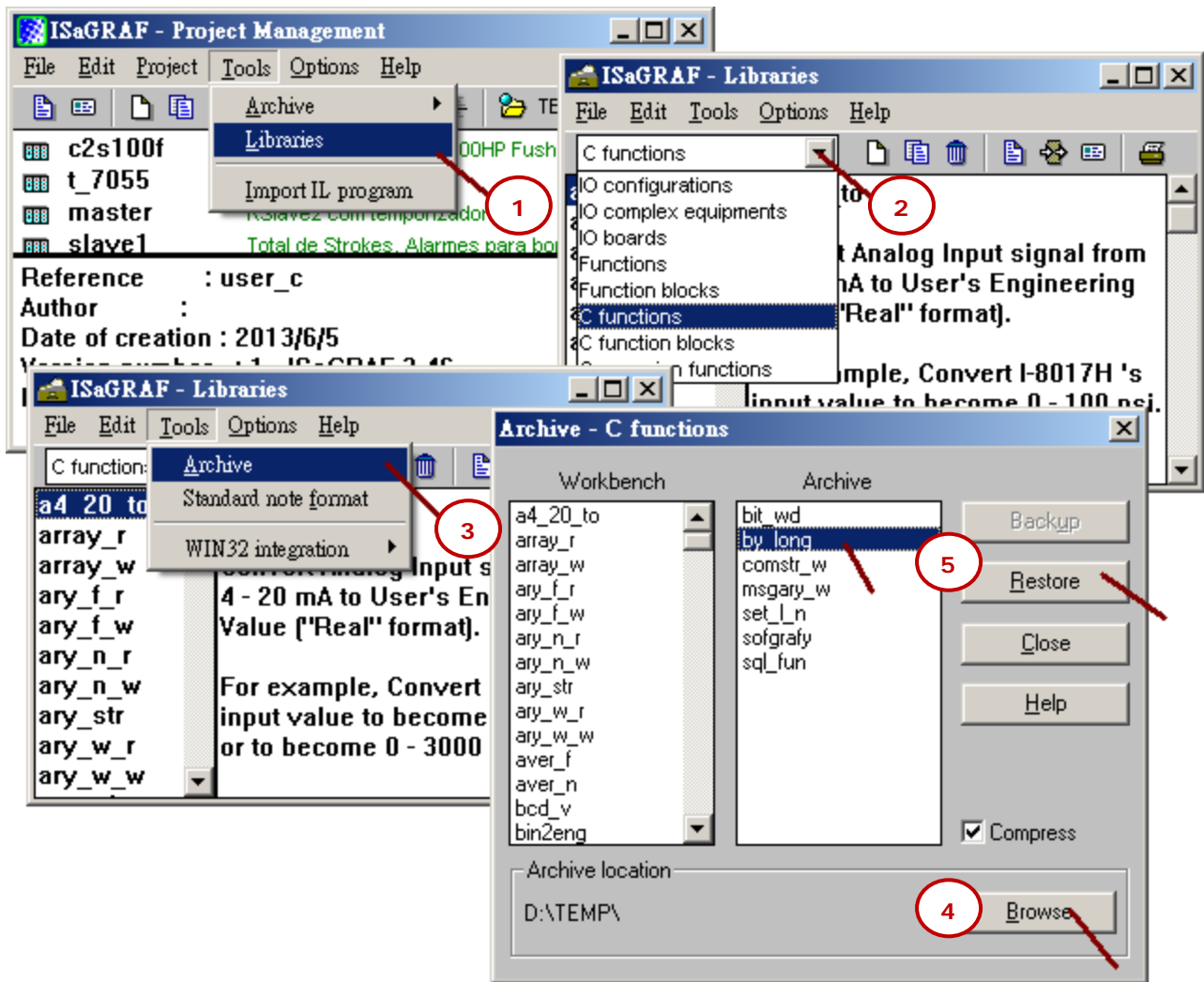
Classification	ISaGRAF Chinese FAQ-167						
Author	Chun Tsai	Version	1.3	Date	Feb. 2014	Page	3 / 31

## 1.2. 回存 ISaGRAF 範例程式

本文件的第三章會針對 XP-8xx7-CE6 PAC 來逐步說明 EVC++4.0 專案. 您可先回存已經完成的 ISaGRAF 範例程式到你的 PC / ISaGRAF (ISaGRAF 3.46 或 3.55 版) 來進行操作演練.

### A. 回存 C-function 範例 - "by\_long"

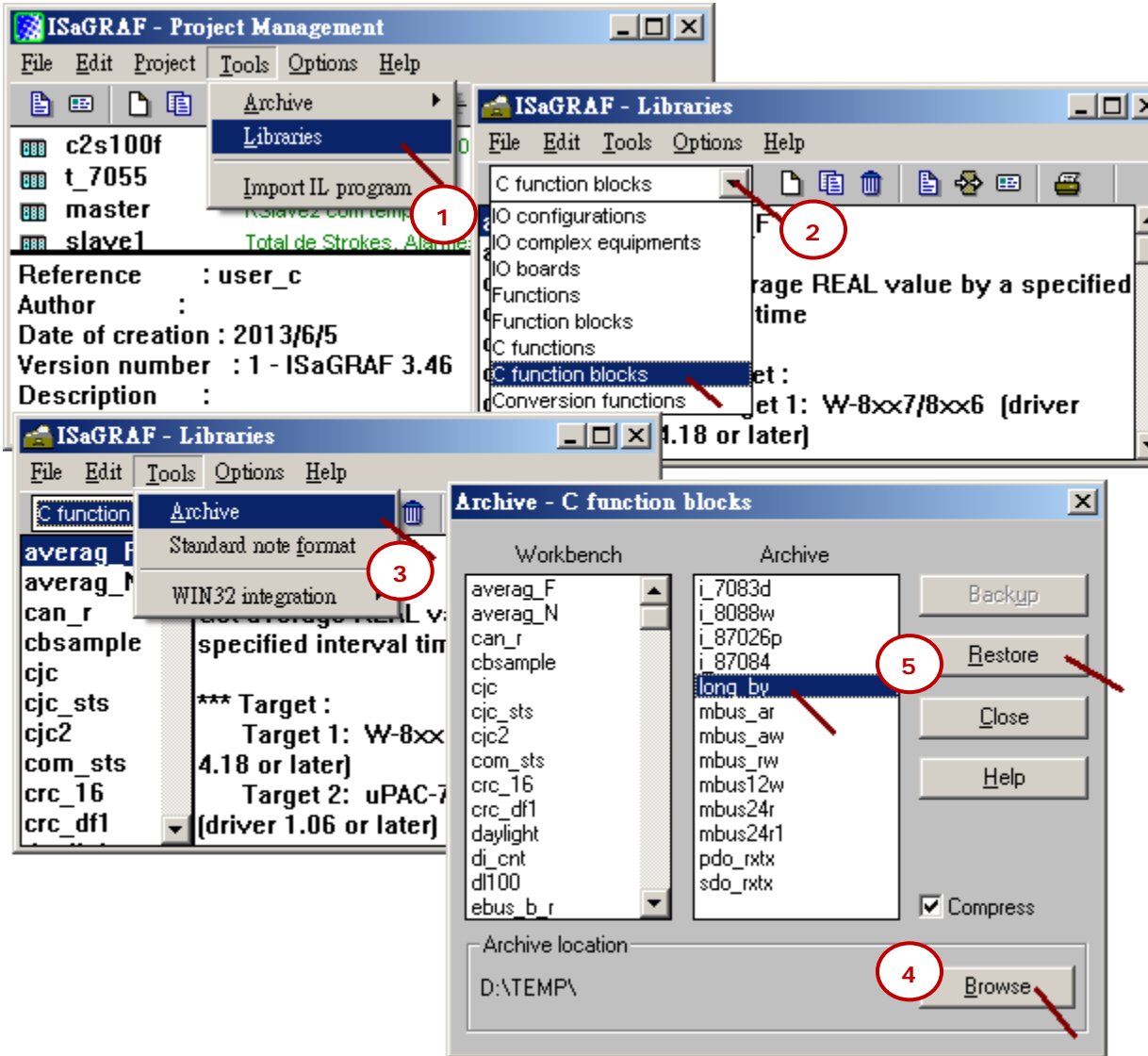
把 4 個 byte (值 0 ~ 255) 轉換組成一個 長整數 (32-bit signed integer).



Classification	ISaGRAF Chinese FAQ-167						
Author	Chun Tsai	Version	1.3	Date	Feb. 2014	Page	4 / 31

**B. 回存 C-function block 範例 - “long\_by”**

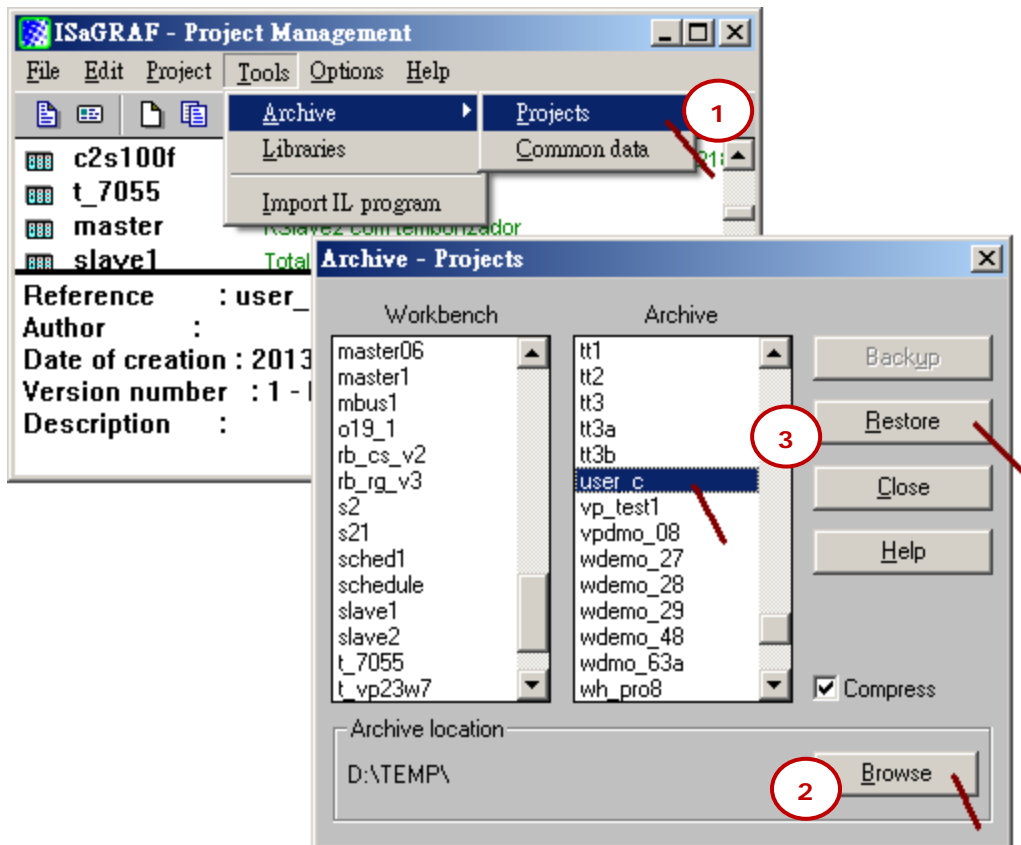
用來將一個長整數 轉換分解成 4 個 byte.



