

Classification	WinPAC SDK FAQ				No.	6-010-00	
Author	Sean Hsu	Version	1.0.1	Date	2012/10/18	Page	1/16

如何將WinPacSDK函式庫更新為PACSDK函式庫

適用於:

平台	OS 版本	WinPAC utility 版本
WP-8x3x	所有版本	所有版本
WP-8x4x	所有版本	所有版本
VP-25W1	所有版本	所有版本
VP-23W1	所有版本	所有版本
WP-5000 系列	所有版本	所有版本

將WinPacSDK函式庫更新為PACSDK函式庫之前，首先需於PC安裝WinPAC平台開發SDK的安裝套件。請確認提供PACSDK函式庫的SDK版本安裝套件，其發佈日期需晚於或為2012/10/15。例如，PAC270_SDK_20121015.msi

最新版本的平台開發 SDK 安裝套件，其出貨 CD 及 FTP 下載路徑為
CD:

CD:\Napdos\wp-8x4x_ce50\SDK\

FTP:

http://ftp.icpdas.com/pub/cd/winpac/napdos/wp-8x4x_ce50/sdk/

檔名: pac270_sdk_YYYYMMDD.msi (YYYYMMDD : SDK 發佈日期)

安裝步驟為:

步驟 1: 執行“PAC270_SDK_YYYYMMDD.msi”

步驟 2: 依著設定畫面顯示按下 Next 鍵，直到 PAC270_SDK_YYYYMMDD.msi 安裝程序完成。



Classification	WinPAC SDK FAQ				No.	6-010-00	
Author	Sean Hsu	Version	1.0.1	Date	2012/10/18	Page	2/16

我們提供 2 種方法將 WinPacSDK 函式庫更新為 PACSDK 函式庫。
 第一種方式是提供詳細的更新步驟提供給使用者手動更新，而另一種是提供更新工具自動更新 PACSDK 函式庫。

手動更新 SDK

eVC 程式

依以下步驟更新 PACSDK 函式庫至你的程式

步驟 1: 更改 WinPacSDK.h 為 PACSDK.h

1-1 開啟你的 WinPAC 系列模組使用的 **Microsoft Embedded Visual C++** 專案(*.vcw)

