

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/root/product/solutions/softplc_based_on_pac/isagraf/download/isagraf-link.html

● 檔案說明

本文件 “FAQ167_chinese.pdf” (與 “FAQ167_demo_chinese.zip”) 可以從以下網址取得
http://www.icpdas.com/root/support/faq/isagraf_tc.html > 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.

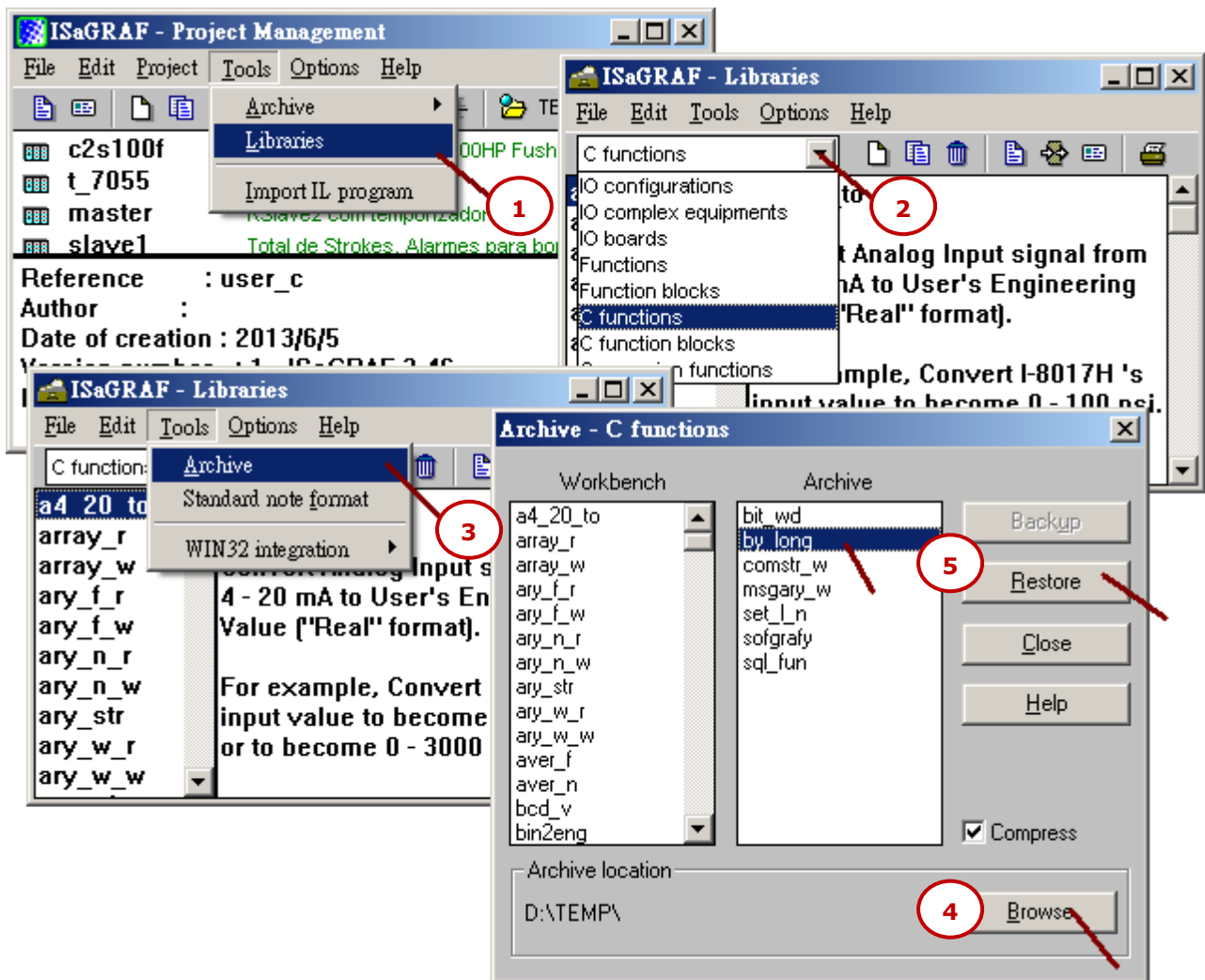
Classification	ISaGRAF Chinese FAQ-167						
Author	Chun Tsai	Version	1.3	Date	Feb. 2014	Page	3/31

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"

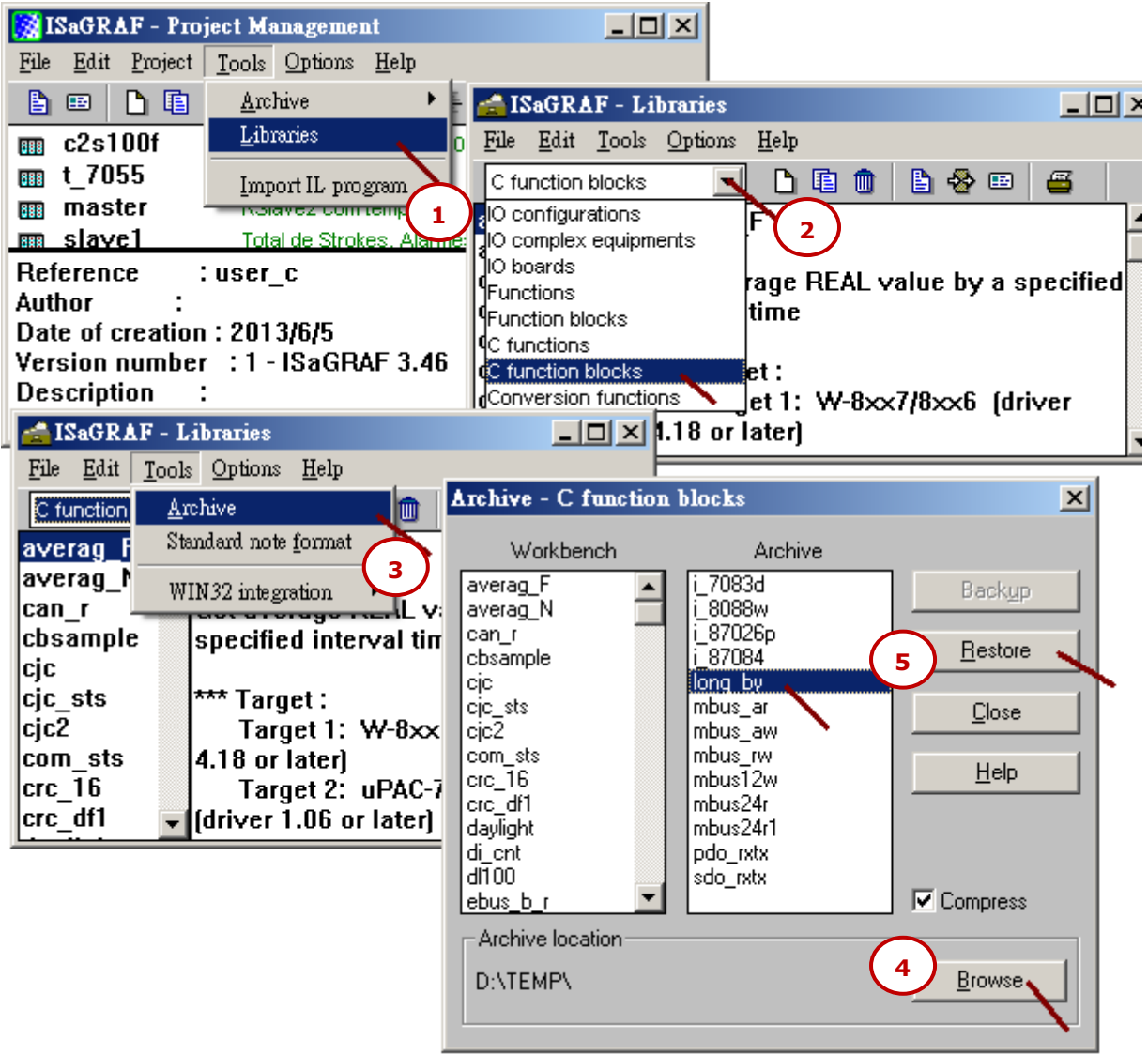
把 4 個 byte (值 0 ~ 255) 轉換組成一個 長整數 (32-bit signed integer).



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

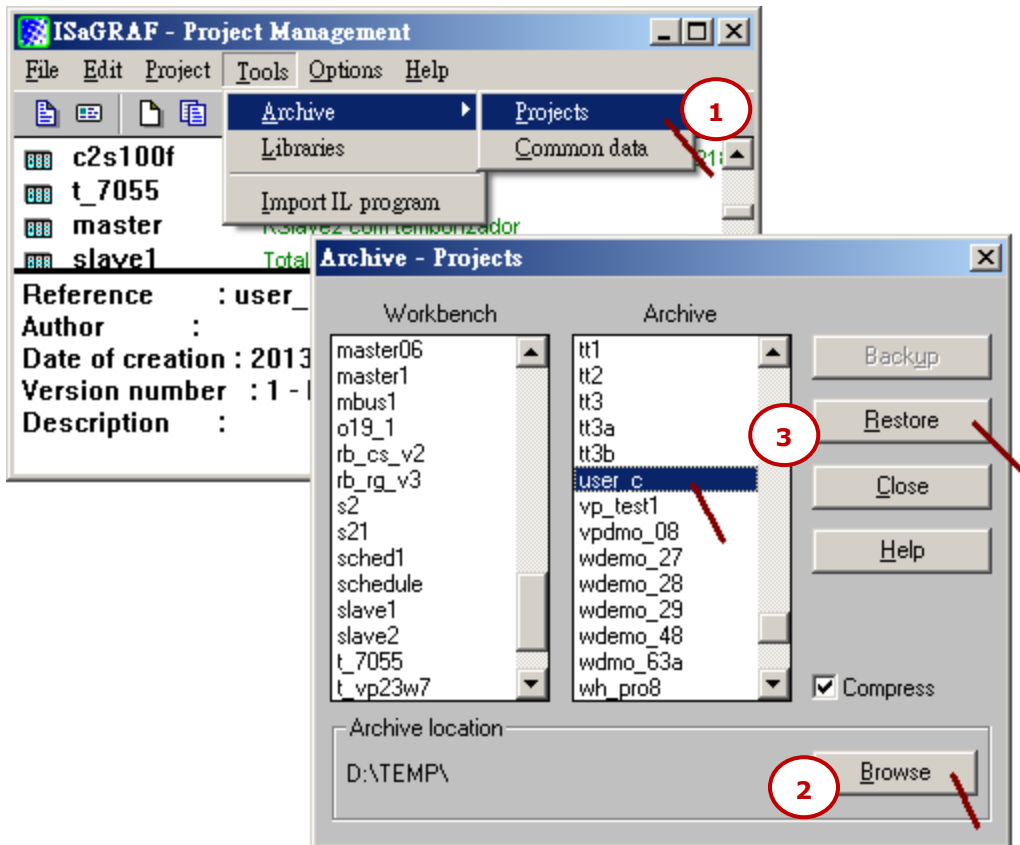
B. 回存 C-function block 範例 - “long_by”

用來將一個長整數 轉換分解成 4 個 byte.