Classification	ISaGRAF Chinese FAQ-167					
Author	Chun Tsai	Version	1.3	Date	Feb. 2014	Page 5 / 31

### C. 回存 ISaGRAF 專案 - “user\_c.pia”

有使用到自定義的 C-function - “by\_long” 與 C-function block - “long\_by” 的應用程式範例 “user\_c.pia”。



回存完成後，請將 ISaGRAF 專案 “user\_c.pia” 重新編譯並下載至控制器中，即可開始使用自定義的 C-function “by\_long” 與 C-function block “long\_by”。

Classification	ISaGRAF Chinese FAQ-167						
Author	Chun Tsai	Version	1.3	Date	Feb. 2014	Page	6 / 31

## 2. 建立開發編譯環境

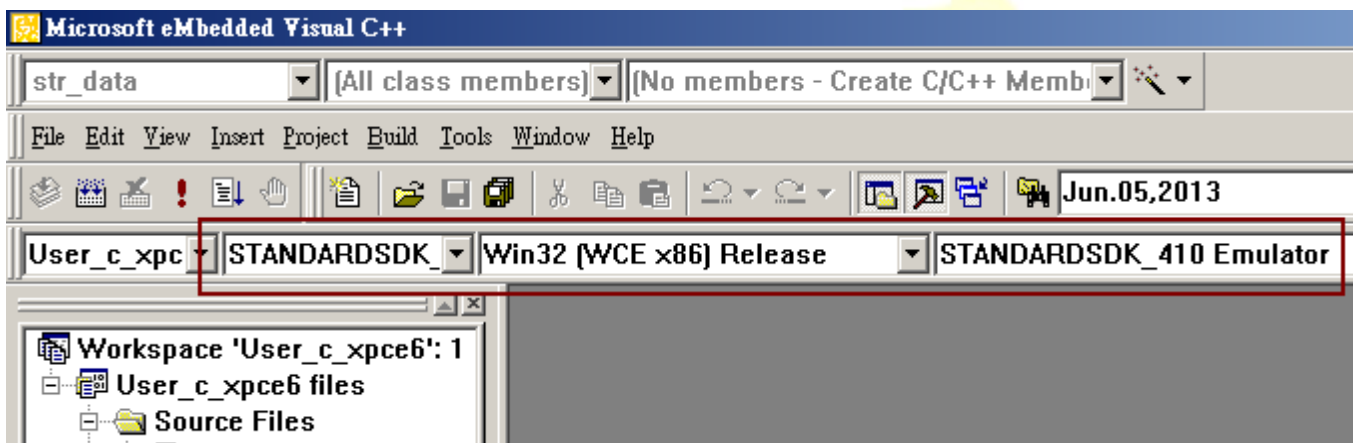
欲開發自定義的 C-function 或 C-function block 必須有以下的開發環境:

1. EVC++ 4.0 或
2. VS2008

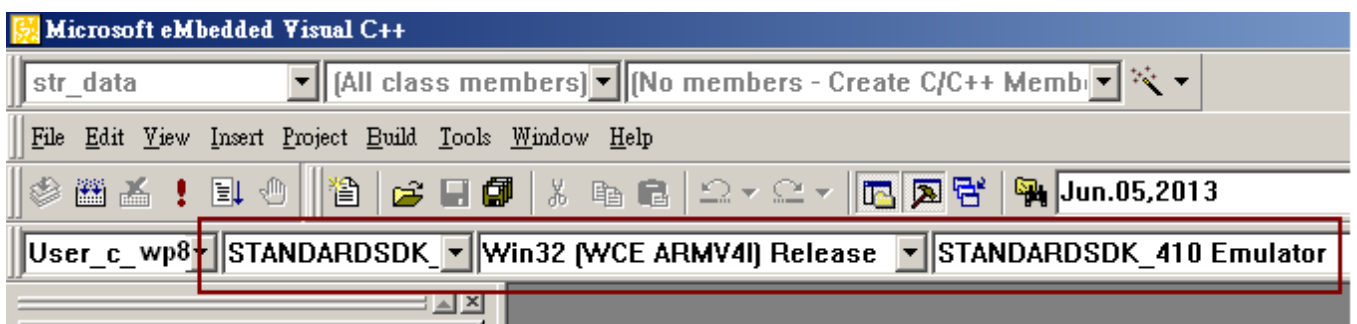
### 2.1. EVC 開發環境

確認 EVC++ 4.0 的 project 設定是否正確? (注意: 不同的 PAC 採用的設定會不相同)

下圖是 XP-8xx7-CE6 與 XP-8xx7-ATOM-CE6 需使用的設定: 必須是 “STANDARDSDK” 與 “Win32 (WCE **x86**) Release” 與 “STANDARDSDK\_4xx Emulator”.



下圖是 WP-8xx7 與 WP-5147 與 VP-25W7 與 VP-4137 需使用的設定: 必需是 “STANDARDSDK” 與 “Win32 (WCE **ARMV4I**) Release” 與 “STANDARDSDK\_4xx Emulator”.



Classification	ISaGRAF Chinese FAQ-167						
Author	Chun Tsai	Version	1.3	Date	Feb. 2014	Page	7 / 31

## 2.2. VS2008 開發環境

下載相關 SDK 開發套件：

客戶可在下列網頁下載相關的開發套件 (SDK)

1. 適用 XPAC (XP-8xx7-CE6, XP-8xx7-ATOM-CE6)

<http://www.icpdas.com/en/download/show.php?num=2488&nation=US&kind1=&model=&kw=wince6>

2. 適用 ViewPAC (VP-2xW7/2xW6, VP-4137)、WinPAC (WP-5147/WP-5146, WP-8xx7)

<http://www.icpdas.com/en/download/show.php?num=2489&nation=US&kind1=&model=&kw=wince5>

### 2.2.1. 安裝 ViewPAC 或 WinPAC 的開發套件 (SDK)

注意：請確認你的 PC 內已經安裝了 Microsoft VS2008，才能進行以下步驟：

1. 滑鼠雙擊下載的 SDK 檔案 (例如: pac270\_sdk\_20121015.msi)，將其安裝至 VS2008 中。



2. 點擊“Next”按鈕。

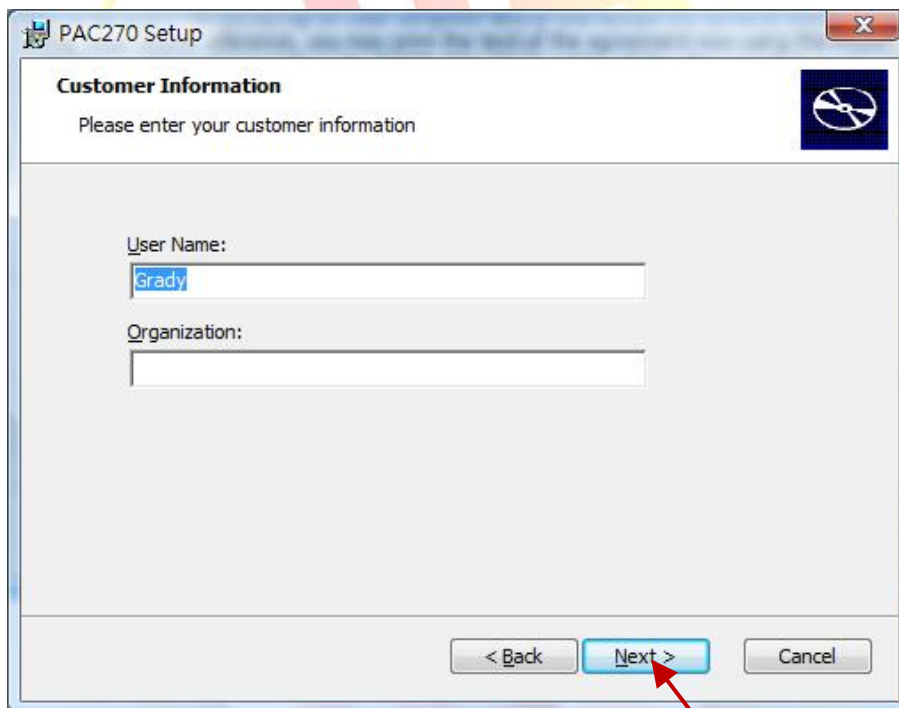


Classification	ISaGRAF Chinese FAQ-167						
Author	Chun Tsai	Version	1.3	Date	Feb. 2014	Page	8 / 31

3. 點選 “Accept”，再點擊 “Next” 按鈕。



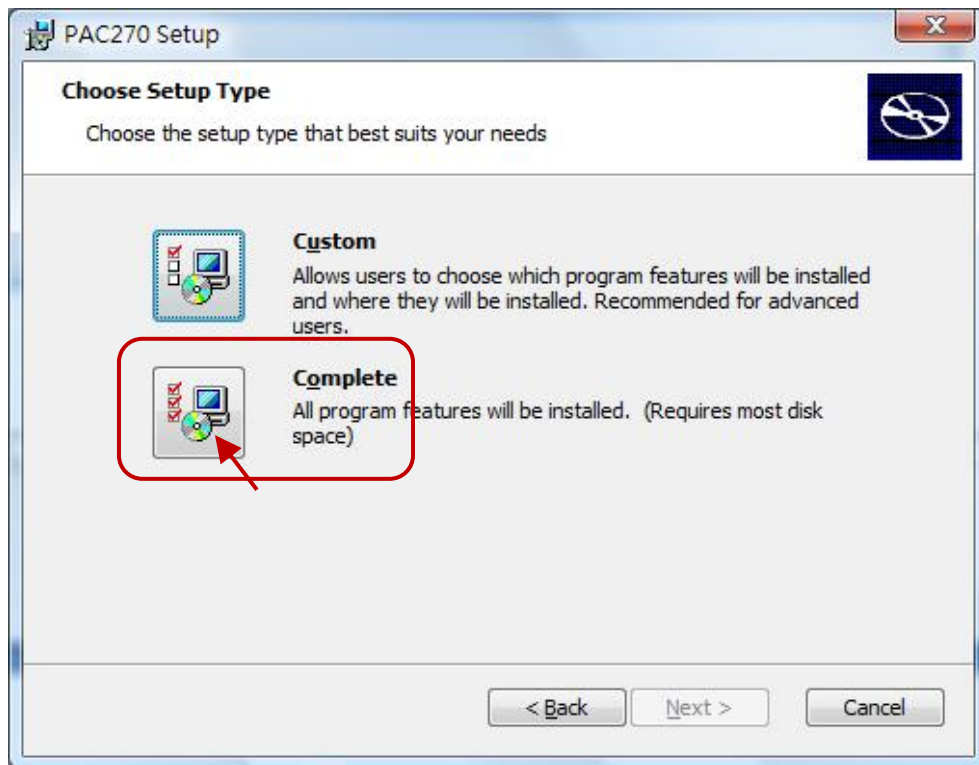
4. 點擊 “Next” 按鈕。



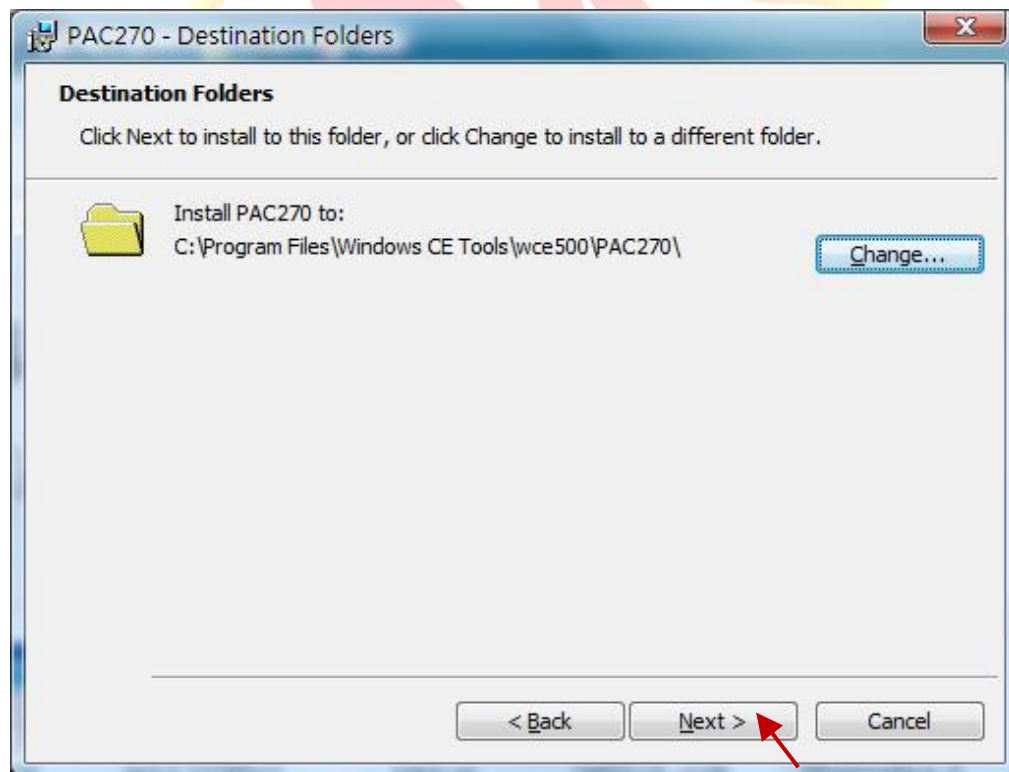


Classification	ISaGRAF Chinese FAQ-167					
Author	Chun Tsai	Version	1.3	Date	Feb. 2014	Page 9 / 31

5. 點擊 “Complete” 按鈕.