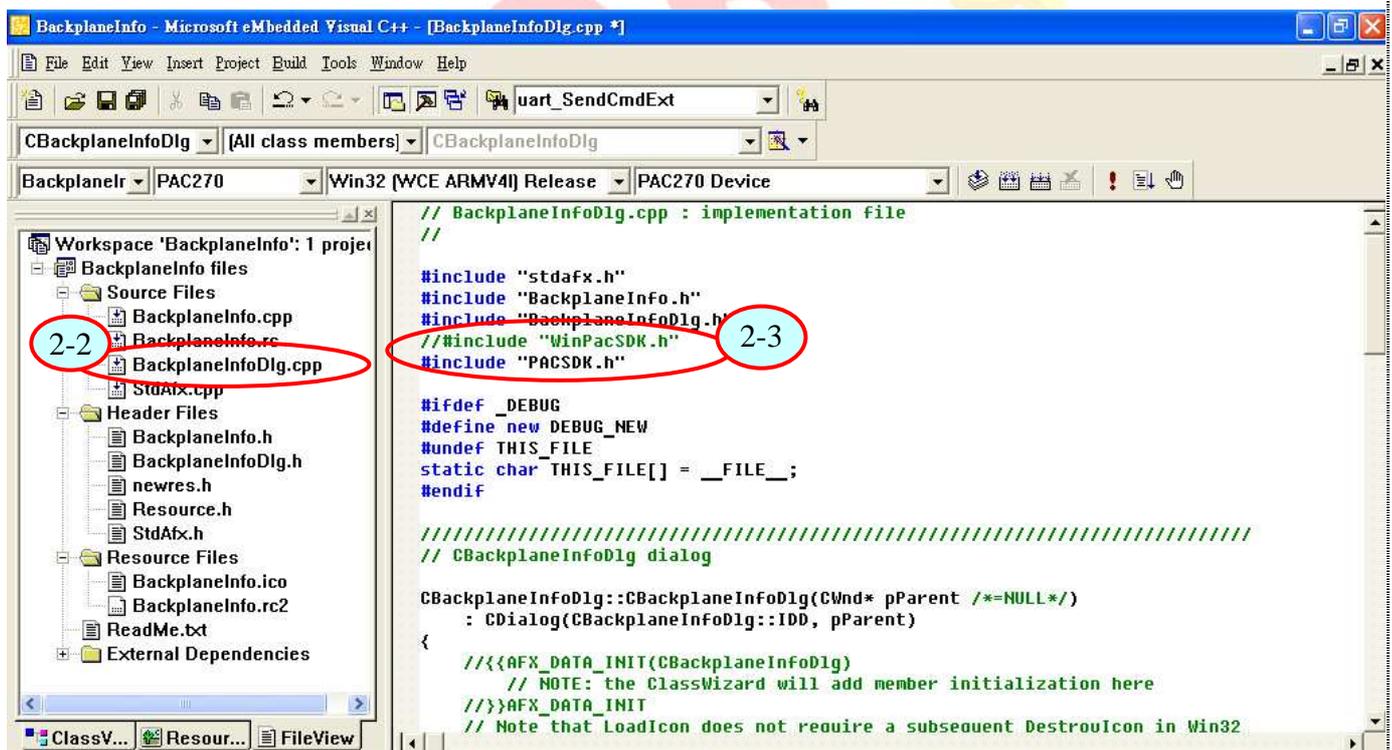
1-2 於 FileView 視窗上點選要修改的*.cpp 檔將此檔開啟

1-3 於 Editor 視窗將程式修改成以下的內容 (參考下圖)

```

// #include "WinPacSDK.h"
#include "PACSDK.h"

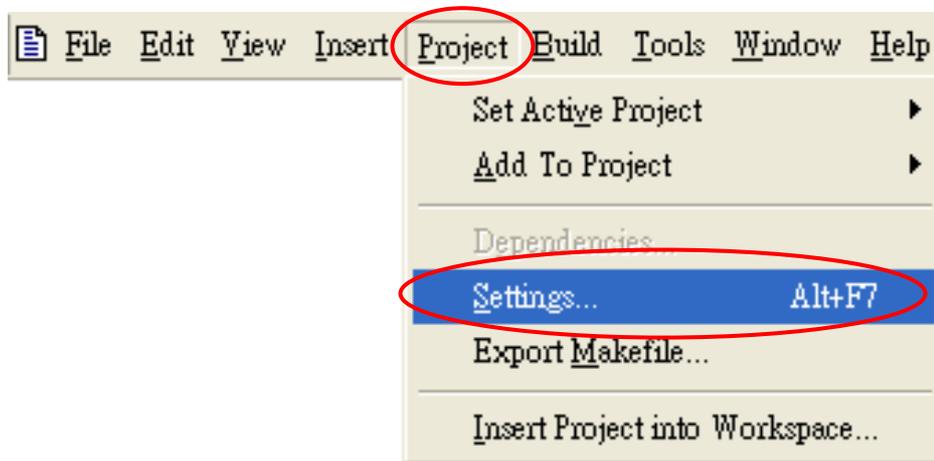
```



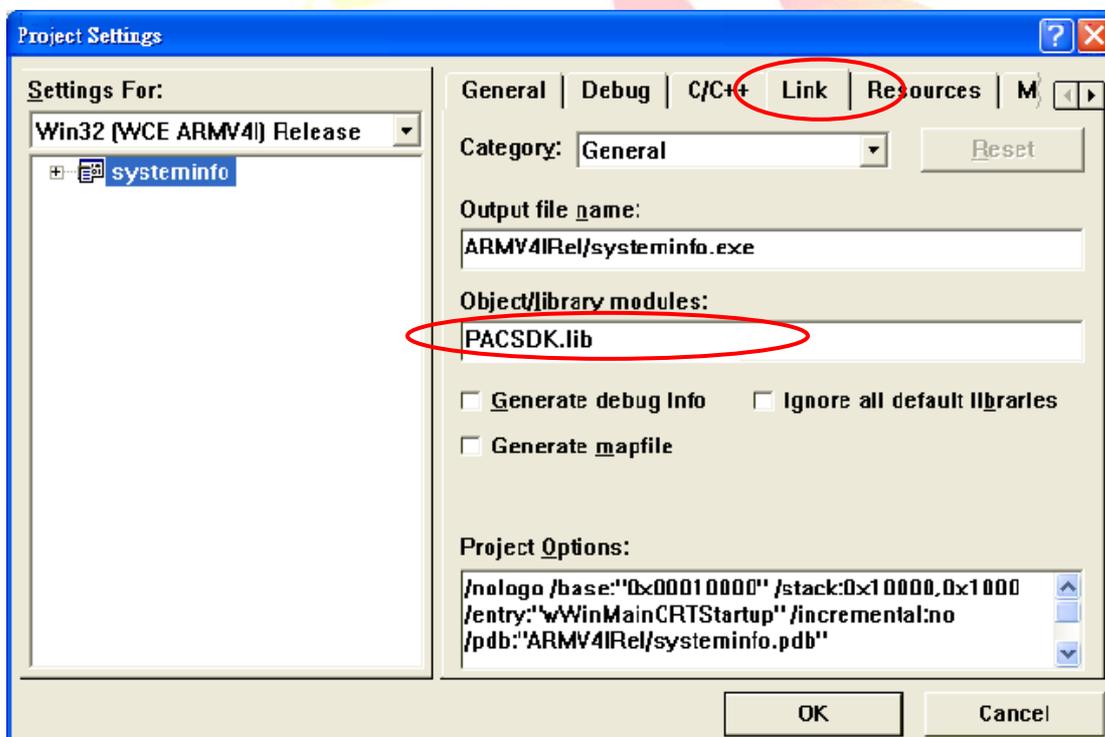
Classification	WinPAC SDK FAQ				No.	6-010-00	
Author	Sean Hsu	Version	1.0.1	Date	2012/10/18	Page	3/16