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

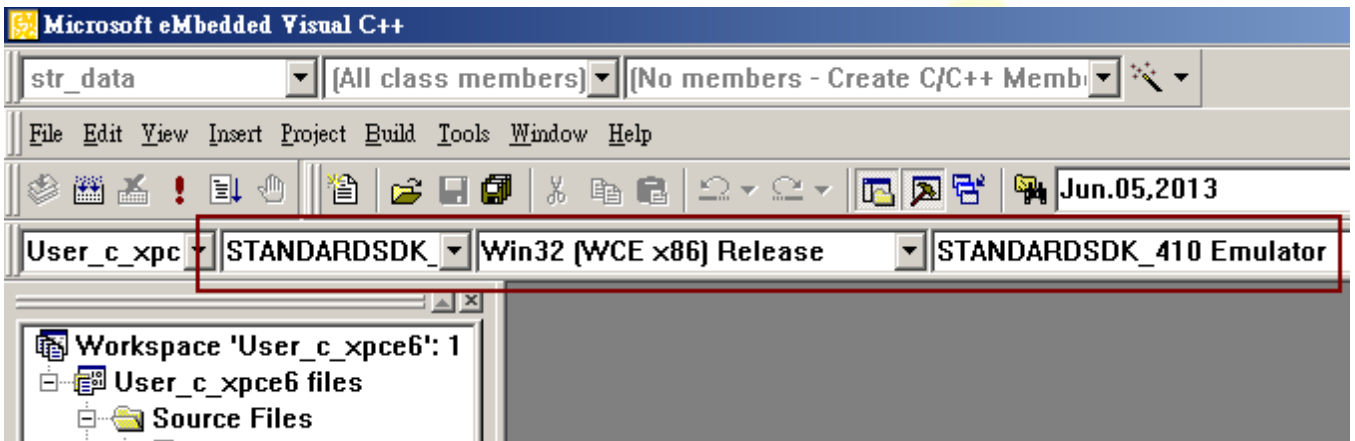
欲開發自定義的 C-function 或 C-function block 必須有以下的開發環境:

1. EVC++ 4.0 或
2. VS2008

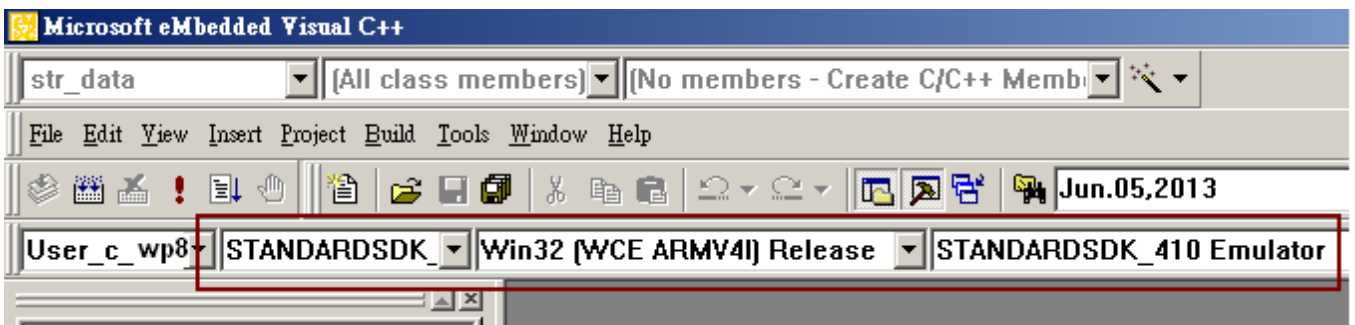
2.1. EVC 開發環境

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)

ftp://ftp.icpdas.com/pub/cd/xp-8000-ce6/sdk/platformsdk/pacsdk_ce_1.4.3_vs2008.msi

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

ftp://ftp.icpdas.com/pub/cd/winpac/napdos/wp-8x4x_ce50/sdk/platformsdk/pac270_sdk_20121015.msi

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。

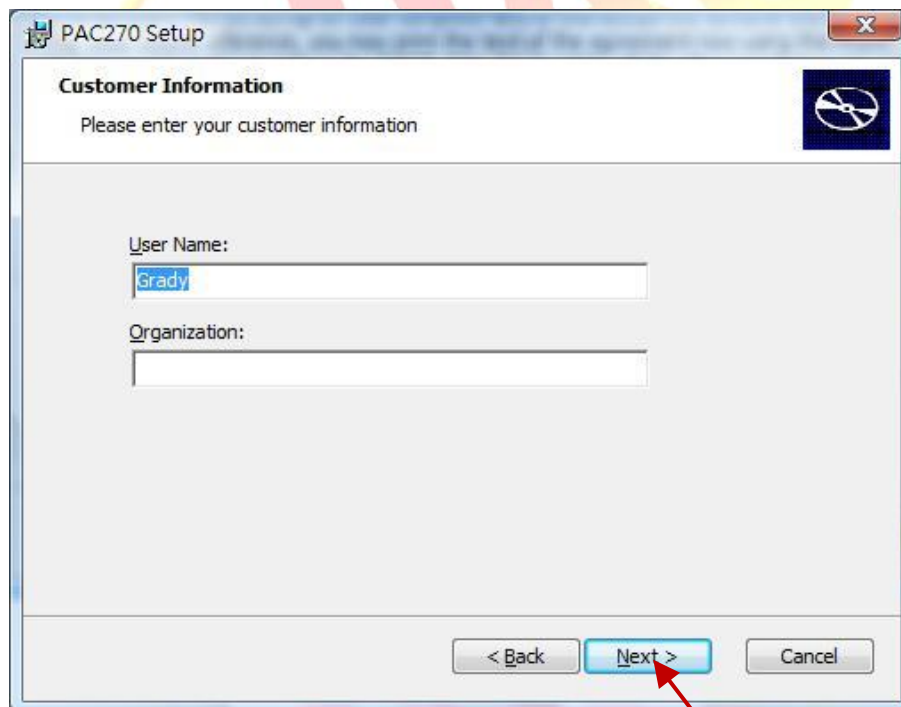


Classification	ISaGRAF Chinese FAQ-167						
Author	Chun Tsai	Version	1.3	Date	Feb. 2014	Page	8/31

3. 點選 “Accept”, 再點擊 “Next” 按鈕.

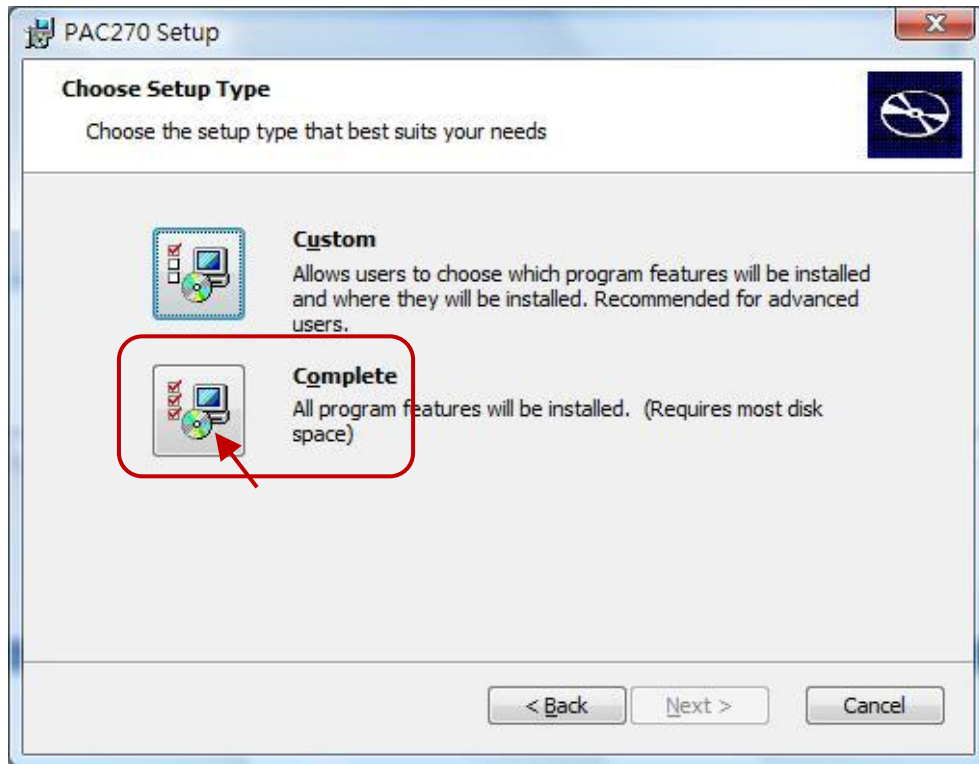


4. 點擊 “Next” 按鈕.

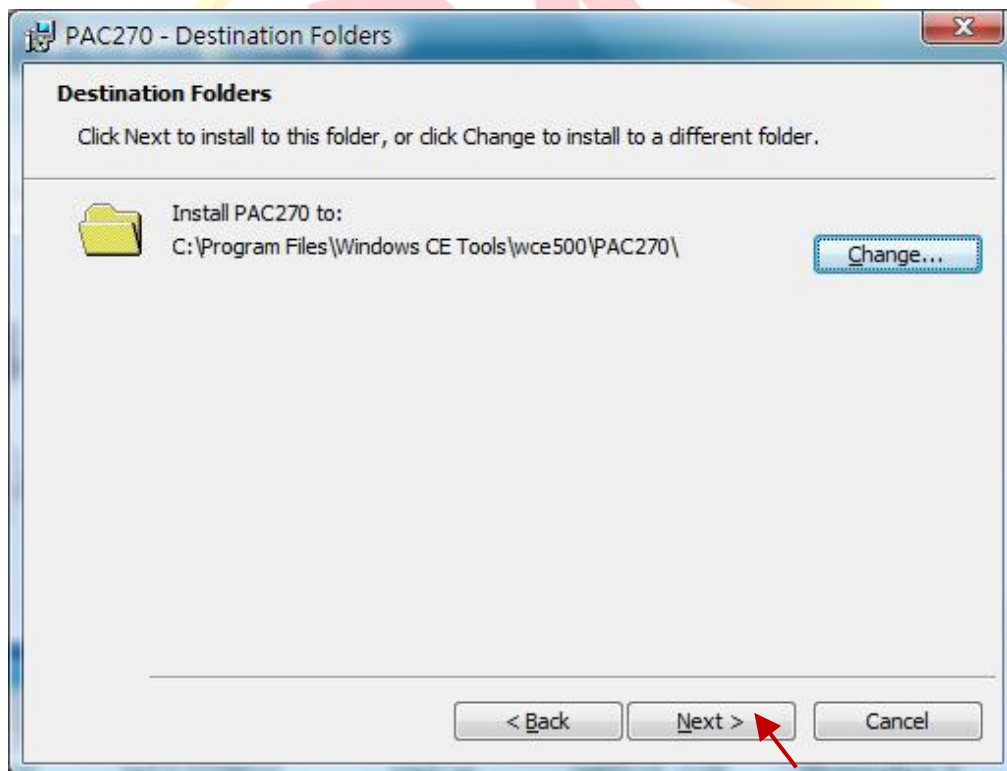


Classification	ISaGRAF Chinese FAQ-167						
Author	Chun Tsai	Version	1.3	Date	Feb. 2014	Page	9/31

5. 點擊 “Complete” 按鈕.