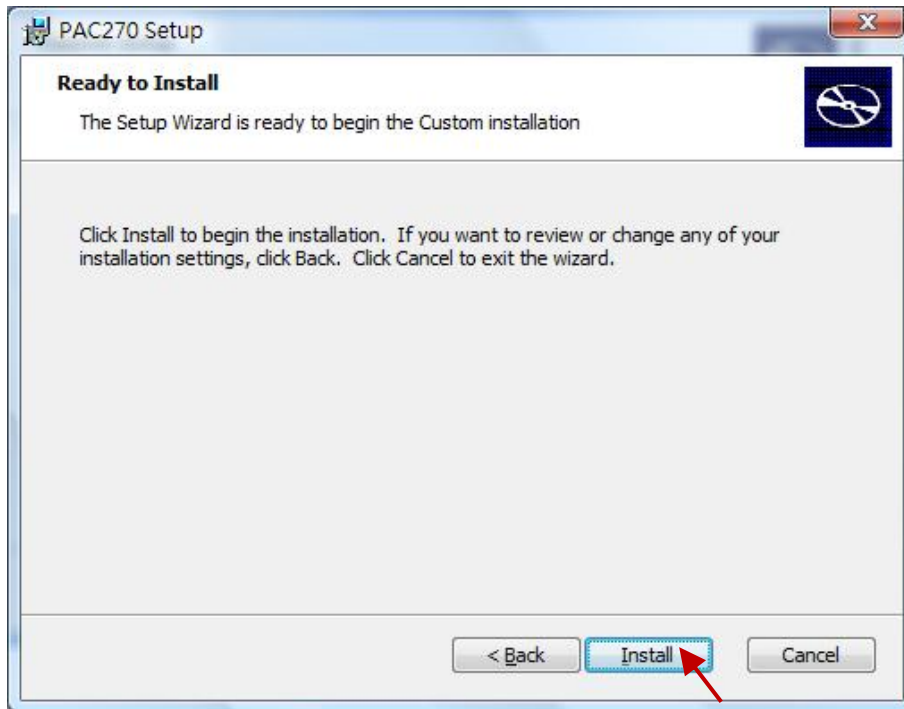


6. 點擊 “Next” 按鈕.

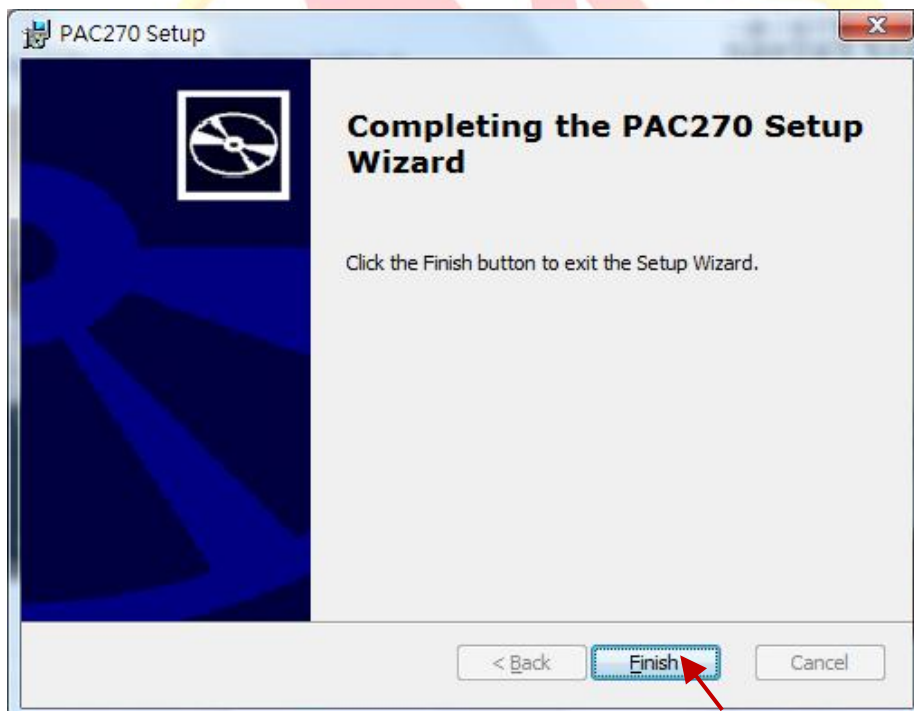


Classification	ISaGRAF Chinese FAQ-167						
Author	Chun Tsai	Version	1.3	Date	Feb. 2014	Page	10 / 31

7. 點擊 “Install” 按鈕，開始安裝 SDK.



8. 待安裝結束後，點擊 “Finish” 按鈕離開安裝程序.



Classification	ISaGRAF Chinese FAQ-167						
Author	Chun Tsai	Version	1.3	Date	Feb. 2014	Page	11 / 31

### 2.2.2. 安裝 XPAC (XP-8xx7-CE6, XP-8xx7-ATOM-CE6) 的開發套件 (SDK)

注意：請確認你的 PC 內已經安裝了 Microsoft VS2008 軟體，才能進行以下步驟：

1. 滑鼠雙擊下載的 SDK 檔案 (例如: pacsdk\_ce\_1.4.3\_vs2008.msi)，將其安裝至 VS2008 中。

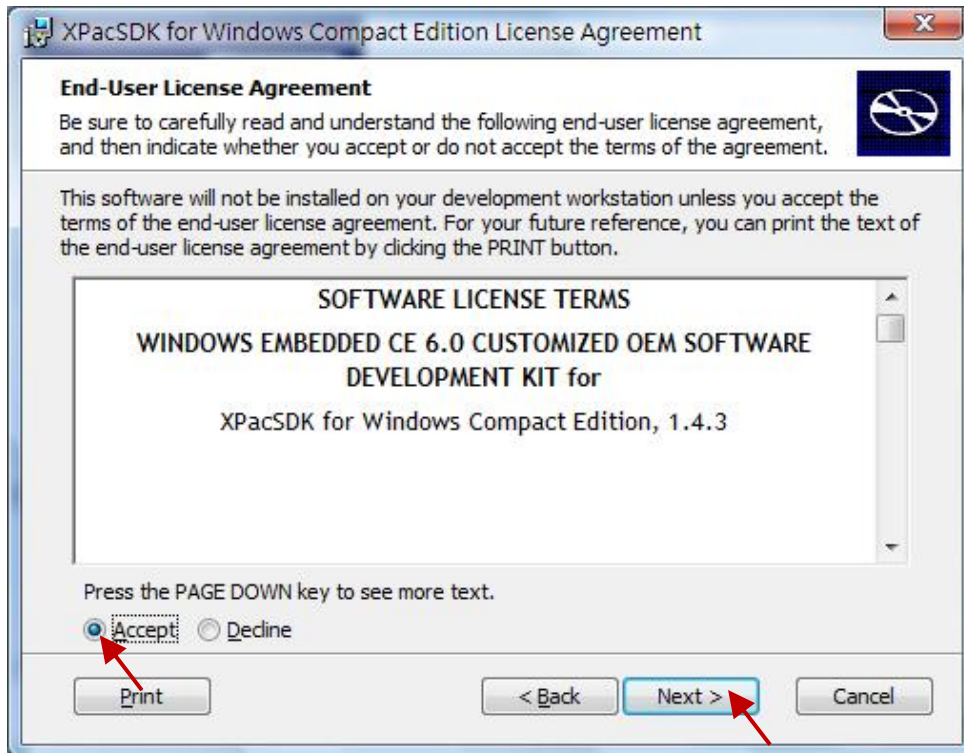


2. 點擊 “Next” 按鈕。

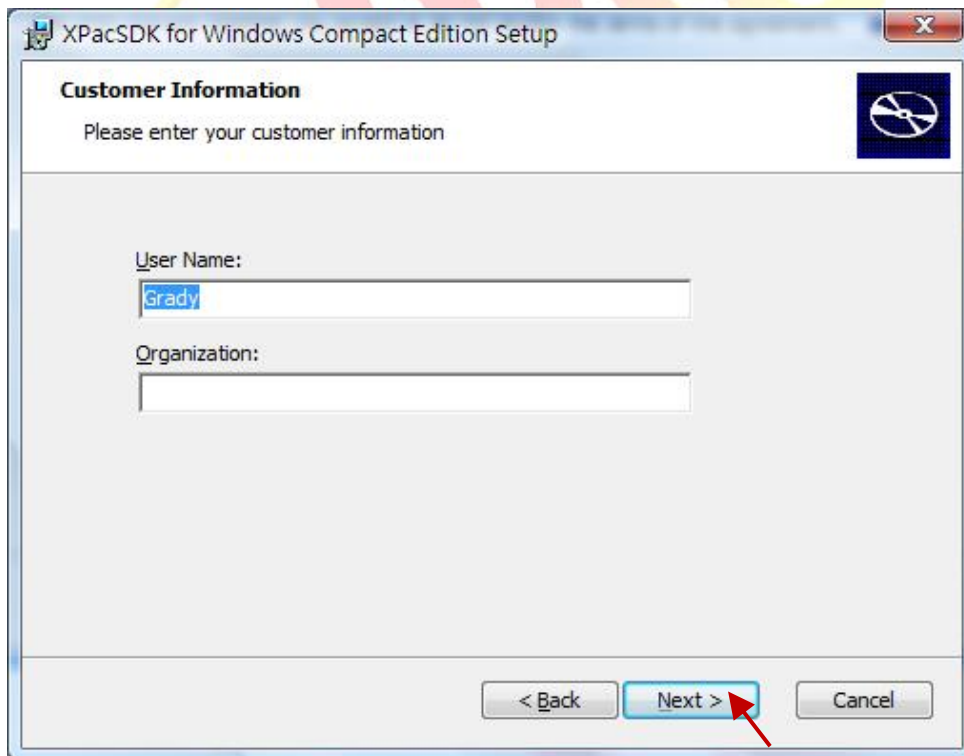


Classification	ISaGRAF Chinese FAQ-167						
Author	Chun Tsai	Version	1.3	Date	Feb. 2014	Page	12 / 31

3. 點選 “Accept”，再點擊 “Next” 按鈕。

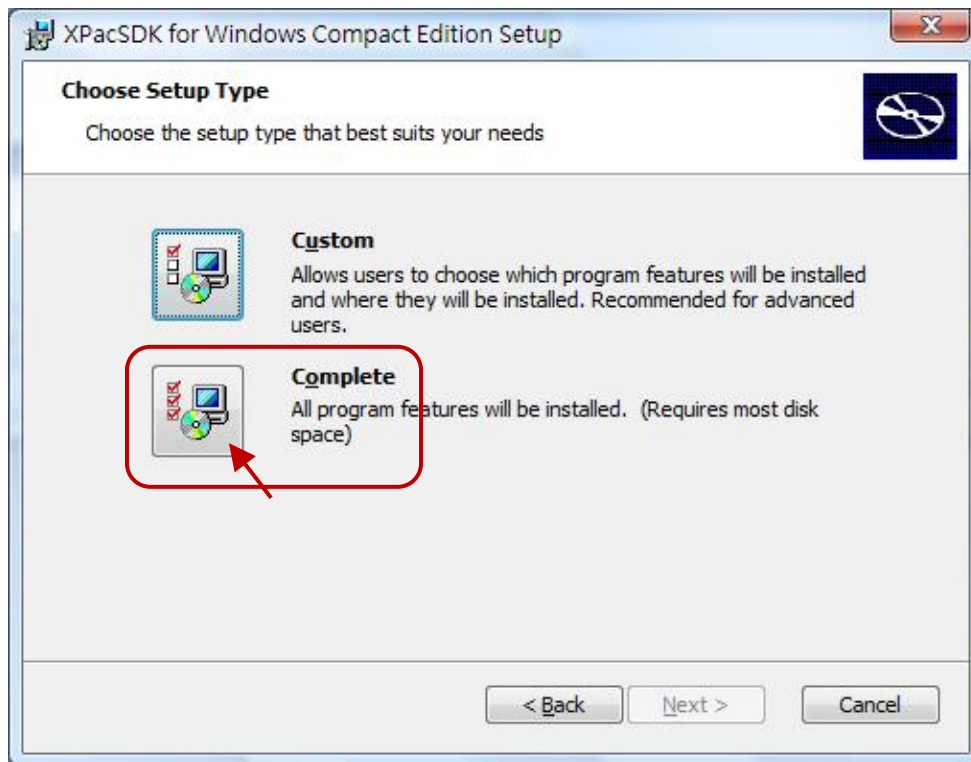


4. 點擊 “Next” 按鈕。

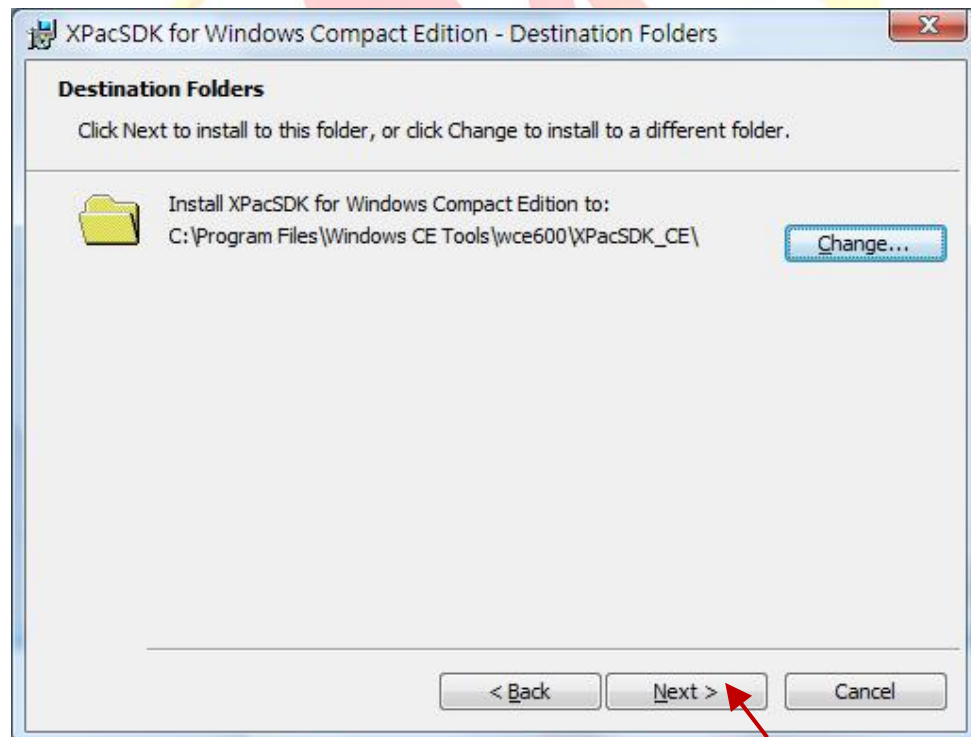


Classification	ISaGRAF Chinese FAQ-167					
Author	Chun Tsai	Version	1.3	Date	Feb. 2014	Page 13 / 31

5. 點擊 “Complete” 按鈕.