步驟 2: 更改 WinPacSDK.lib 為 PACSDK.lib

2-1 於 Project 選單，點選“Settings...”項目 (如下圖)



2-2 選擇 Project Settings 對話盒中的 Link 標籤，於“Object/library modules:”文字框內輸入 PACSDK.lib



Classification	WinPAC SDK FAQ				No.	6-010-00	
Author	Sean Hsu	Version	1.0.1	Date	2012/10/18	Page	4/16

步驟 3: 參考以下說明並確認你程式碼是否有以下的項目，如果有請修改你的程式碼
錯誤碼修改

```
//if(pac_GetLastError() == PAC_ERR_EEP_ACCESS_RESTRICTION)
```

修改為

```
if(pac_GetLastError() == PAC_ERR_EEP_INVALID_ADDRESS)
{
    //...
}
```

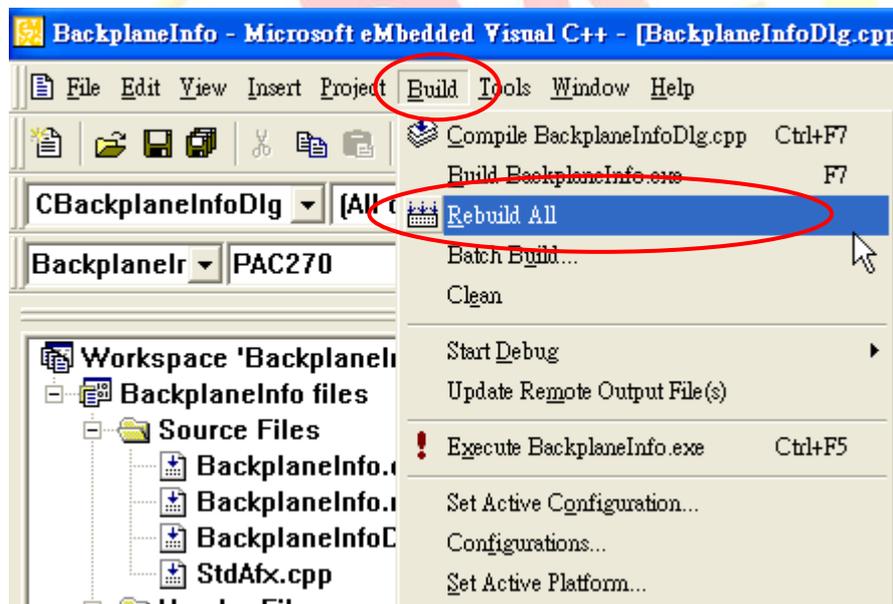
```
//if(pac_GetLastError() == PAC_ERR_SRAM_INVALID_TYPE)
```