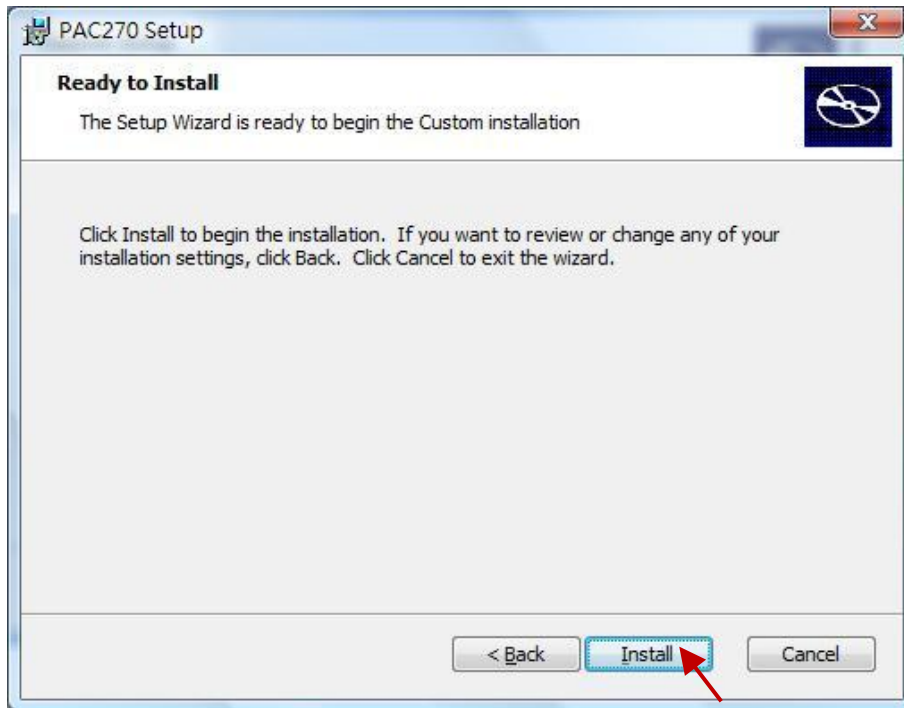


6. 點擊 “Next” 按鈕.

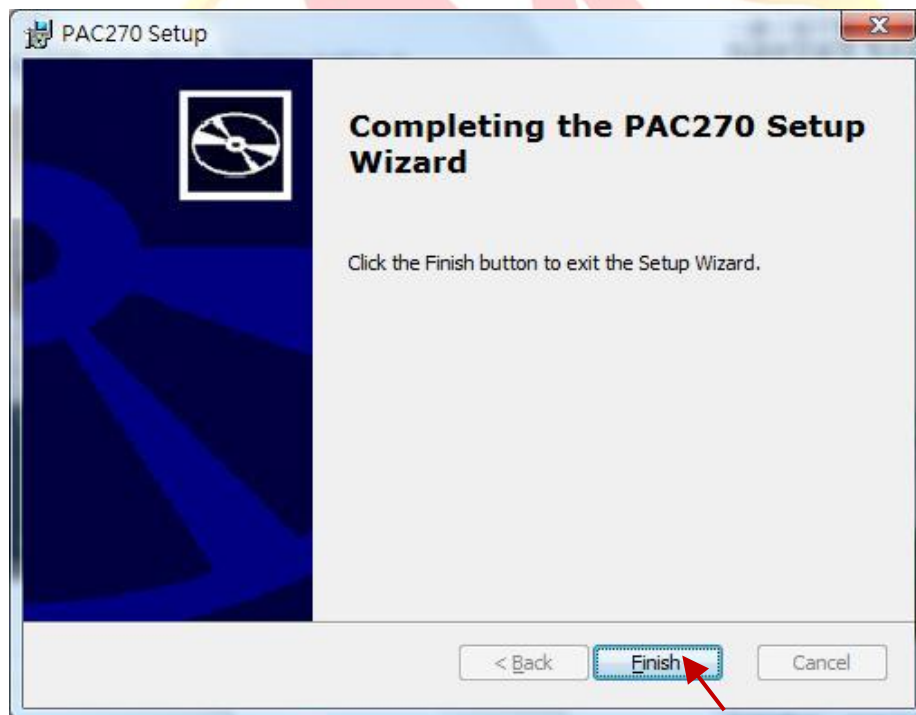


Classification	ISaGRAF Chinese FAQ-167						
Author	Chun Tsai	Version	1.3	Date	Feb. 2014	Page	10/31

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。

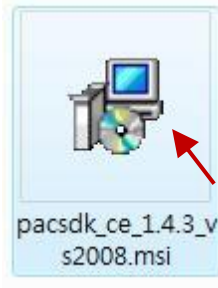


Classification	ISaGRAF Chinese FAQ-167						
Author	Chun Tsai	Version	1.3	Date	Feb. 2014	Page	11/31

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.

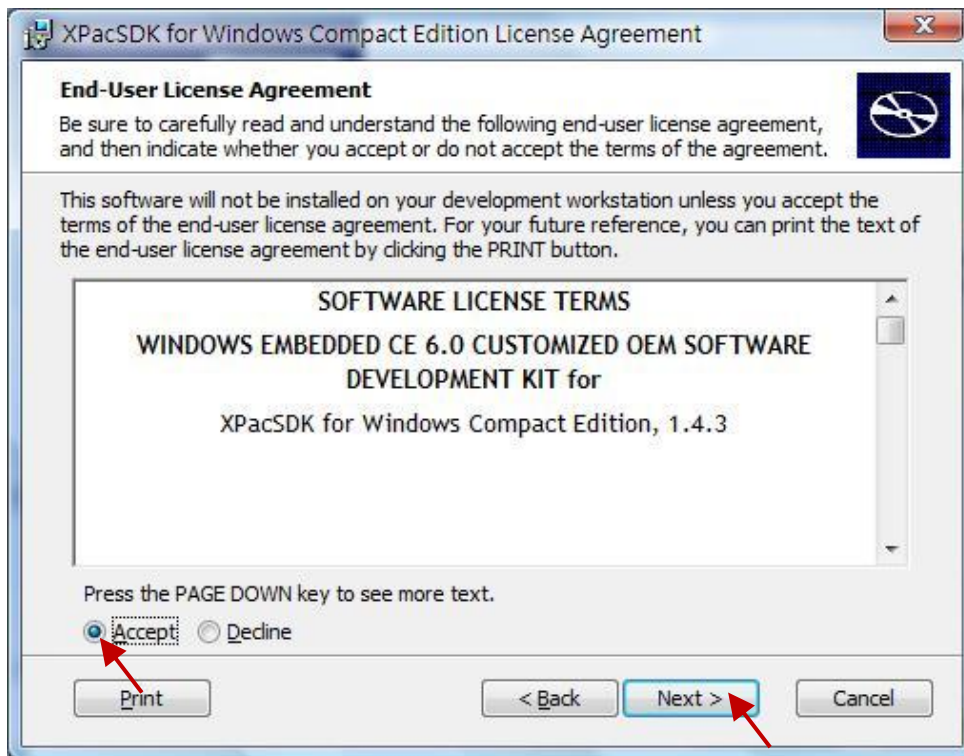


2. 點擊 “Next” 按鈕.

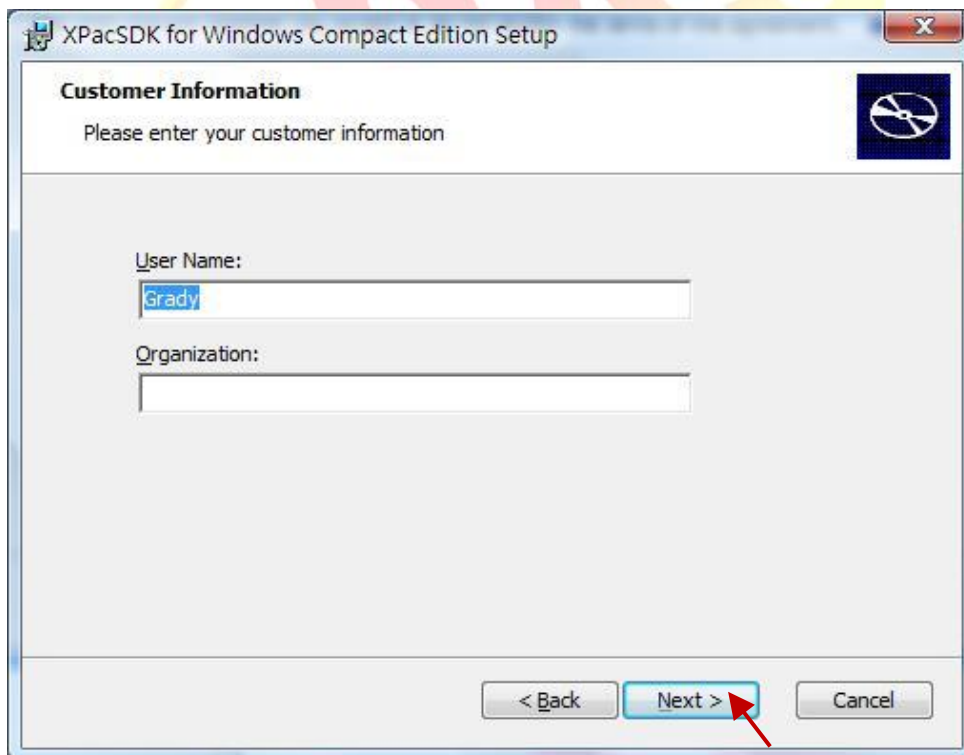


Classification	ISaGRAF Chinese FAQ-167						
Author	Chun Tsai	Version	1.3	Date	Feb. 2014	Page	12/31

3. 點選 “Accept”, 再點擊 “Next” 按鈕.

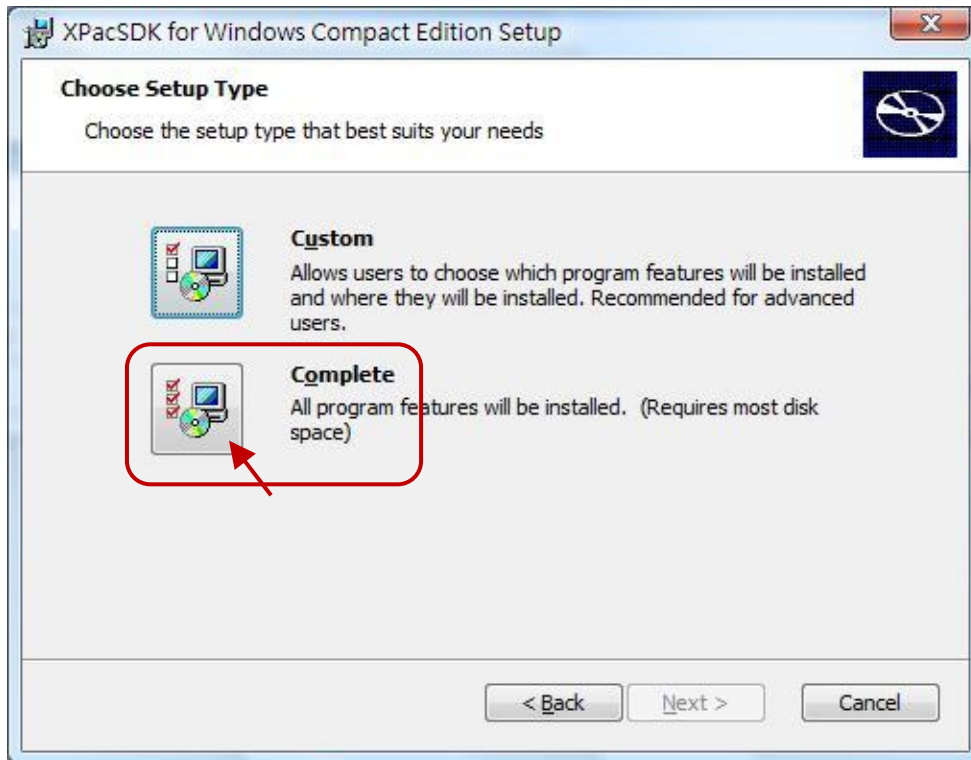


4. 點擊 “Next” 按鈕.