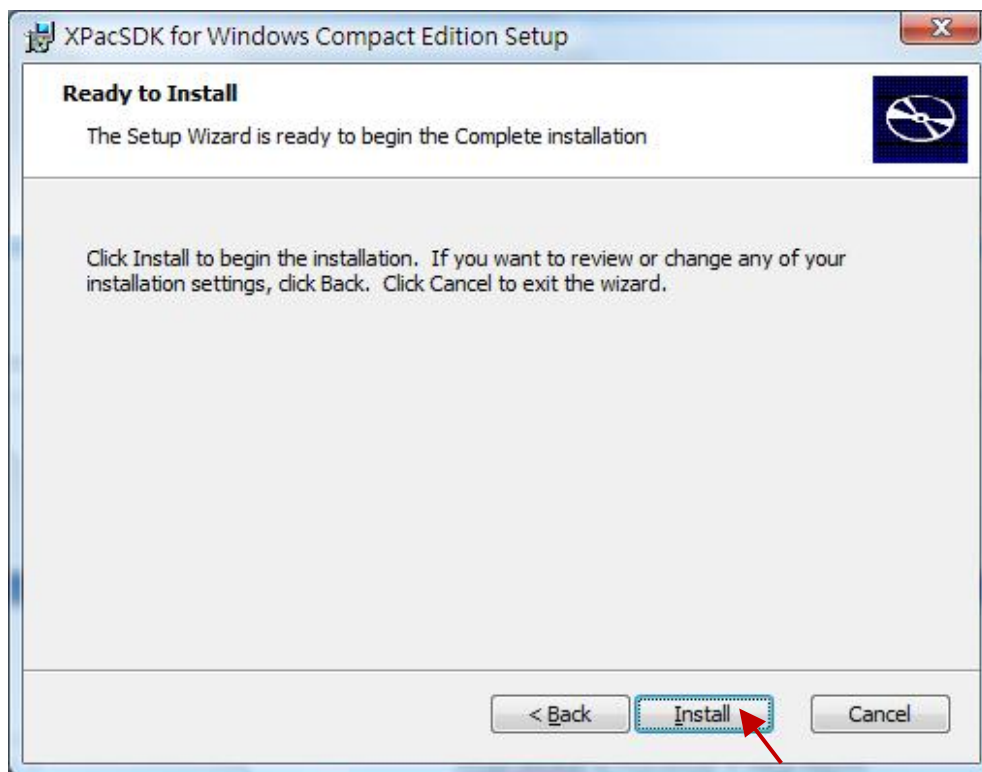


6. 點擊 “Next” 按鈕.

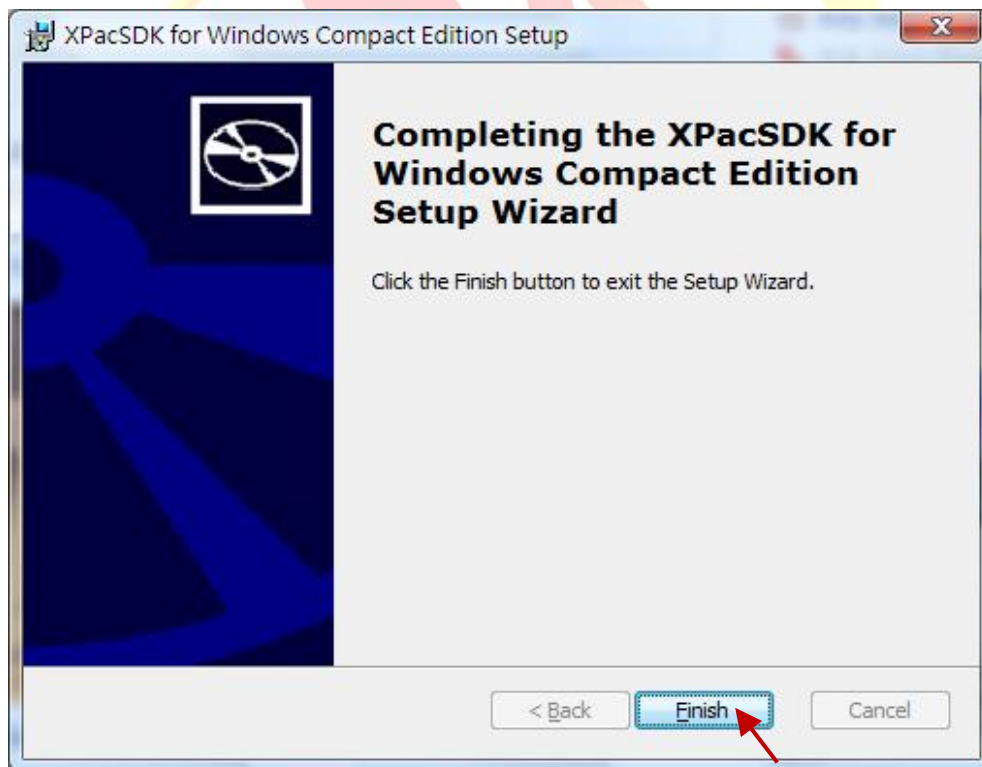


Classification	ISaGRAF Chinese FAQ-167						
Author	Chun Tsai	Version	1.3	Date	Feb. 2014	Page	14 / 31

7. 點擊 “Install” 按鈕, 開始安裝 SDK.



8. 待安裝結束後, 點擊 “Finish” 按鈕離開安裝程序.



Classification	ISaGRAF Chinese FAQ-167						
Author	Chun Tsai	Version	1.3	Date	Feb. 2014	Page	15 / 31

### 3. 定義 C-function 或 C-function block

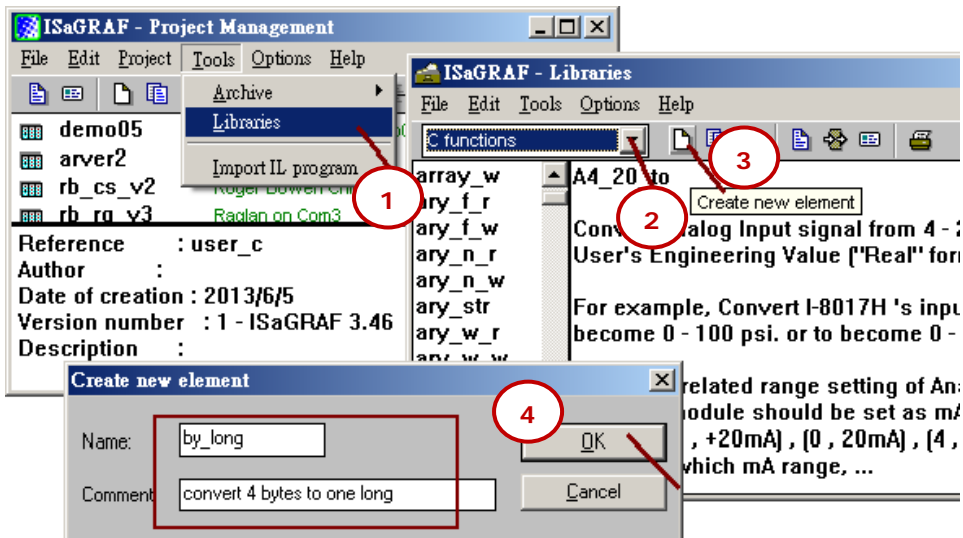
#### 3.1. 定義 C-function Lib

要建立自定的 C-Function, 首先需定義好它的 ISaGRAF Libraries.

例如, “by\_long” 這個範例是將 4 個 byte (0 ~ 255) 轉換成一個長整數 (32-bit signed integer).

步驟如下:

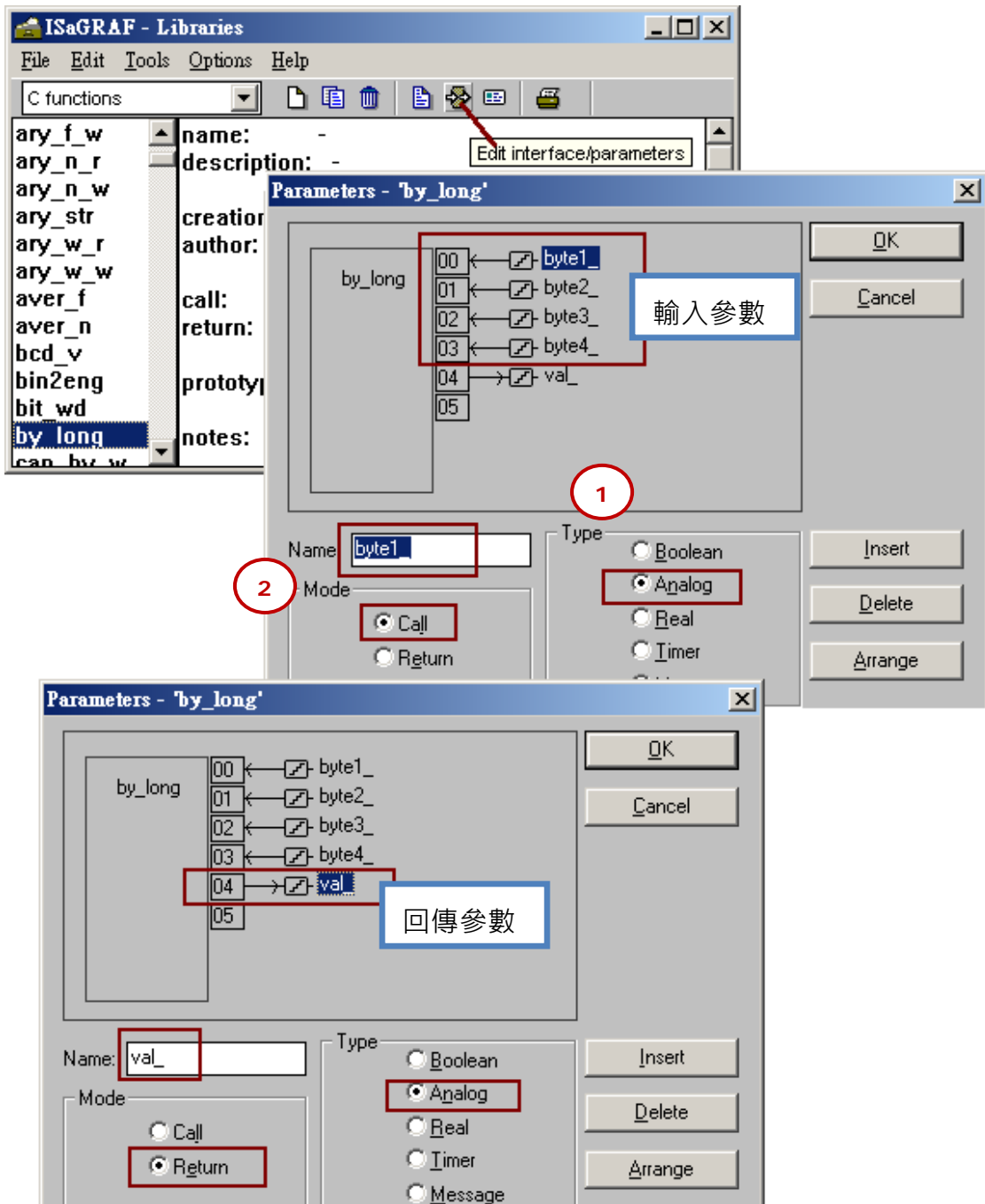
1. 使用 ISaGRAF workbench 建立一個新的 C-functions.