修改為

```
if(pac_GetLastError() == PAC_ERR_MEMORY_INVALID_TYPE)
{
    //...
}
```

`PAC_ERR_EEP_ACCESS_RESTRICTION` 及 `PAC_ERR_SRAM_INVALID_TYPE` 兩個錯誤碼是定義在 `WinPacSDK.h`，使用 `PACSDK.h`，需將它們修改為 `PAC_ERR_EEP_INVALID_ADDRESS` 及 `PAC_ERR_MEMORY_INVALID_TYPE`。

步驟 4: 在 Embedded Visual C++ IDE 視窗上，點選 Build 選單的“Rebuild All”項目，重新編譯你的專案。



Classification	WinPAC SDK FAQ				No.	6-010-00	
Author	Sean Hsu	Version	1.0.1	Date	2012/10/18	Page	5/16

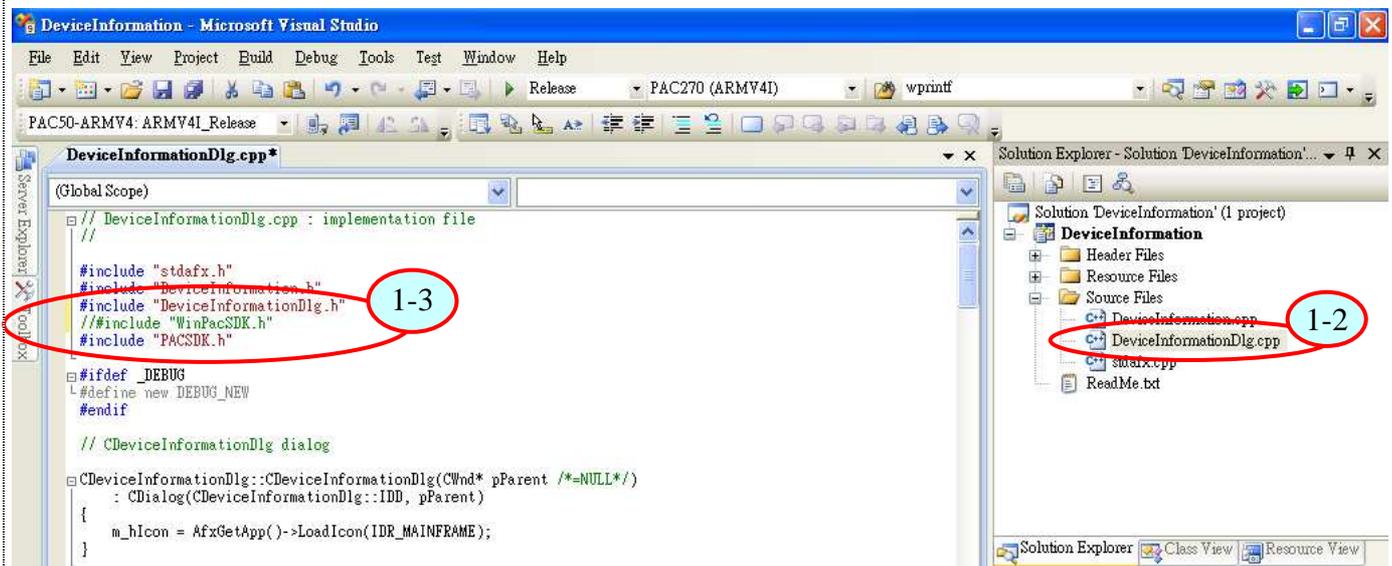
VC 程式

依以下步驟更新 PACSDK 函式庫至你的程式

步驟 1: 更改 WinPacSDK.h 為 PACSDK.h

- 1.1 開啟你的 WinPAC 系列模組使用的 **Visual Studio 2005 or 2008** 專案 (*.sln)
- 1.2 於 Solution explorer 上點選要修改的*.cpp 檔將此檔開啟
- 1.3 於 Editor 視窗將程式修改成以下的內容 (參考下圖)