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

5. 點擊 “Complete” 按鈕.

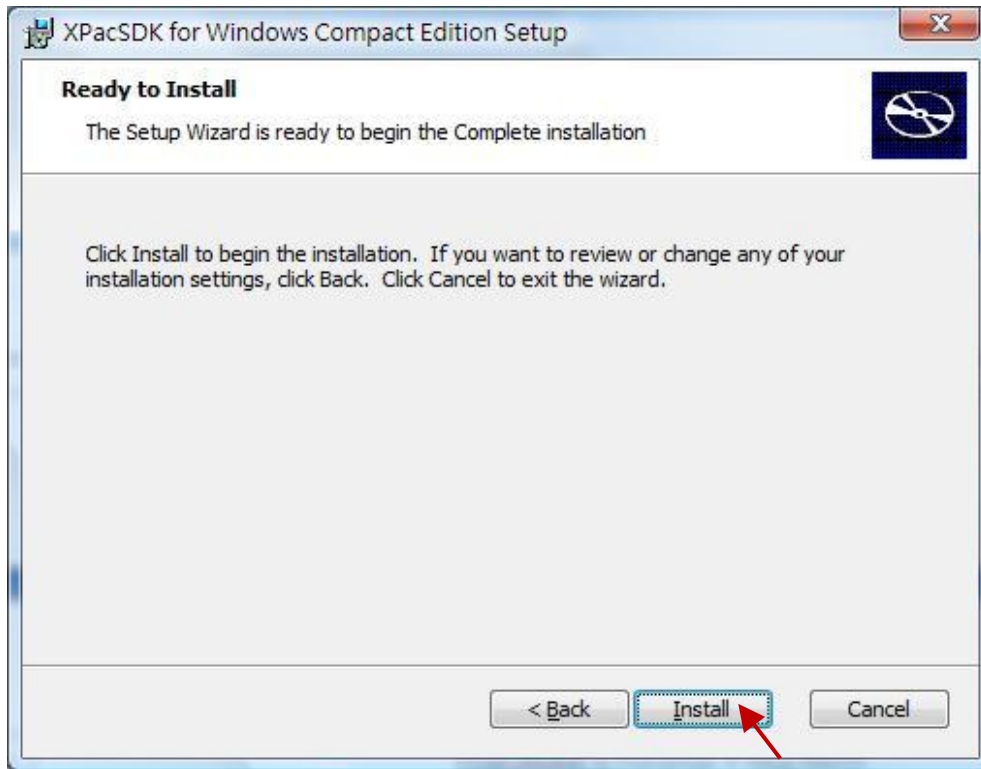


6. 點擊 “Next” 按鈕.

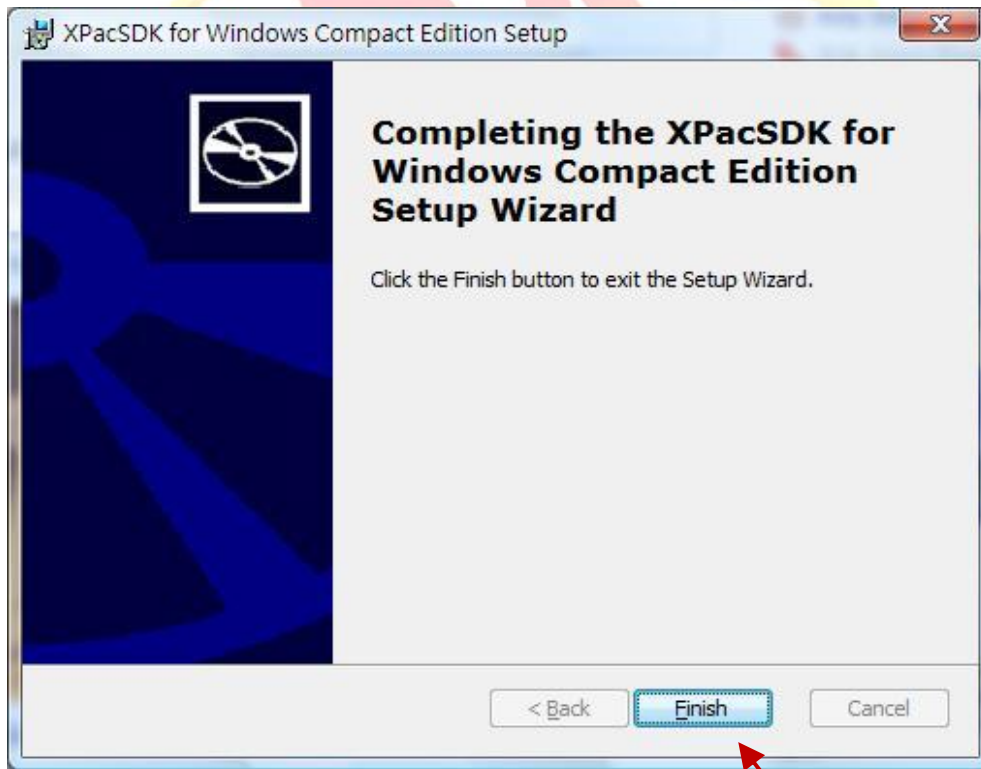


Classification	ISaGRAF Chinese FAQ-167						
Author	Chun Tsai	Version	1.3	Date	Feb. 2014	Page	14/31

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

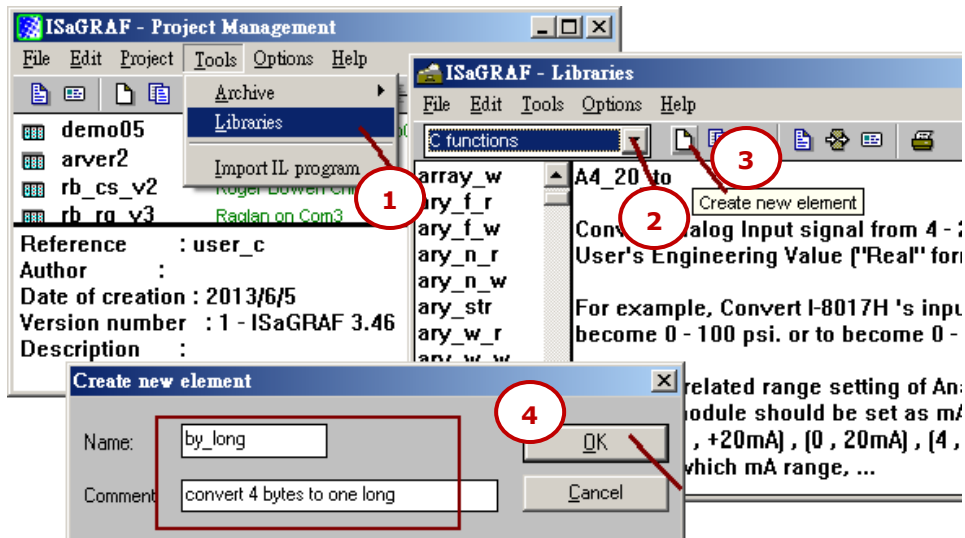
3.1. 定義 C-function Lib

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).

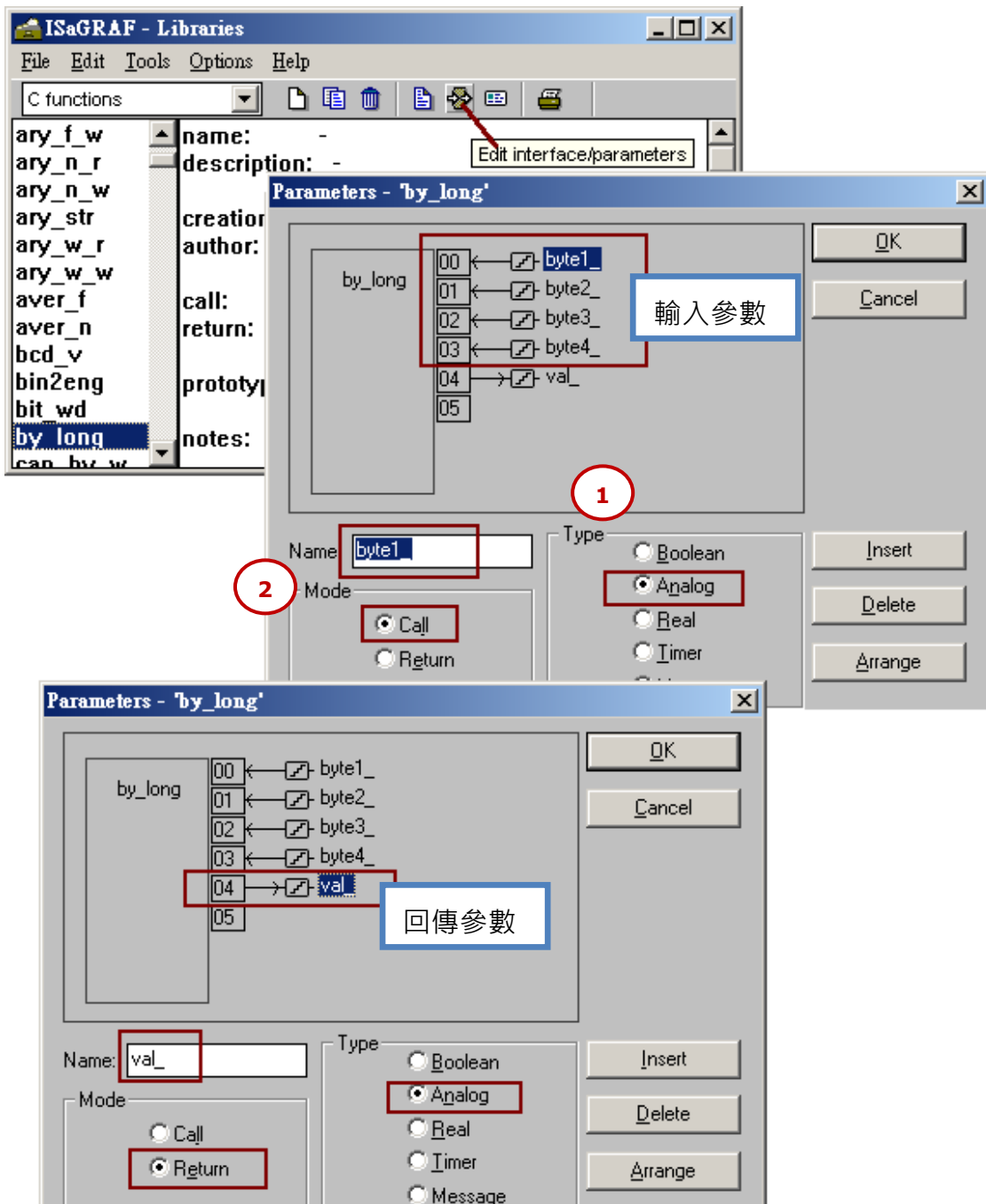
步驟如下:

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

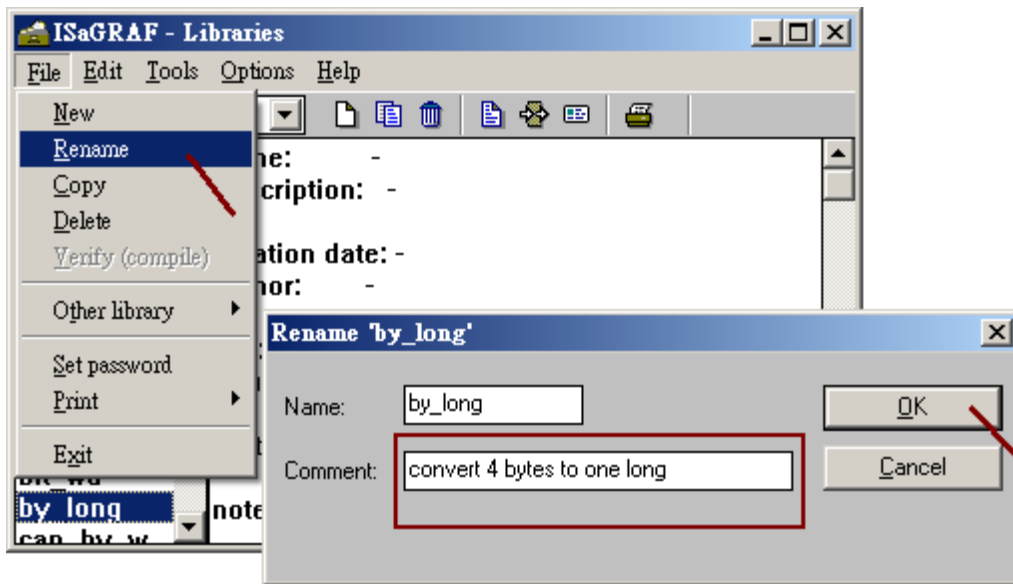


2. 定義 C-function 的 輸入 與 回傳 參數:

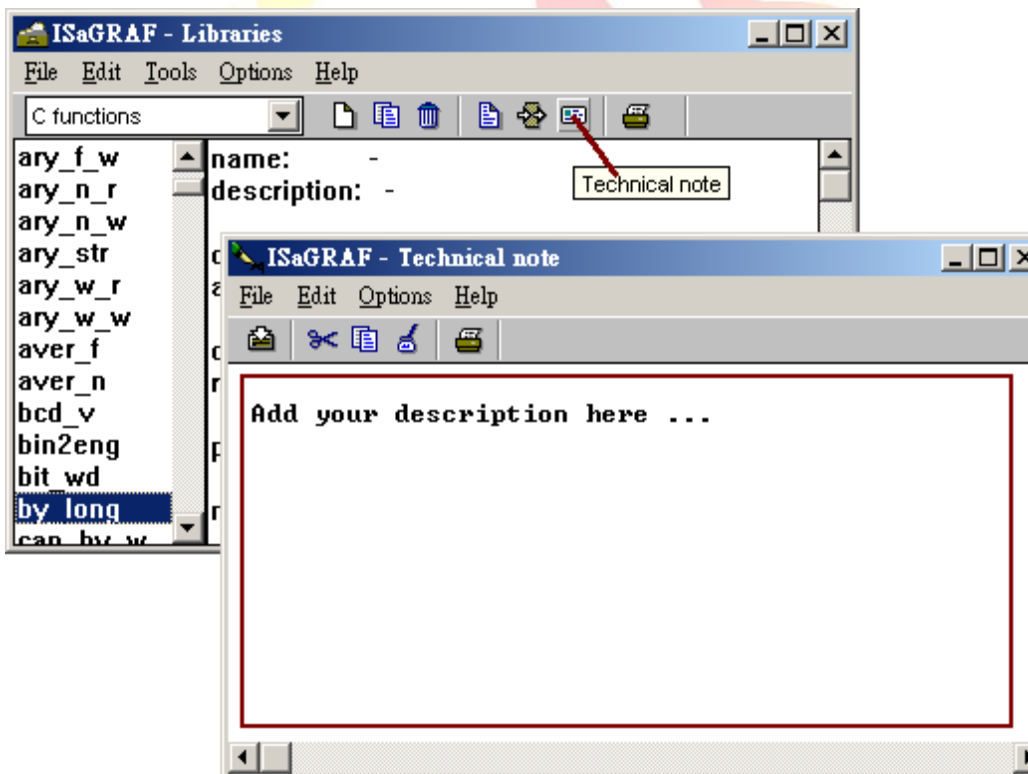
下圖顯示此 C-function 有 4 個輸入參數, 分別是 “byte1_”, “byte2_”, “byte3_” 與 “byte4_”. 它們的型態都是 “Analog” (表示 32-bit signed integer), 而 Mode 的設定為 “Call” (表示輸入參數). 請在每個參數名稱後方加一個字元 “_”. 主要是因為輸入 與 回傳參數會被 ISaGRAF 認定為保留字, 不可在程式內使用. 所以加個 “_” 會避開程式內一些用到保留字的機會. 此 C-function 的回傳參數為 “val_”, 而 Mode 設定是 “Return”, 型態是 “Analog”.



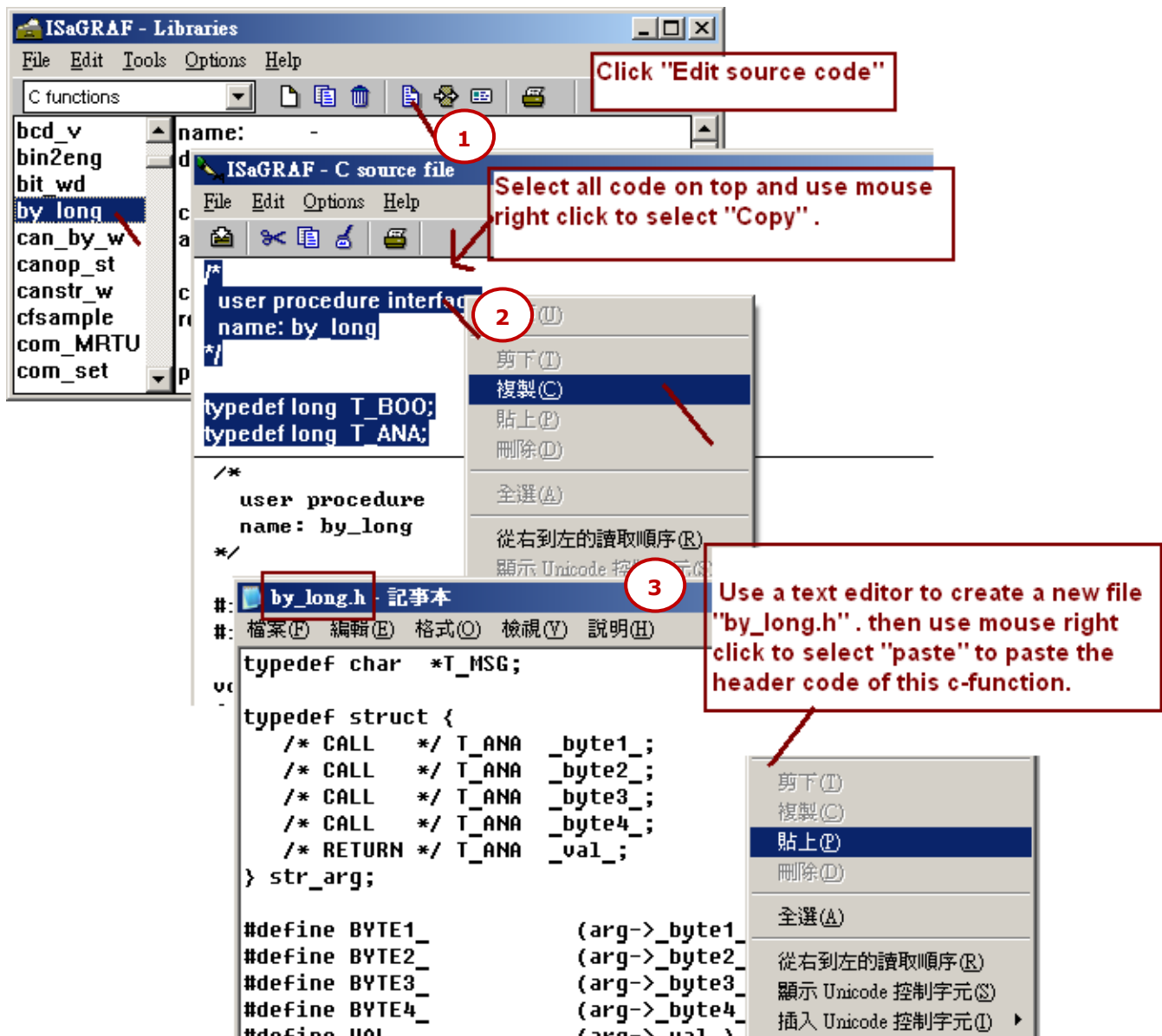
3. 為此 C-function 加上 註解 說明文字:



4. 為此 C-function 加上 技術 說明文字:



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".



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 shows two windows. The top window, titled "ISaGRAF - C source file", contains the following code:

```

/*
 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
<
>
USP us
<
sys_
retl
>

```

A red box highlights the code block from the first comment line to the end of the function definition. A red arrow points from this box to a callout box that says: "Select all code and use mouse right click to select 'Copy'". A red circle with the number "1" is next to the callout box. A context menu is open over the selected code, with "Copy Ctrl+C" highlighted.