Classification	ISaGRAF Chinese FAQ-167					
Author	Chun Tsai	Version	1.3	Date	Feb. 2014	Page 16 / 31

2. 定義 C-function 的 輸入 與 回傳 參數:

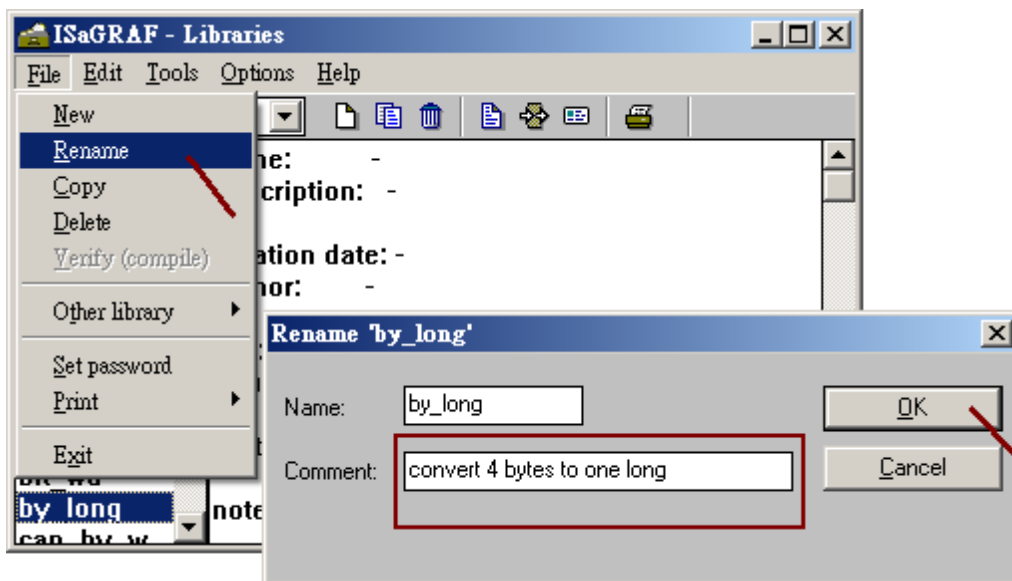
下圖顯示此 C-function 有 4 個輸入參數, 分別是 “byte1\_”, byte2\_”, “byte3\_” 與 “byte4\_”. 它們的型態都是 “Analog” (表示 32-bit signed integer), 而 Mode 的設定為 “Call” (表示輸入參數). 請在每個參數名稱後方加一個字元 “\_”. 主要是因為輸入 與 回傳參數會被 ISaGRAF 認定為保留字, 不可在程式內使用. 所以加個 “\_” 會避開程式內一些用到保留字的機會. 此 C-function 的回傳參數為 “val\_”, 而 Mode 設定是 “Return”, 型態是 “Analog”.



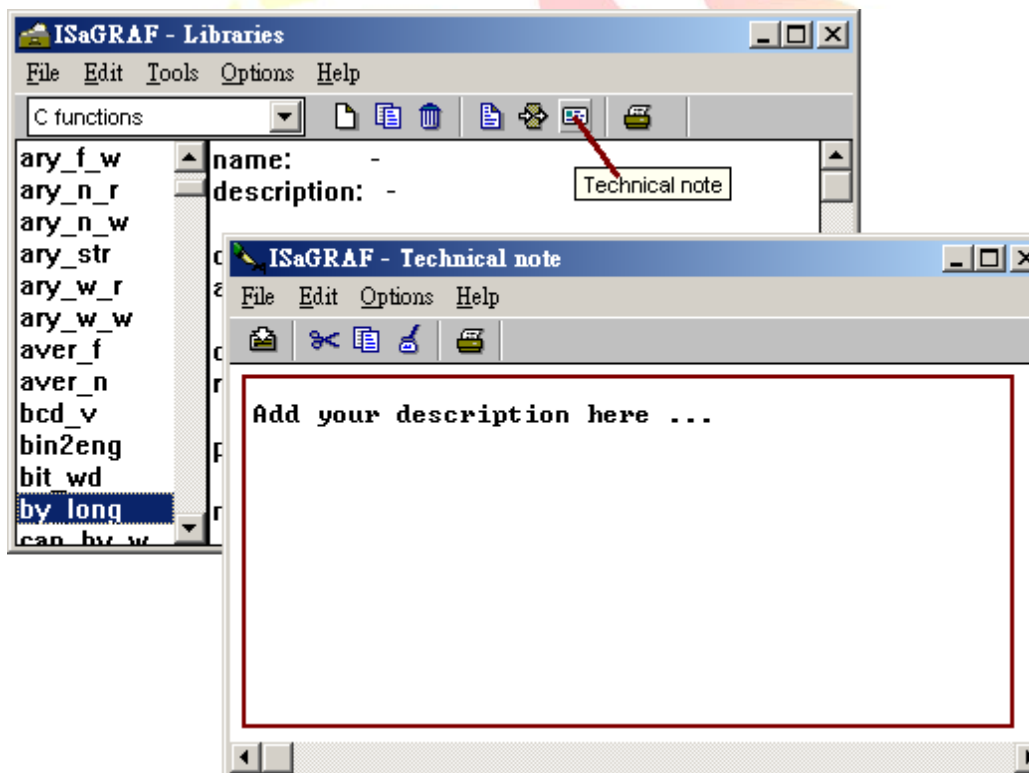


Classification	ISaGRAF Chinese FAQ-167					
Author	Chun Tsai	Version	1.3	Date	Feb. 2014	Page 17 / 31

3. 為此 C-function 加上 註解 說明文字:



4. 為此 C-function 加上 技術 說明文字:



Classification	ISaGRAF Chinese FAQ-167						
Author	Chun Tsai	Version	1.3	Date	Feb. 2014	Page	18 / 31

5. 將此 C-function 的 header 範本 (prototype), 複製到一個純文字編輯器 (比如, 記事本) 並取名為 “by\_long.h”.

**1** Click "Edit source code"

**2** Select all code on top and use mouse right click to select "Copy".

**3** Use a text editor to create a new file "by\_long.h". then use mouse right click to select "paste" to paste the header code of this c-function.

```

name: -
ISaGRAF - C source file
File Edit Options Help
user procedure interface
name: by_long
typedef long T_BOO;
typedef long T_ANA;
/*
user procedure
name: by_long
*/
by_long.h 記事本
# 檔案(F) 編輯(E) 格式(O) 檢視(V) 說明(H)
typedef char *T_MSG;
typedef struct {
/* CALL */ T_ANA _byte1;
/* CALL */ T_ANA _byte2;
/* CALL */ T_ANA _byte3;
/* CALL */ T_ANA _byte4;
/* RETURN */ T_ANA _val;
} str_arg;
#define BYTE1_ (arg->_byte1_
#define BYTE2_ (arg->_byte2_
#define BYTE3_ (arg->_byte3_
#define BYTE4_ (arg->_byte4_
#define U01_ (arg->_u01_

```

Classification	ISaGRAF Chinese FAQ-167						
Author	Chun Tsai	Version	1.3	Date	Feb. 2014	Page	19 / 31

6. 將此 C-function 的 C 程式範本, 複製到一個純文字編輯器 (比如, 記事本) 並取名為 "by\_long.c".

The image illustrates the process of copying C code from ISaGRAF to a text editor. It shows two windows: 'ISaGRAF - C source file' and 'by\_long.c - 記事本'.

**Step 1:** In the ISaGRAF window, the code for the 'user procedure' is selected. A context menu is open, and 'Copy' is highlighted. A red box with the number '1' and an arrow points to the 'Copy' option. A text box says: "Select all code and use mouse right click to select 'Copy'".

**Step 2:** In the 'by\_long.c - 記事本' window, the code is pasted. A context menu is open, and '貼上 (P)' (Paste) is highlighted. A red box with the number '2' and an arrow points to the '貼上 (P)' option. A text box says: "Then use mouse right click to paste it to a text editor and name it as 'by\_long.c'".

```

/*
 user procedure interface
 name: by_long
*/

typedef long T_BOO;
typedef long T_ANA;

user procedure
 name: by_long
*/
#include <tasy0def.h>
#include <grus0206.h>
void USP_by_long <str
<
>
UFP us
<
sys_
retl
>
/*
 user procedure
 name: by_long
*/
#include <tasy0def.h>
#include <grus0206.h>

void USP_by_long (str_arg *arg)
{
}

UFP uspdef_by_long (char *name)
{
 sys_strcpy (name, "BY_LONG");
}

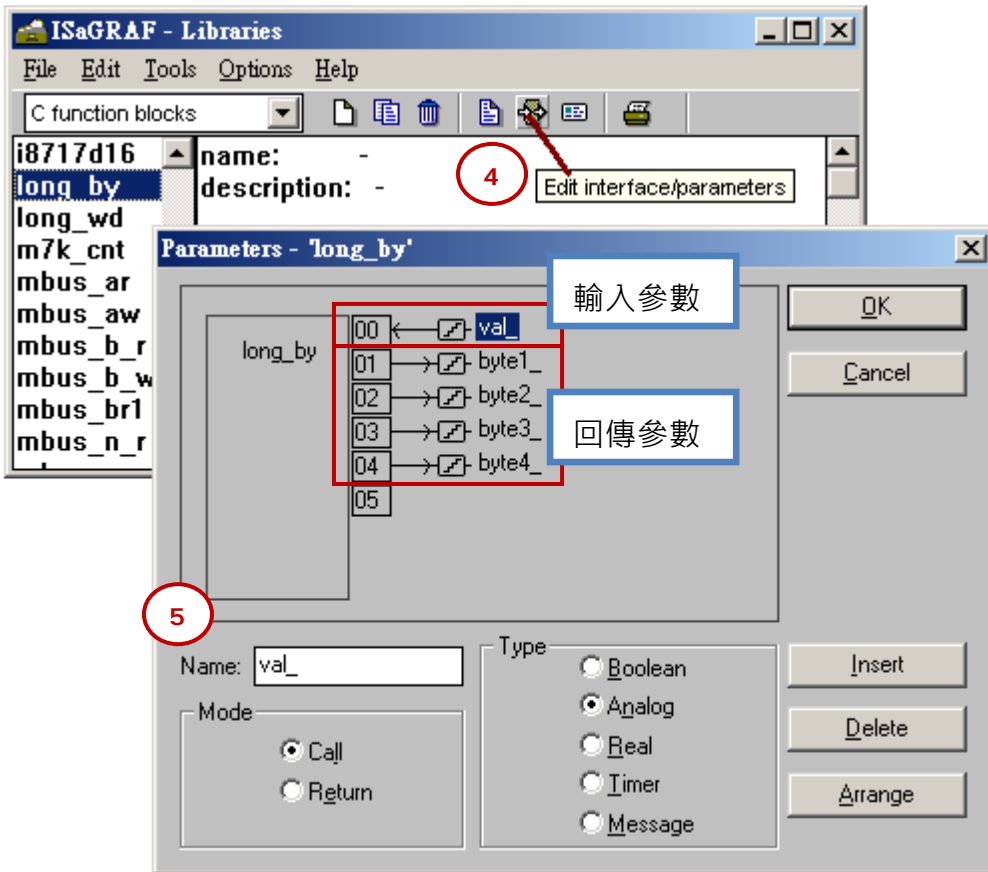
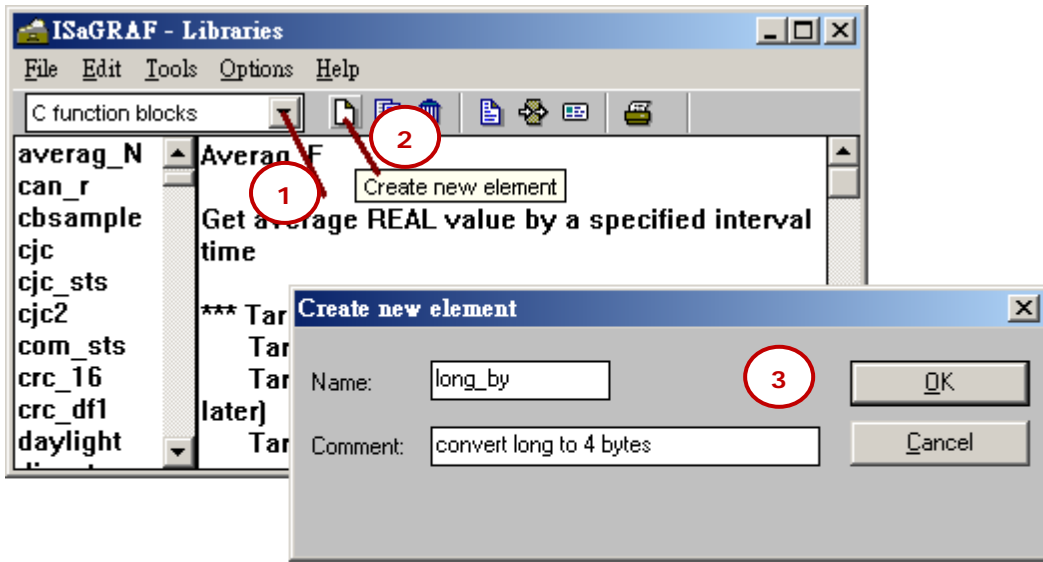
```

Classification	ISaGRAF Chinese FAQ-167					
Author	Chun Tsai	Version	1.3	Date	Feb. 2014	Page 20 / 31

### 3.2. 定義 C-function block Lib

本節的方法與前一節類似 (請參考前一節). 本例是要定義出 C-function block 範例 “long\_by” 的 ISaGRAF Lib 與 產生它的範本檔案“long\_by.h” 與 “long\_by.c” .