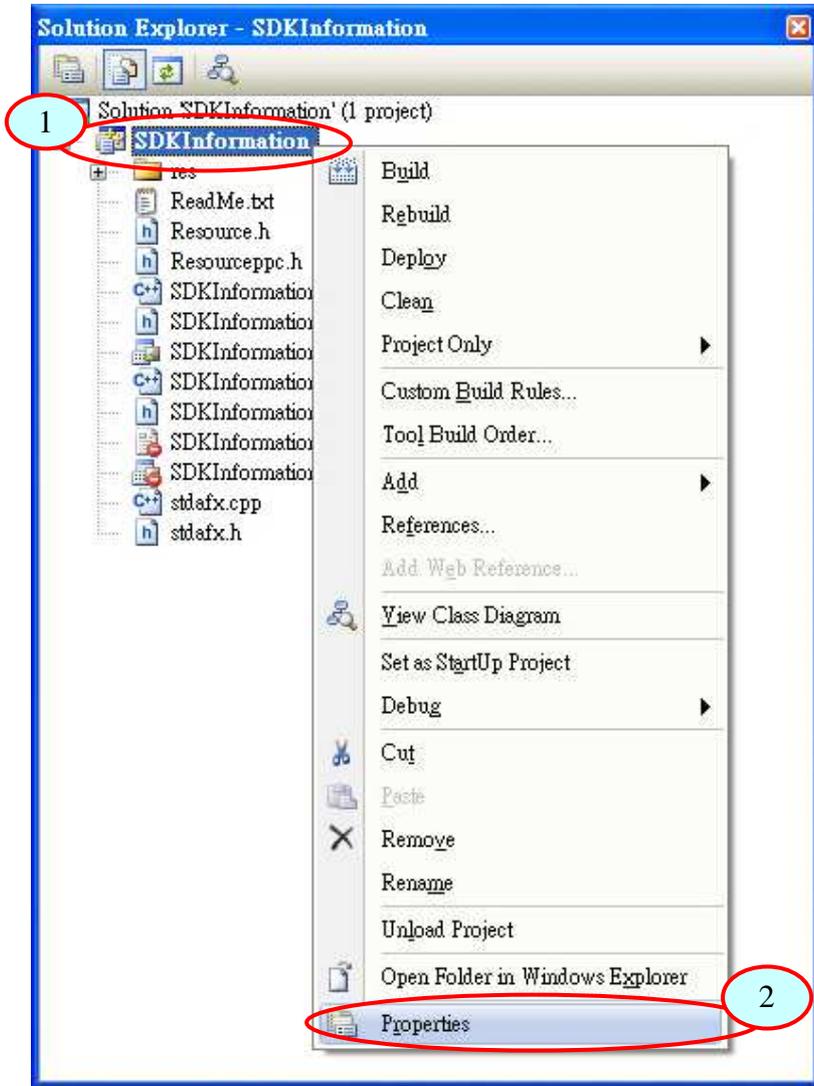
```
//#include "WinPacSDK.h"  
#include "PACSDK.h"
```



Classification	WinPAC SDK FAQ				No.	6-010-00	
Author	Sean Hsu	Version	1.0.1	Date	2012/10/18	Page	6/16