The bottom window, titled "by_long.c - 記事本", shows the code pasted into a text editor. A red box highlights the title bar of this window. A red arrow points from this box to a callout box that says: "Then use mouse right click to paste it to a text editor and name it as 'by_long.c'". A red circle with the number "2" is next to the callout box. A context menu is open over the text editor, with "貼上(P)" (Paste) highlighted.

```

/*
 user procedure
 name: by_long
*/

#include <tasy0def.h>
#include <grus0206.h>

void USP_by_long (str_arg *arg)
{
}

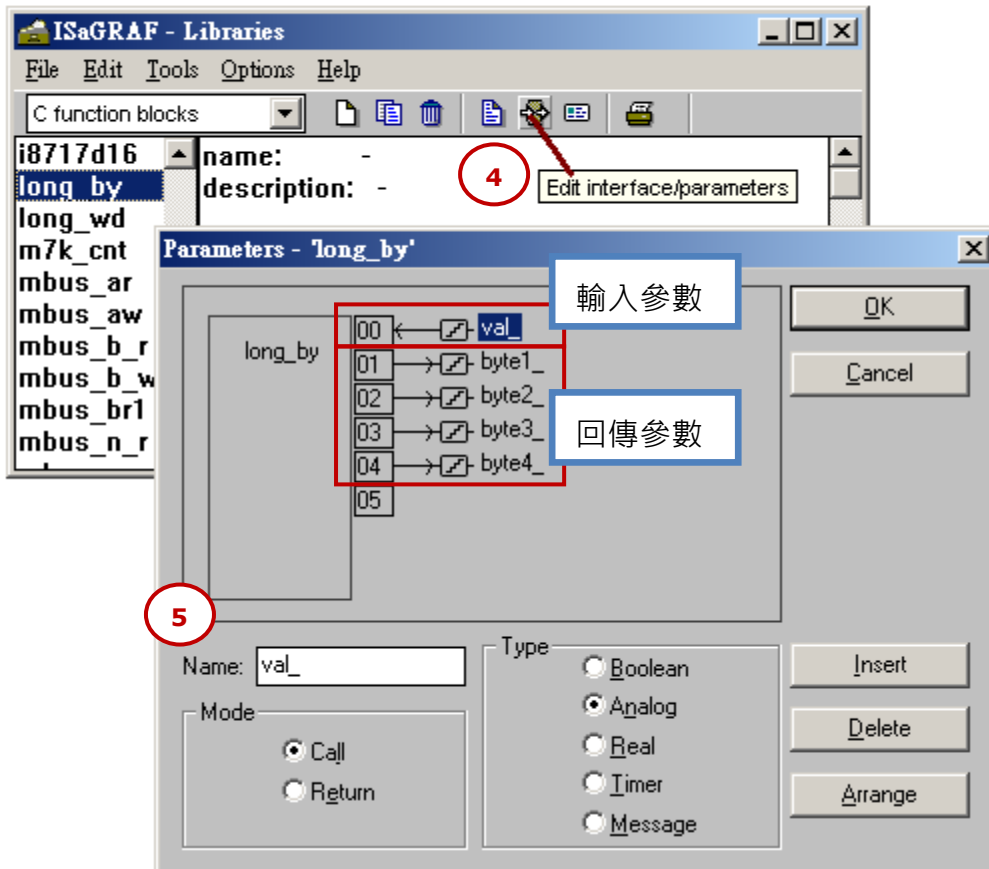
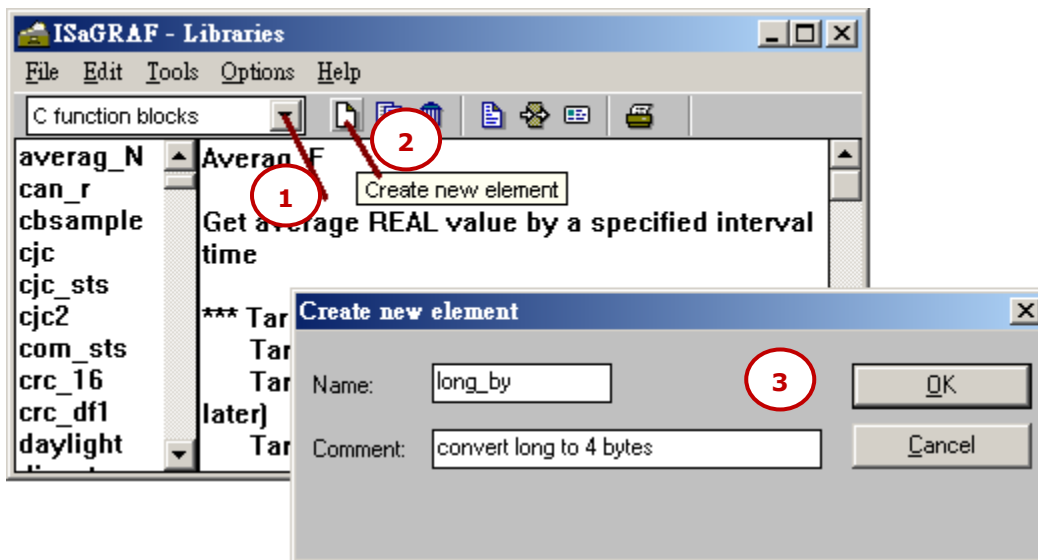
USP uspdev_by_long (char *name)
{
 sys_strcpy (name, "BY_LONG");
}

```

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
/*
function block interface
name: long_by
*/
typedef long T_B00;
typedef long T_ANA;
typedef float T_REAL;
typedef long T_TMR;
typedef c

long_by.c - 記事本
typedef s
/* CAL
*/
} str_arg
#define u

/*
function block
name: long_by
*/
#include <tasy0def.h>
#include <grfb0162.h>

typedef
struct {

} str_data;