1. 此 C-function block 有 1 個 輸入參數 與 4 個 回傳參數, 如下圖.



Classification	ISaGRAF Chinese FAQ-167						
Author	Chun Tsai	Version	1.3	Date	Feb. 2014	Page	21 / 31

2. 範本檔案 “long\_by.h” 與 “long\_by.c” 產生的方法 與 前一節 (步驟 5,6) 類似.

```

long_by.h - 記事本
檔案(F) 編輯(E) 格式(O) 檢視(V) 說明(H)
/*
function block interface
name: long_by
*/

typedef long T_B00;
typedef long T_ANA;
typedef float T_REAL;
typedef long T_TMR;
typedef long

long_by.c - 記事本
檔案(F) 編輯(E) 格式(O) 檢視(V) 說明(H)
typedef s
/* CAL
} str_ar
#define C
/*
#include <tasy0def.h>
#include <grfb0162.h>

typedef
struct {

} str_data;

uint16 FBINIT_long_by (uint16 hinstance)
{
return (sizeof (str_data));
}

void FBACT_long_by (uint16 hinstance, str_data *data, str_
{
}
第 40 列, 第 1 行

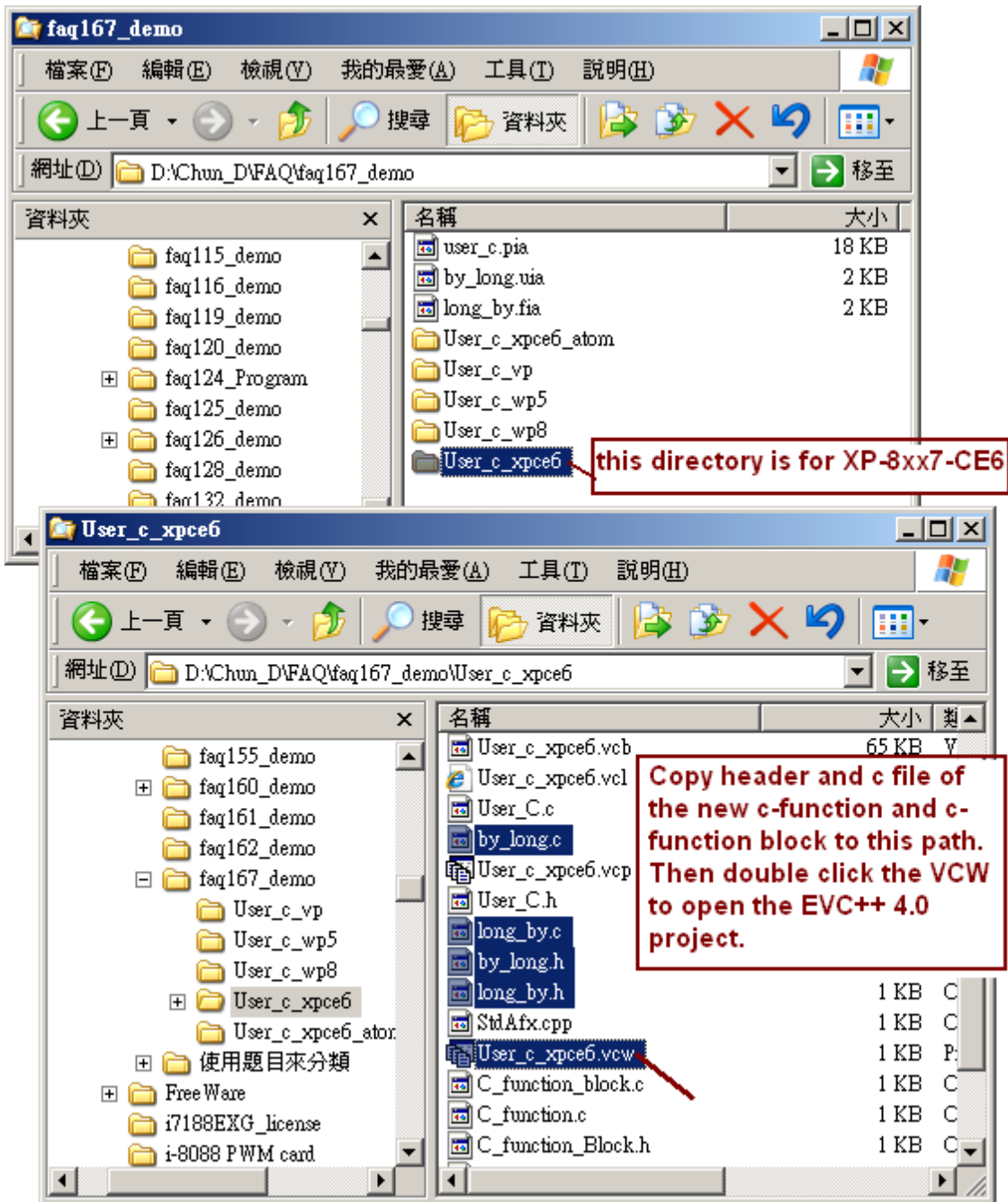
```

Classification	ISaGRAF Chinese FAQ-167						
Author	Chun Tsai	Version	1.3	Date	Feb. 2014	Page	22 / 31

#### 4. 編寫 C-function 或 C-function block 的邏輯程式

請確認你的 PC 內已經安裝了 Microsoft EVC++ 4.0 或 VS 2008 軟體，才能進行以下步驟：

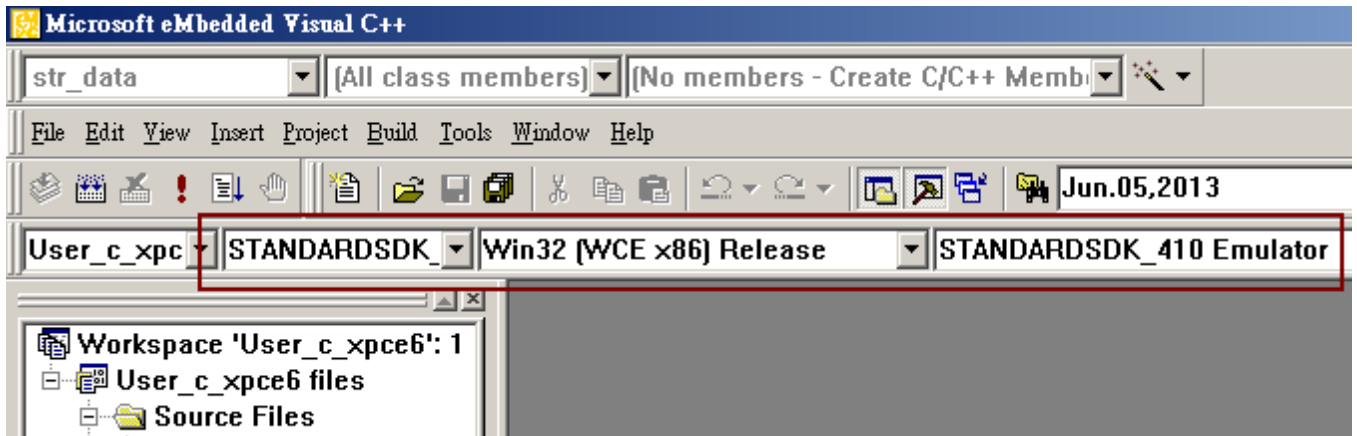
1. 先把 C-function 與 C-function block 的 header 與 C 範本檔案，複製到你的 PC 內的相對應的專案資料夾內，然後用 Mouse 雙擊該資料夾內的 VCW 檔案來開啟這個 EVC++ 4.0 專案 (此處是以 XP-8xx7-CE6 當例子)



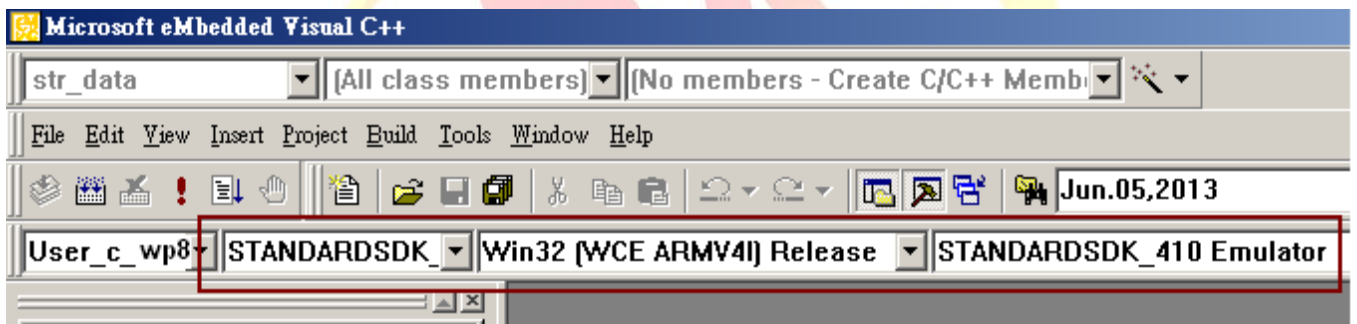
Classification	ISaGRAF Chinese FAQ-167						
Author	Chun Tsai	Version	1.3	Date	Feb. 2014	Page	23 / 31

2. 接下來，需確認 EVC++ 4.0 的 project 設定是否正確？  
(注意: 不同的 PAC 採用的設定會不相同)

下圖是 XP-8xx7-CE6 與 XP-8xx7-ATOM-CE6 需使用的設定：必需是  
“STANDARDSDK” 與 “Win32 (WCE **x86**) Release” 與 “STANDARDSDK\_4xx Emulator”。

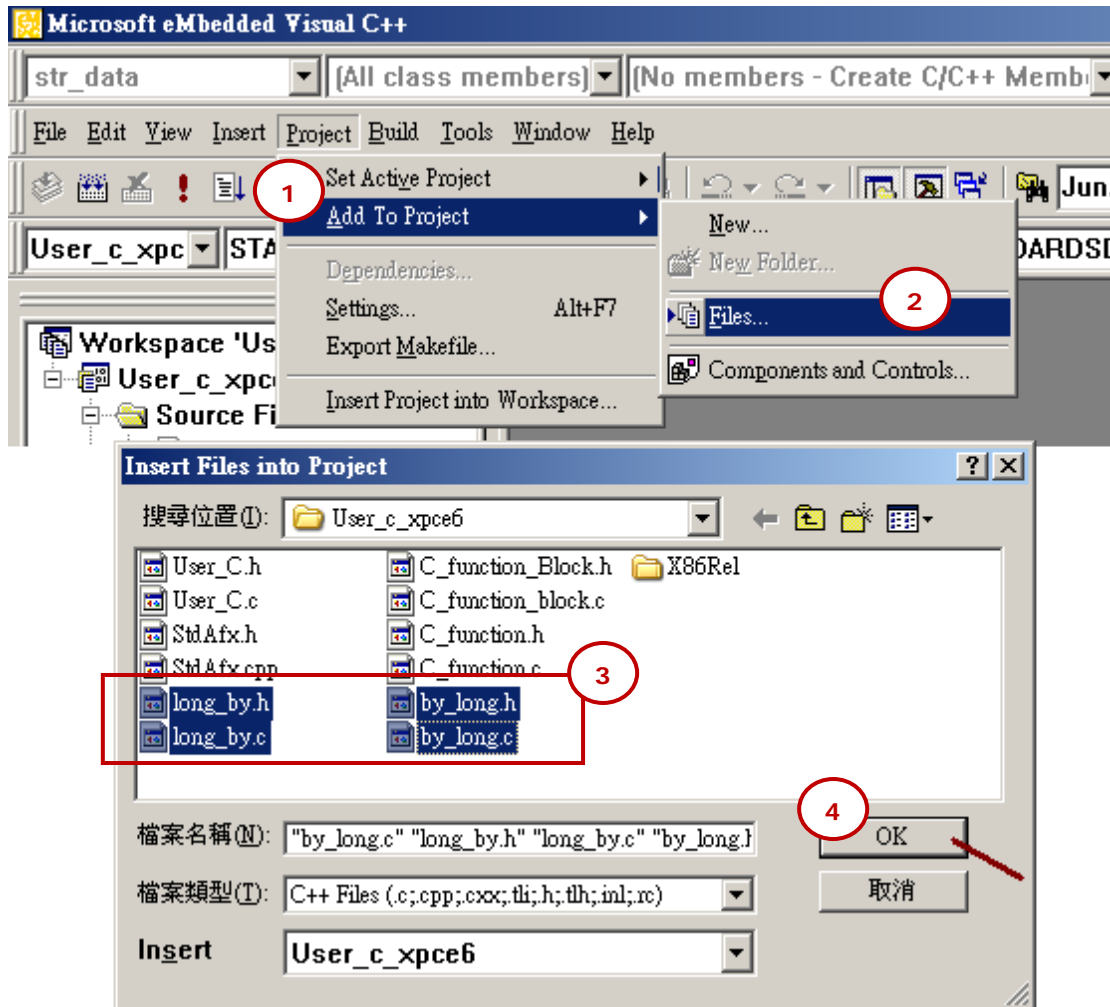


下圖是 WP-8xx7 與 WP-5147 與 VP-25W7 與 VP-4137 需使用的設定：必需是  
“STANDARDSDK” 與 “Win32 (WCE **ARMV4I**) Release” 與 “STANDARDSDK\_4xx Emulator”。