步驟 2: 更改 WinPacSDK.lib 為 PACSDK.lib

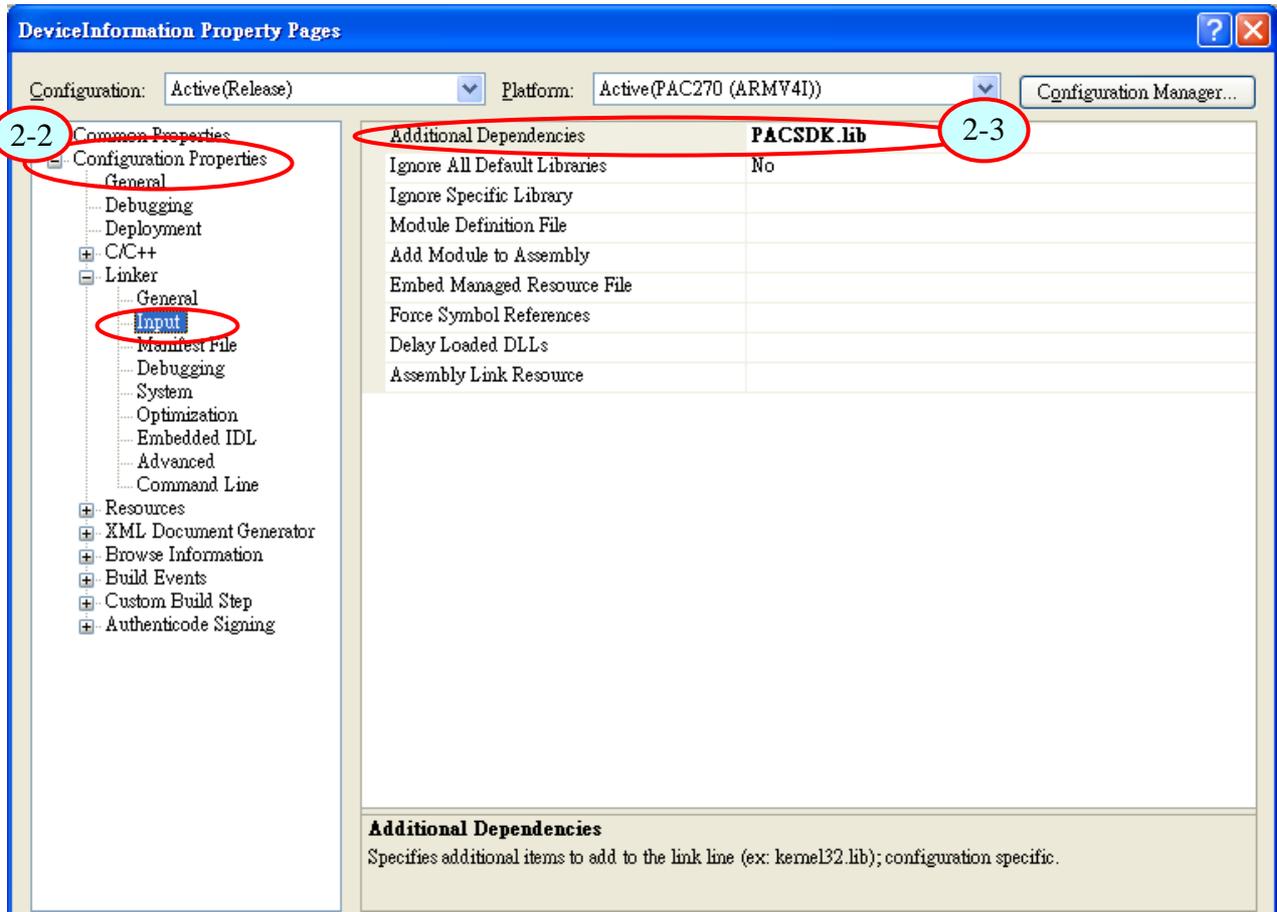
2-1 右鍵點擊 Solution explorer 視窗上的專案名稱，接著點選“Properties”項目 (如下圖，此例子專案名稱為 SDKInformation)



2-2 於出現的 Property Pages 視窗的左邊窗格(pane)上，展開 Configuration Properties 及 Linker 項目，然後點選 Input 選項。

2-3 在視窗的右邊窗格上，於 Additional Dependencies 文字框上輸入 PACSDK.lib (如下圖)。

Classification	WinPAC SDK FAQ				No.	6-010-00	
Author	Sean Hsu	Version	1.0.1	Date	2012/10/18	Page	7/16



步驟 3: 參考以下說明並確認你程式碼是否有以下的項目，如果有請修改你的程式碼

錯誤碼修改

```
//if(pac_GetLastError() == PAC_ERR_EEP_ACCESS_RESTRICTION)
```

修改為

```
if(pac_GetLastError() == PAC_ERR_EEP_INVALID_ADDRESS)
{
    //...
}
```

```
//if(pac_GetLastError() == PAC_ERR_SRAM_INVALID_TYPE)
```

修改為

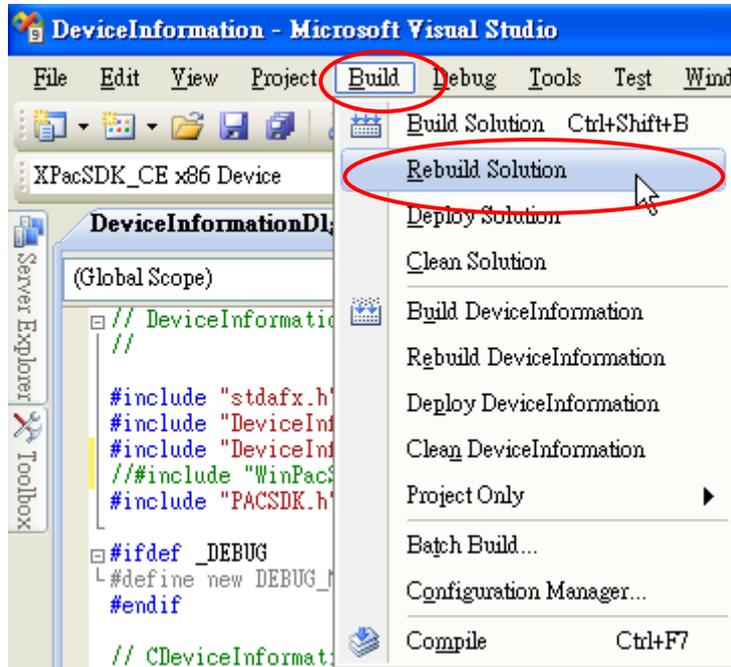
```
if(pac_GetLastError() == PAC_ERR_MEMORY_INVALID_TYPE)
{
    //...
}
```