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

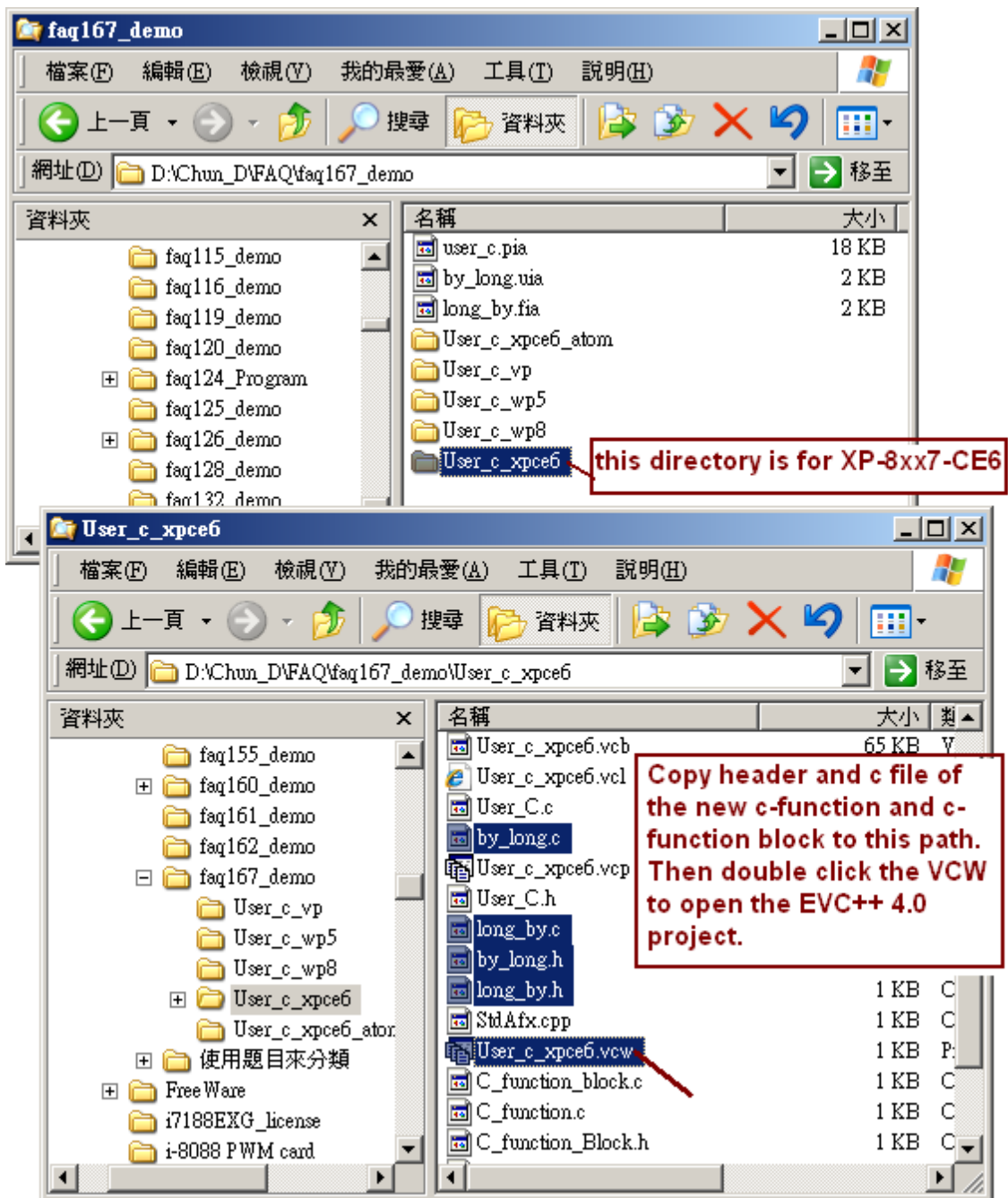
void FBACT_long_by (uint16 hinstance, str_data *data, str_
{
}
第 40 列, 第 1 行

```

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：

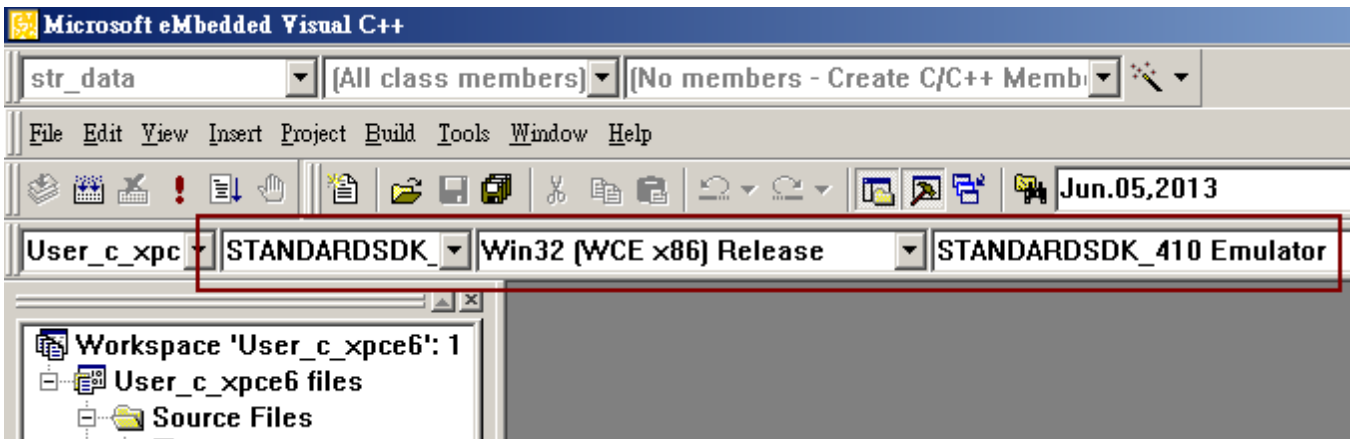
1. 先把 C-function 與 C-function block 的 header 與 C 範本檔案，複製到你的 PC 內的相對應的專案資料夾內，然後用 Mouse 雙擊該資料夾內的 VCW 檔案來開啟這個 EVC++ 4.0 專案 (此處是以 XP-8xx7-CE6 當例子)



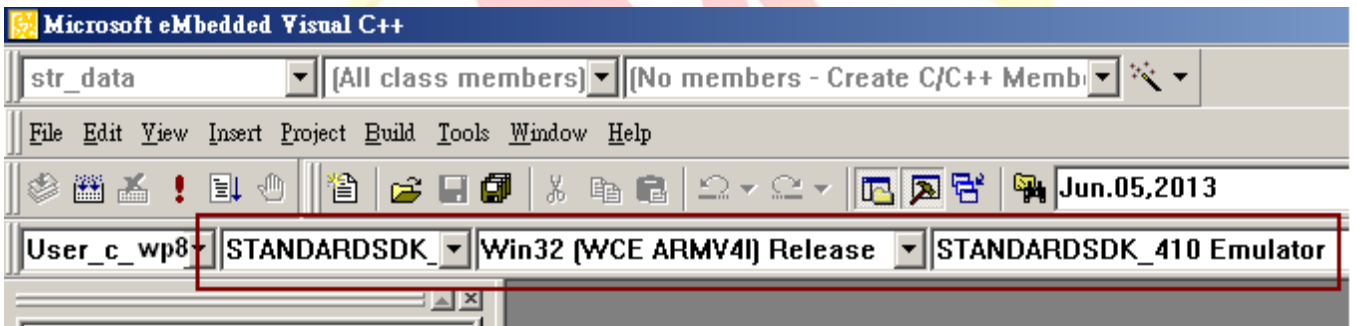
Classification	ISaGRAF Chinese FAQ-167						
Author	Chun Tsai	Version	1.3	Date	Feb. 2014	Page	23/31

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” .

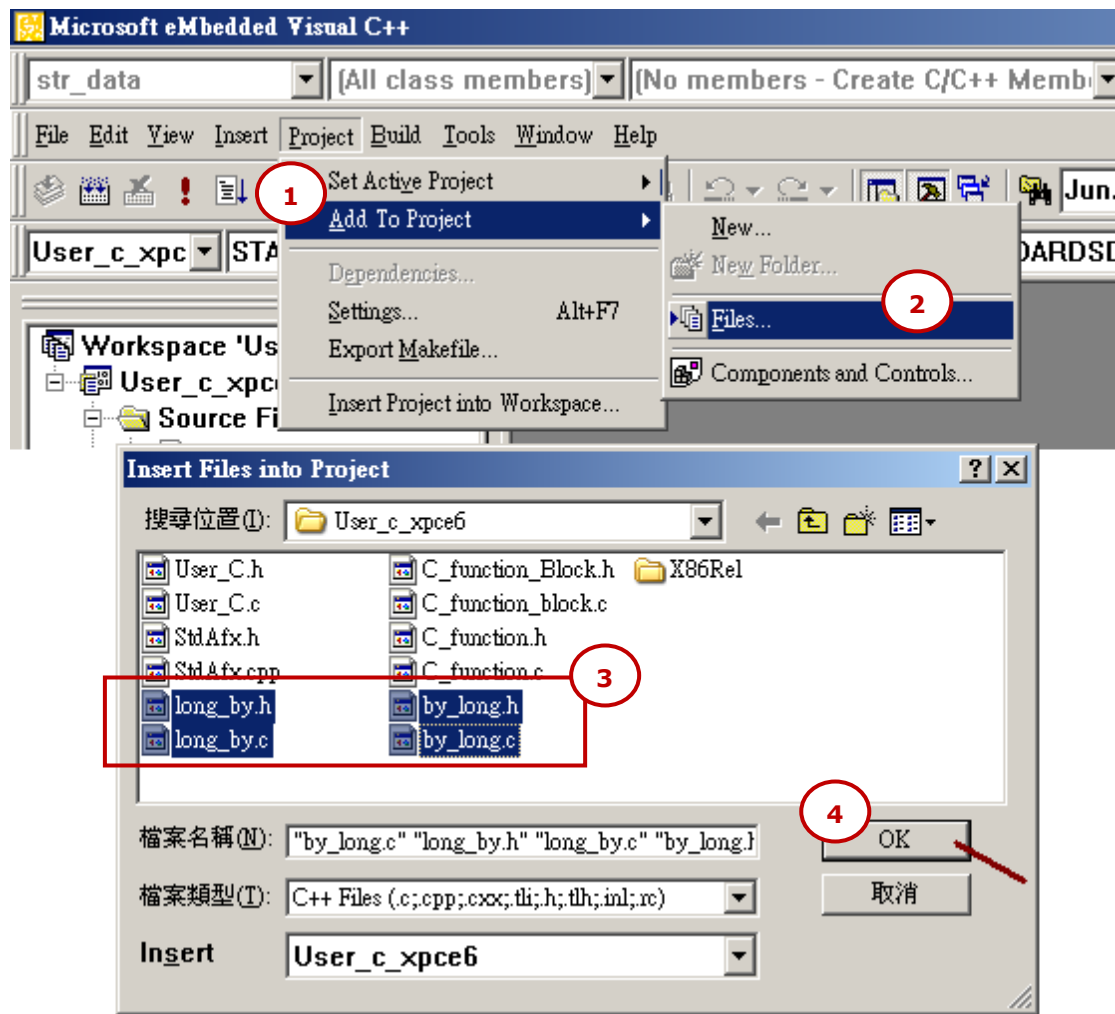


下圖是 WP-8xx7 與 WP-5147 與 VP-25W7 與 VP-4137 需使用的設定 : 必需是
 “STANDARDSDK” 與 “Win32 (WCE **ARMV4I**) Release” 與 “STANDARDSDK_4xx Emulator” .



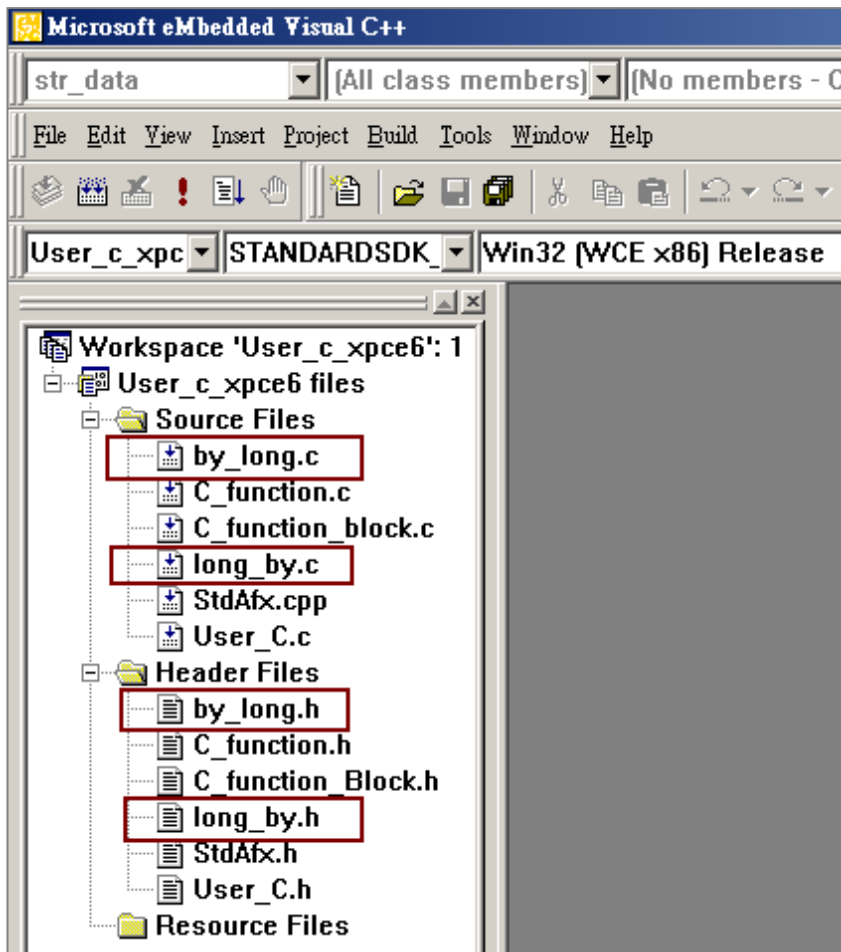
Classification	ISaGRAF Chinese FAQ-167						
Author	Chun Tsai	Version	1.3	Date	Feb. 2014	Page	24/31

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

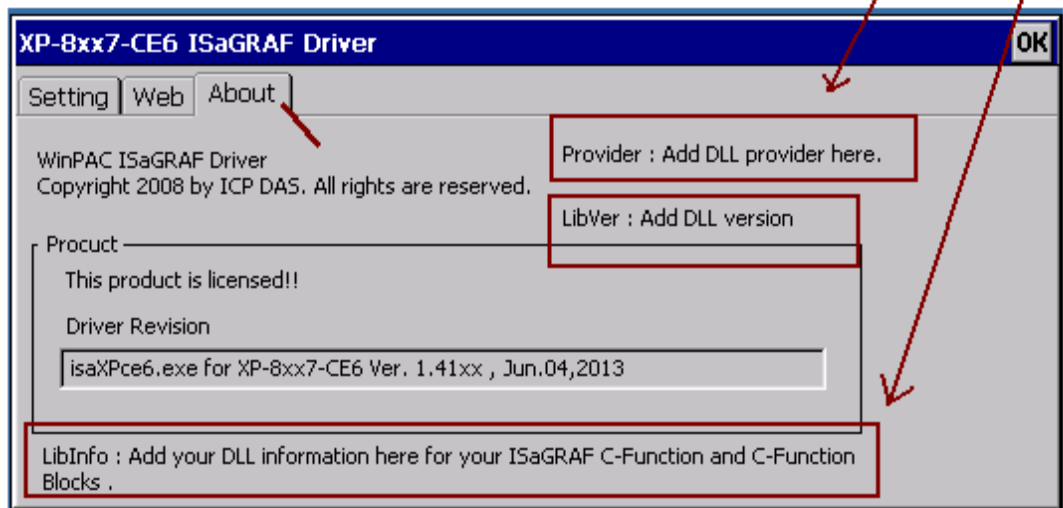
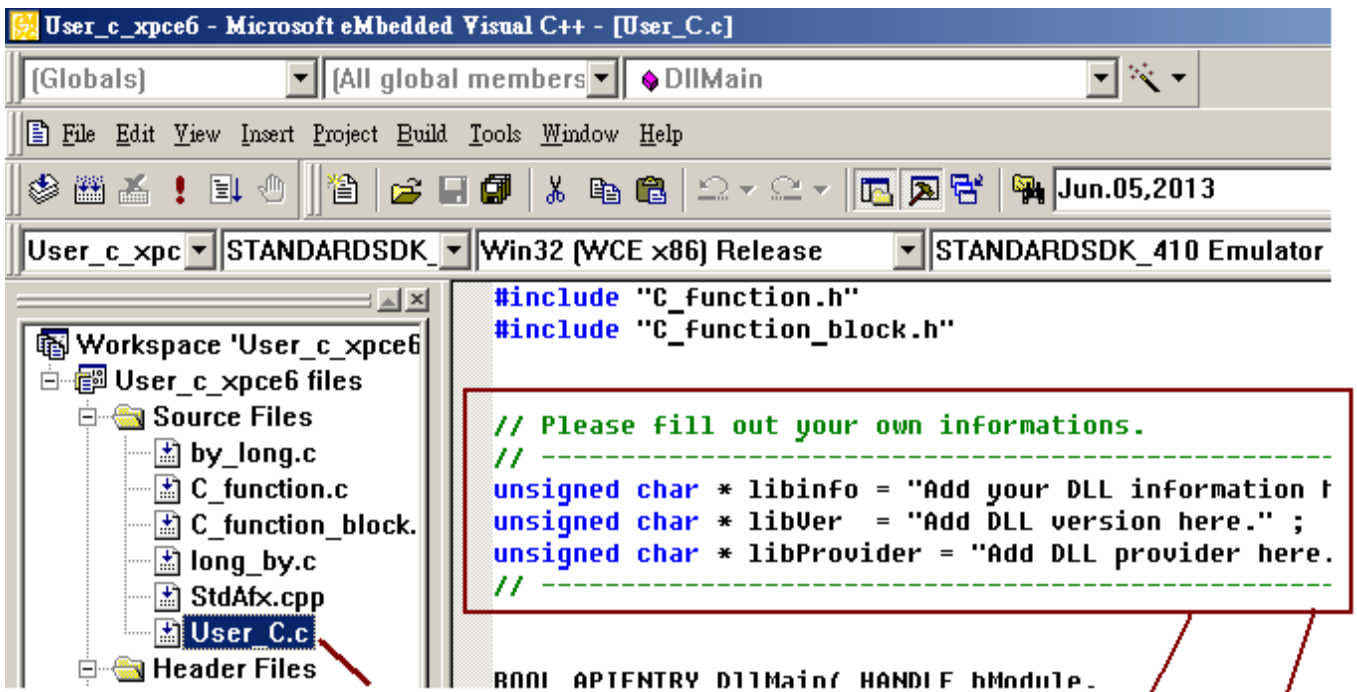
之後會顯示類似下方的 Project.



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

4.1. 編寫 “User_C.c”

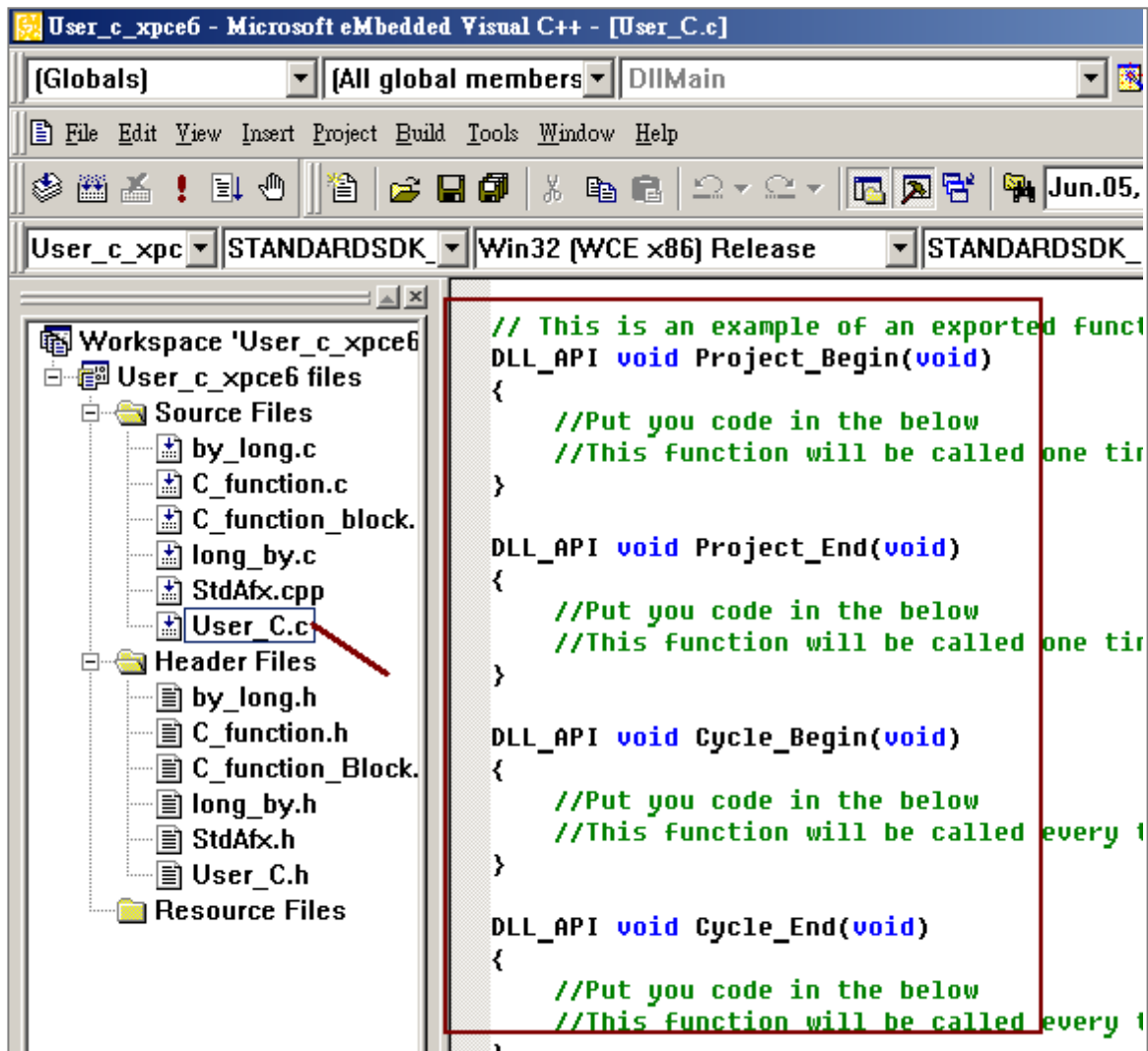
請把下圖內 “User_C.c” 內的 “libinfo” 與 “libVer” 與 “libProvider” 所定義的資訊，修改成你想要顯示的資訊。這 3 個資訊會顯示在控制器的 ISaGRAF 驅動畫面的 “About” 視窗上。



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。



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 screenshot shows the ISaGRAF IDE with the project 'User_c_xpc' and configuration 'STANDARDSDK Win32 [WCE x86] Release'. The workspace contains source files and header files. The file 'by_long.c' is selected and its code is displayed in the editor. Red boxes and arrows highlight key parts of the code and provide instructions:

- File Explorer:** A red circle with the number '1' highlights the 'by_long.c' file in the Source Files folder.
- Code Editor:**
 - Comments: `/* user procedure name: by_long */` are highlighted with a red box and an arrow pointing to a callout: "Add 'User_c.h' Add 'by_long.h'".
 - Includes: `#include "User_C.h"` and `#include "by_long.h"` are highlighted with a red box and an arrow pointing to a callout: "Add your logic for this c-function."
 - Function Definition: The `void USP_by_long (str_arg *arg)` function is shown with its logic:


```

          long ret = 0;

          ret |= (BYTE4_ & 0xFF) << 24;
          ret |= (BYTE3_ & 0xFF) << 16;
          ret |= (BYTE2_ & 0xFF) << 8;
          ret |= (BYTE1_ & 0xFF) << 0;

          VAL_ = ret;
          
```
 - Function Pointer Definition: The `UFP uspdef_by_long (char *name)` function is shown with its logic:


```

          sys_strcpy (name, "BY_LONG");
          return ((UFP)USP_by_long);
          
```

The screenshot shows the ISaGRAF IDE with the same project and configuration. The file 'C_function.c' is selected and its code is displayed in the editor. Red boxes and arrows highlight key parts of the code and provide instructions:

- File Explorer:** A red circle with the number '2' highlights the 'C_function.c' file in the Source Files folder.
- Code Editor:**
 - Includes: `#include "stdafx.h"`, `#include "User_C.h"`, and `#include "C_function.h"` are highlighted with a red box and an arrow pointing to a callout: "Add your logic for this c-function."
 - Function Pointer Definition: The `UFP uspdef_by_long (char *name);` declaration is highlighted with a red box and an arrow pointing to a callout: "Add 'User_c.h' Add 'by_long.h'".
 - Function Pointer List: The `UFP_LIST UFPBUF[] =` list is shown with the entry:


```

          {
            uspdef_by_long,
            NULL,
          };
          
```

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:

- Red box: "Add 'User_C.h' Add 'long_by.h'" with arrows pointing to the #include lines.
- Red box: "Each c-function block instance in the ISaGRAF has its own memory here." with an arrow pointing to the unsigned char AO[4]; line.
- Red box: "add logic for this c-function block here." with an arrow pointing to the FBACT_long_by function body.
- Red box: "Add code to get return parameters of the c-function block." with an arrow pointing to the FBREAD_long_by function body.

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

```

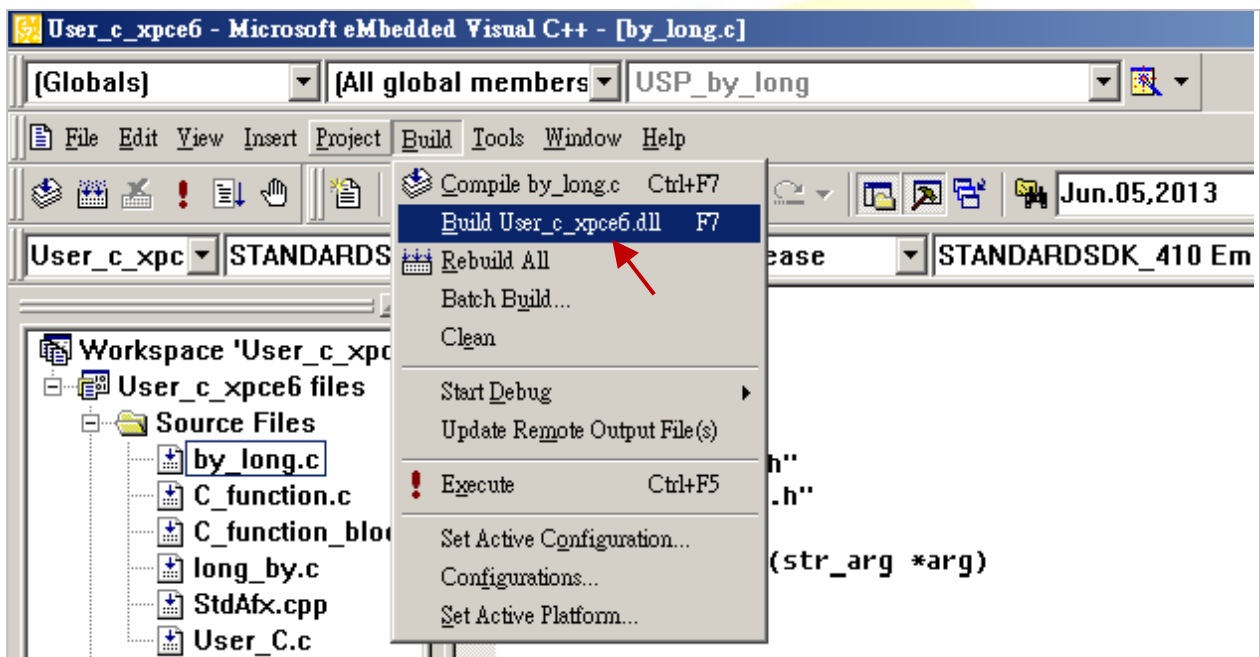
#include "stdafx.h"
#include "User_C.h"
#include "C_function_block.h"

ABP fbldf_long_by (char *name, IBP *initproc, RBP *readproc);

FBL_LIST FBLBUF[] =
{
    fbldf_long_by,
    NULL,
};

```

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