Classification	ISaGRAF Chinese FAQ-167						
Author	Chun Tsai	Version	1.3	Date	Feb. 2014	Page	24 / 31

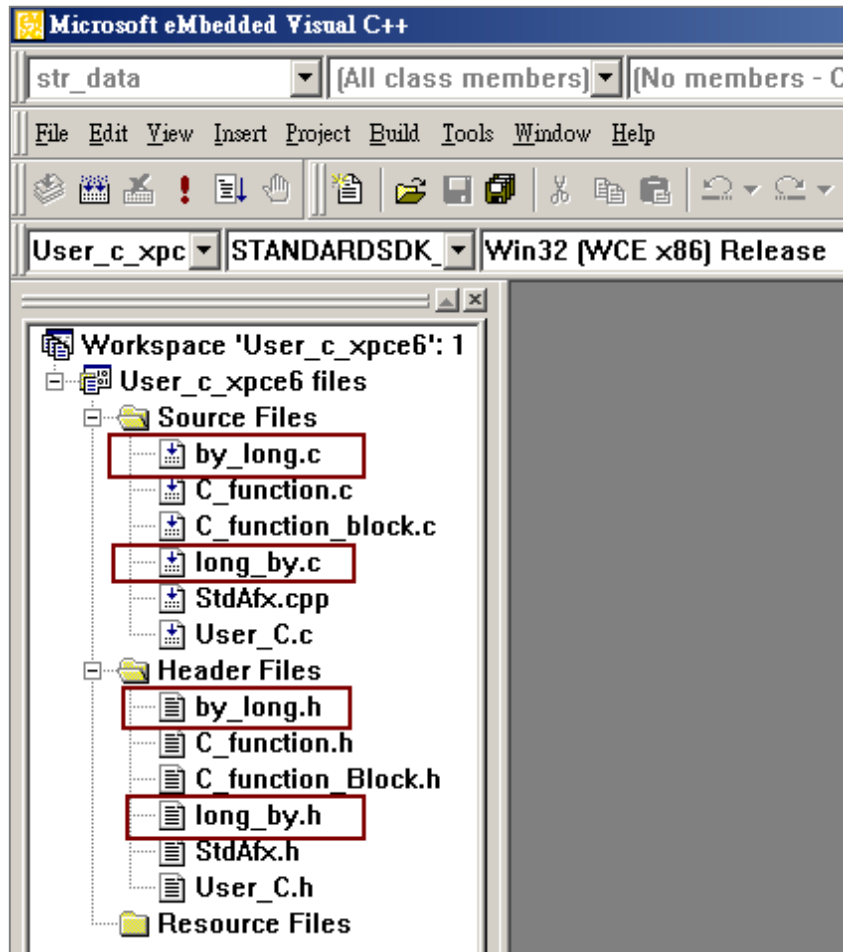
然後把 C-function 與 C-function block 的 header 與 C 檔案 加進此 EVC++ project 內 .





Classification	ISaGRAF Chinese FAQ-167					
Author	Chun Tsai	Version	1.3	Date	Feb. 2014	Page 25 / 31

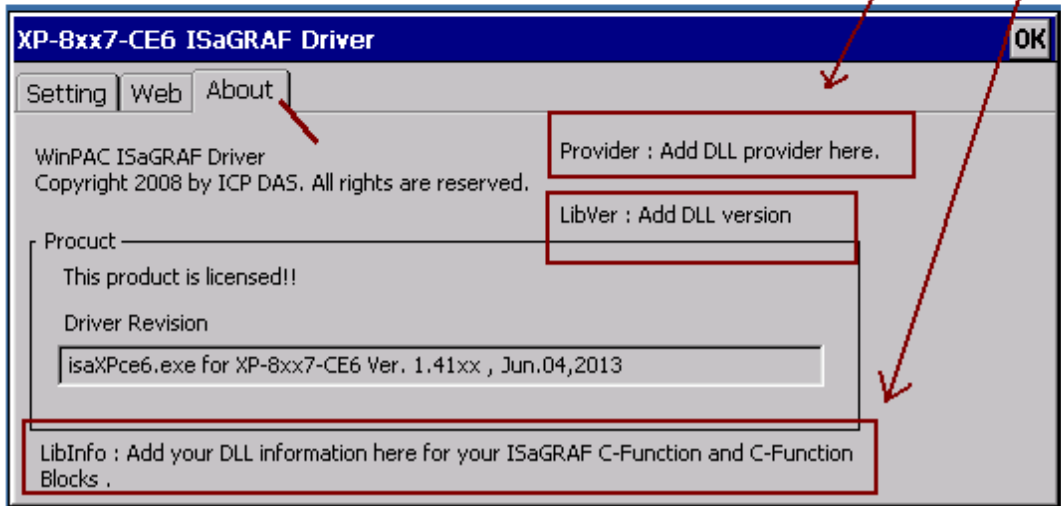
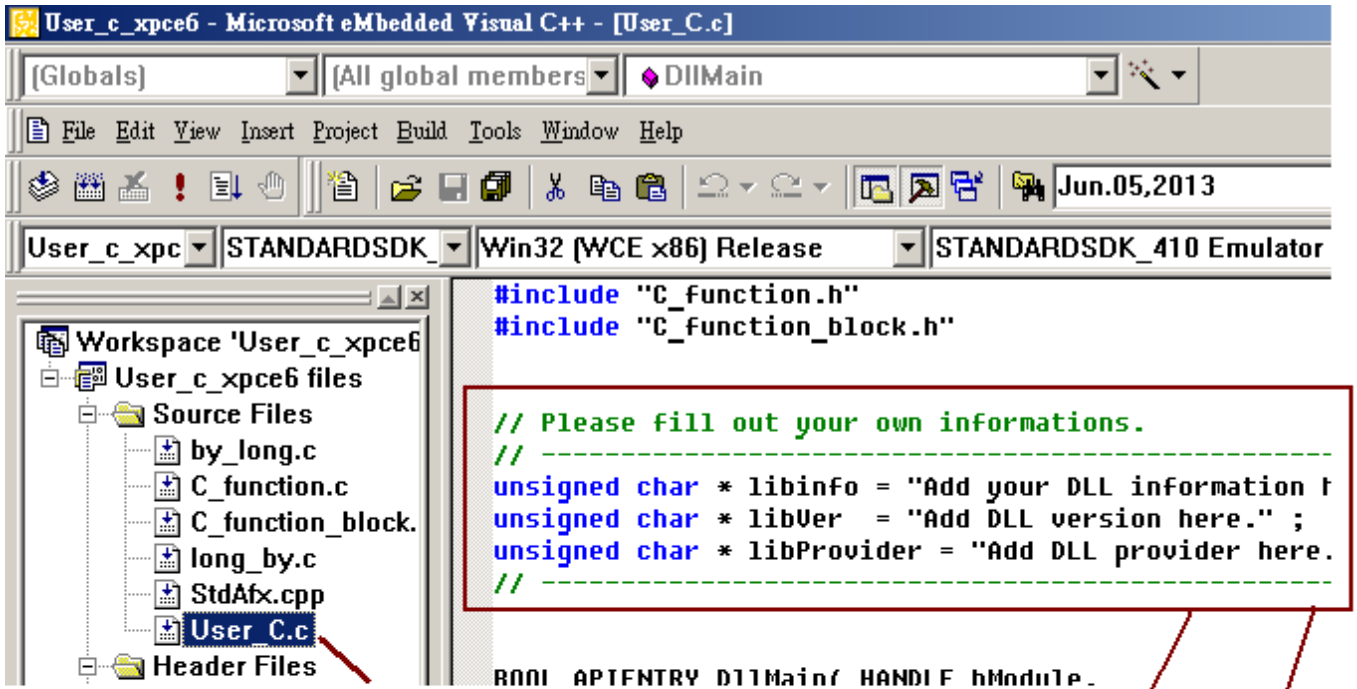
之後會顯示類似下方的 Project.



Classification	ISaGRAF Chinese FAQ-167						
Author	Chun Tsai	Version	1.3	Date	Feb. 2014	Page	26 / 31

### 4.1. 編寫 “User\_C.c”

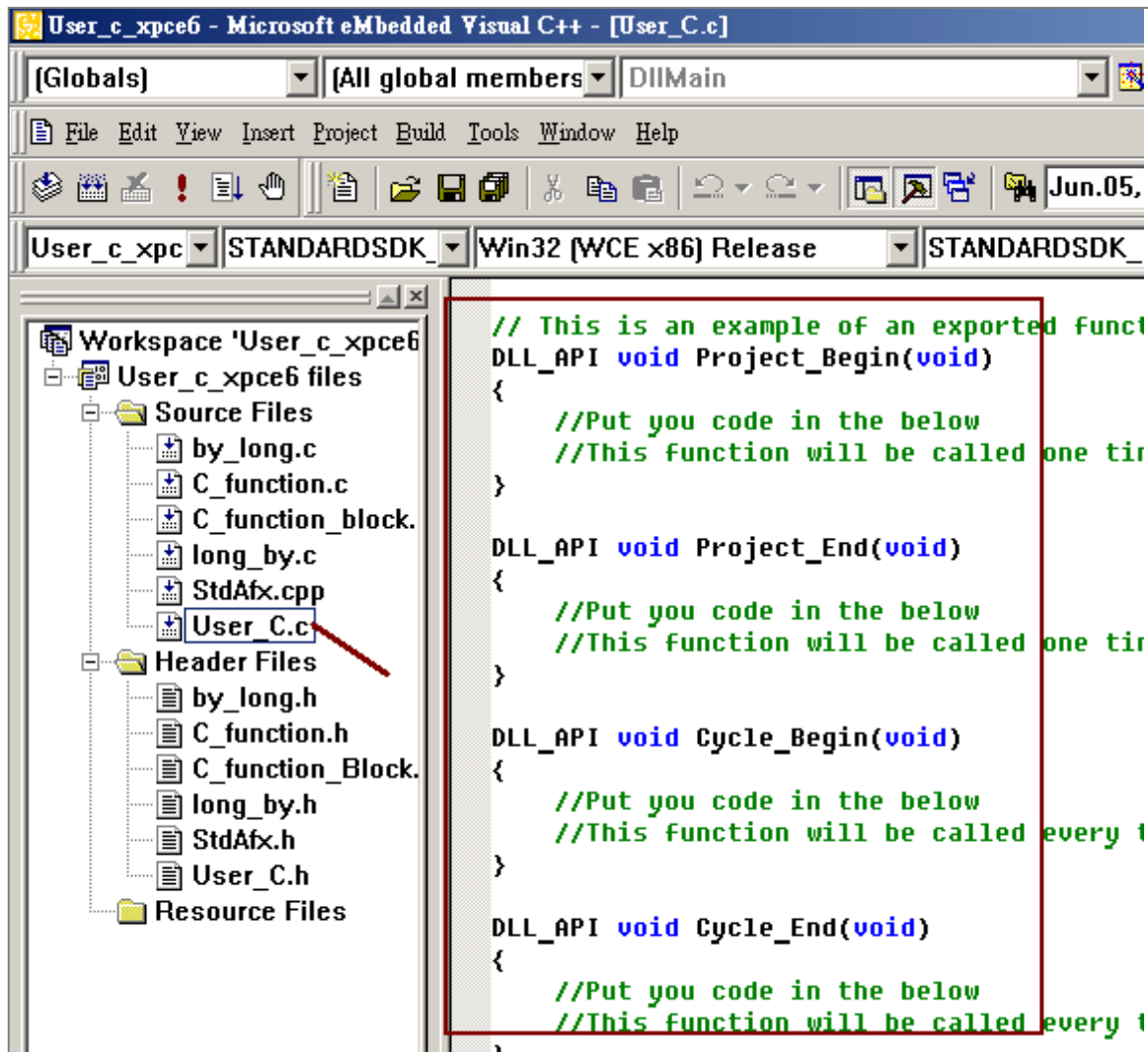
請把下圖內 “User\_C.c” 內的 “libinfo” 與 “libVer” 與 “libProvider” 所定義的資訊，修改成你想要顯示的資訊。這 3 個資訊會顯示在控制器的 ISaGRAF 驅動畫面的 “About” 視窗上。



Classification	ISaGRAF Chinese FAQ-167						
Author	Chun Tsai	Version	1.3	Date	Feb. 2014	Page	27 / 31

如果你想要在 ISaGRAF 專案程式 “一開始 Run” 與 “結束 Run 前” 進行一些動作，請編寫 “User\_C.c” 內 “Project\_Begin” 與 “Project\_End” 這 2 個函式的程式。

如果你想要在 ISaGRAF 專案開始 Run 之後的每一個 Scan Cycle 的前方 與 每一個 Scan Cycle 的後方 進行一些動作，請編寫 “Cycle\_Begin” 與 “Cycle\_End” 這 2 個函式的程式。



Classification	ISaGRAF Chinese FAQ-167					
Author	Chun Tsai	Version	1.3	Date	Feb. 2014	Page 28 / 31

## 4.2. 編寫 C-function 內的邏輯程式 (本例是 “by\_long.c” )

本例 “by\_long.c” 的輸入與回傳參數的變數定義是存放在 “by\_long.h” 內  
請記得將此 C-function 的 pointer 加到 “C\_function.c” 這個檔案內。

The image shows two screenshots of a development environment (likely Visual Studio) illustrating the implementation of a C-function named "by\_long".

**Top Screenshot (1):** Shows the source file "by\_long.c" in the workspace. The code includes headers for "User\_C.h" and "by\_long.h". The function "USP\_by\_long" is defined, which takes a pointer to a string "str\_arg" and returns a long value. The logic inside the function extracts the first four bytes of the string and shifts them to the right to form a long value. A call to "uspdef\_by\_long" is made, which registers the function.

**Bottom Screenshot (2):** Shows the source file "C\_function.c" in the workspace. It includes headers for "stdafx.h", "User\_C.h", and "C\_function.h". The function "uspdef\_by\_long" is declared. A list of function pointers, "UFP\_LIST UFPBUF", is defined, and "uspdef\_by\_long" is added to this list.

Annotations in red boxes and arrows provide additional context:

- "Add 'User\_c.h' Add 'by\_long.h'" points to the include statements in "by\_long.c".
- "Add your logic for this c-function." points to the logic inside the "USP\_by\_long" function.
- "uspdef\_by\_long" points to the function declaration in "C\_function.c".
- "uspdef\_by\_long" points to the function registration in the "UFP\_LIST UFPBUF" array in "C\_function.c".

Classification	ISaGRAF Chinese FAQ-167					
Author	Chun Tsai	Version	1.3	Date	Feb. 2014	Page 29 / 31

### 4.3. 編寫 C-function block 內的邏輯程式 (本例是 “long\_by.c” )

本例 “long\_by.c” 的 輸入 與 回傳參數的 變數定義 是存放在 “long\_by.h” 內。

```

#include "User_C.h"
#include "long_by.h"

typedef struct
{
    unsigned char AO[4];
} str_data;

uint16 FBINIT_long_by (uint16 hinstance)
{
    return (sizeof (str_data));
}

void FBACT_long_by (uint16 hinstance, str_data *data, str_arg *arg)
{
    data->AO[0] = (unsigned char)(VAL_ & 0xFF);
    data->AO[1] = (unsigned char)(VAL_>>8 & 0xFF);
    data->AO[2] = (unsigned char)(VAL_>>16 & 0xFF);
    data->AO[3] = (unsigned char)(VAL_>>24 & 0xFF);
}

#define BOO_VALUE ((T_BOO *)value)
#define ANA_VALUE ((T_ANA *)value)
#define REAL_VALUE ((T_REAL *)value)
#define TMR_VALUE ((T_TMR *)value)
#define MSG_VALUE ((T_MSG *)value)

void FBREAD_long_by (uint16 hinstance, str_data *data,
                    uint16 parno, void *value)
{
    switch(parno)
    {
        case FBLPNO_BYTE1_:
        case FBLPNO_BYTE2_:
        case FBLPNO_BYTE3_:
        case FBLPNO_BYTE4_:
            *ANA_VALUE = (T_ANA)data->AO[parno];
            break;
    }
}

ABP fbldf_long_by (char *name, IBP *initproc, RBP *readproc)
{
    sys_strcpy (name, "LONG_BY");
    *initproc = (IBP)FBINIT_long_by;
    *readproc = (RBP)FBREAD_long_by;
    return ((ABP)FBACT_long_by);
}

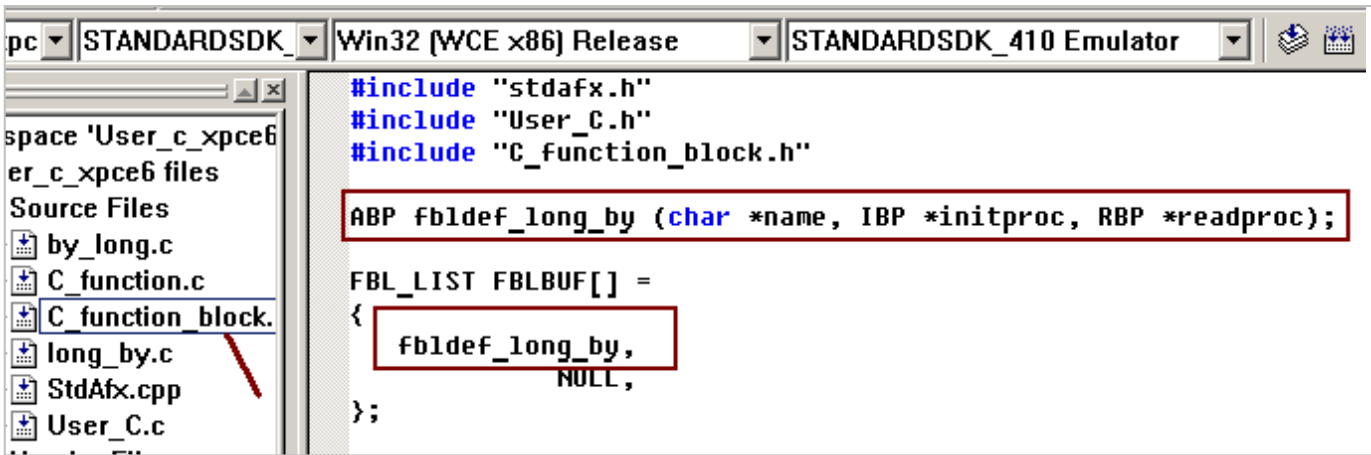
```

Annotations in the image:

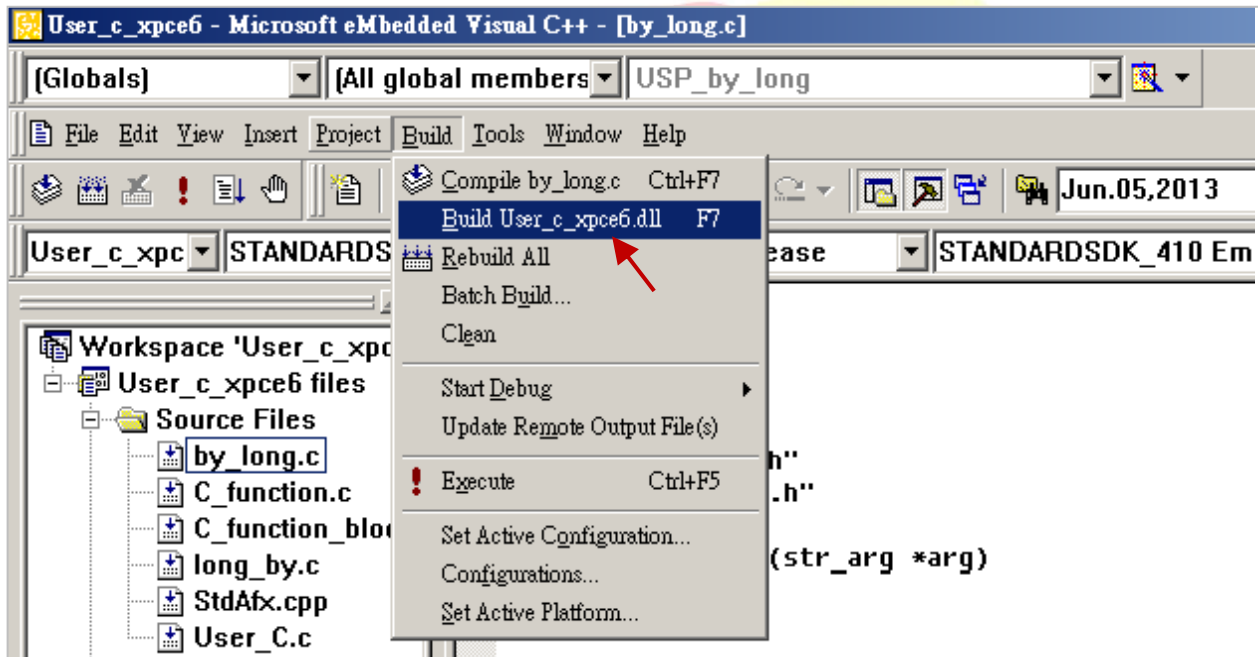
- Box 1: "Add 'User\_C.h' Add 'long\_by.h'" points to the #include lines.
- Box 2: "Each c-function block instance in the ISaGRAF has its own memory here." points to the unsigned char AO[4]; line.
- Box 3: "add logic for this c-function block here." points to the data->AO[] assignment lines.
- Box 4: "Add code to get return parameters of the c-function block." points to the switch statement.

Classification	ISaGRAF Chinese FAQ-167						
Author	Chun Tsai	Version	1.3	Date	Feb. 2014	Page	30 / 31

1. 請記得將此 C-function block 的 pointer 加到 “C\_function\_block.c” 這個檔案內。



2. 編譯 (Compile) 此 project 來產生 DLL :



3. 若編譯成功, 複製此 DLL 到你的 PAC 的正確路徑內, 重開機一次就可。

PAC 內要支持自定 C-function 與 C-function block 必須具備的 DLL 檔案:

XP-8xx7-CE6	\\System_disk\\ISaGRAF\\User_c_xpce6.dll
WP-8xx7	\\System_disk\\ISaGRAF\\User_c_wp8.dll
WP-5147	\\Micro_SD\\ISaGRAF\\User_c_wp5.dll
VP-25W7 / VP-4137	\\System_disk\\ISaGRAF\\User_c_vp.dll
XP-8xx7-ATOM-CE6	\\System_disk\\ISaGRAF\\User_c_xpce6_atom.dll

Classification	ISaGRAF Chinese FAQ-167						
Author	Chun Tsai	Version	1.3	Date	Feb. 2014	Page	31 / 31

## 5. 多執行緒的相關考量

ICP DAS 的 ISaGRAF WinCE 控制器內的 ISaGRAF Driver (驅動), 是使用 Microsoft EVC++ 4.0 編譯出來的一種 多執行緒 應用程式. 主要的執行緒 (非全部執行緒) 如下表.

這些執行緒 (thread) 的 Priority 設定是使用 “CeSetThreadPriority” 函式來設定. 越大的 Priority 設定值, 表示越低的 Priority. 相反的越小的 Priority 設定值, 表示越高的 Priority.

這些執行緒的 Slice 值是使用 “CeSetThreadQuantum” 函式來設定. “\_tmain” 這個執行緒是處理 ISaGRAF 每個 Scan cycle 內的動作, 它的 “sleep” 值是一個變動的數值, 跟 User 寫的 ISaGRAF 程式的 cycle scan time 有關 (一般是 Cycle time 越長, 該 Cycle 的 Sleep time 也會越長).

請不要在 DLL 內建立一個 Priority 設定值跟 113 一樣 或比 113 值還小的執行緒 (表示高 priority). 不然整個 PAC 的程式運作可能會垮掉. (最好是設成比 118 還大的 Priority 值)

User 自定的 C-function 與 C-function block 是在 “\_tmain” 執行緒內被調用.

Thread Name	Thread Priority WP-8xx7, WP-5147, VP-25W7, XP-8xx7-CE6 (Driver 1.44 or earlier version)	Thread Priority XP-8xx7-CE6 (Driver 1.44A or later version)	Slice (ms)	Sleep (ms)
_tmain( )	113	213	X	X
Go_Thread_pwm( )	109	209	1	1
init_SMS( )	116	216	1	50
Go_COM2_write( )	116	216	1	5
Go_COM2_write( )	116	216	1	5
WriteToXML	Above normal	Above normal	1	7
WriteToXML	116	216	1	7
CheckSMemory( )	116	216	1	8
InitEbus	118	218	1	6
Go_COMM	115	215	1	>7
Go_Other_Thread1	124	224	2	7
LP_thread( )	116	216	1	5
T_Go_RDN2( )	118	218	1	25
Thread_Go_RDN( )	118	218	1	60
Go_Thread_I8KE8	117	217	1	x
go_COM1_87K_init	116	216	1	4
Go_Thread_mbus_tcp	117	217	1	5