PAC_ERR_EEP_ACCESS_RESTRICTION 及 PAC_ERR_SRAM_INVALID_TYPE 兩個錯誤碼是定義在 WinPacSDK.h，使用 PACSDK.h，需將它們修改為 PAC_ERR_EEP_INVALID_ADDRESS 及

Classification	WinPAC SDK FAQ				No.	6-010-00	
Author	Sean Hsu	Version	1.0.1	Date	2012/10/18	Page	8/16

PAC_ERR_MEMORY_INVALID_TYPE。

步驟 4: 在 Visual studio IDE 視窗上，點選 Build 選單的“Rebuild Solution”項目，重新編譯你的專案。



Classification	WinPAC SDK FAQ				No.	6-010-00	
Author	Sean Hsu	Version	1.0.1	Date	2012/10/18	Page	9/16

.NET Compact Framework 程式(C#, VB.net)

依以下步驟更新 PACNET 函式庫至你的程式

步驟 1: 更改 WinPacNet.dll 為 PACSDK.dll

1-1 取得 PACNET.dll 檔案及將它置於 .NET CF 專案目錄內。

PACNET.dll 可以由以下連結路徑得到

出貨 CD 內:

CD:\WinPAC\napdos\wp-8x4x_ce50\sdk\WinPacNet

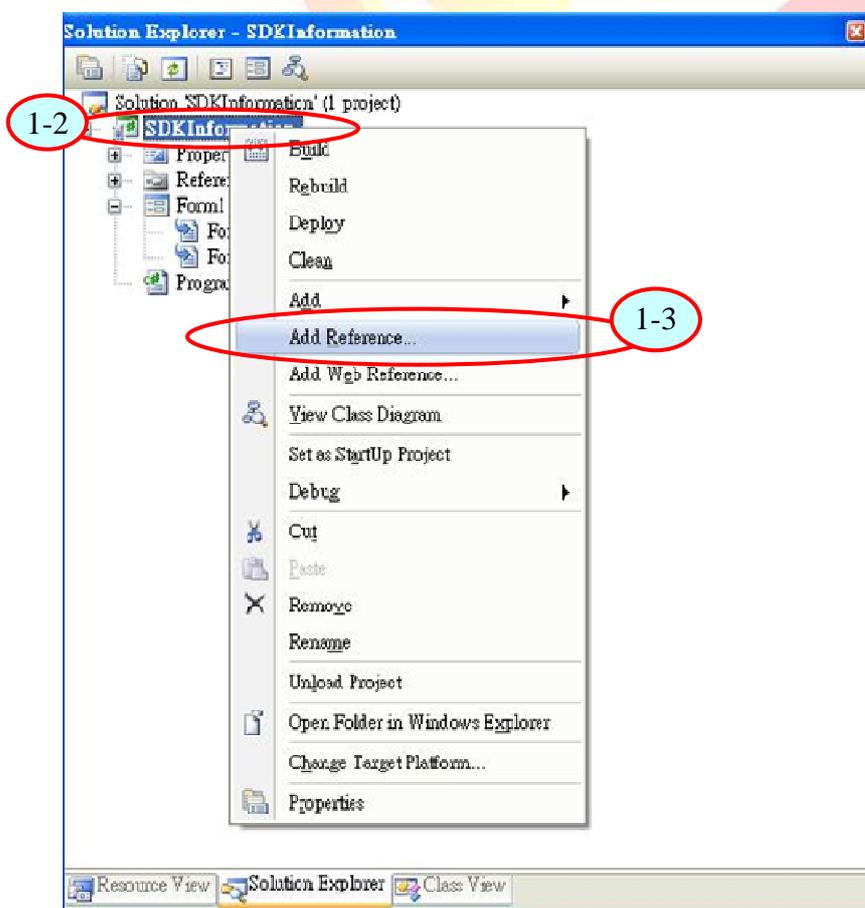
ICP DAS 網站:

http://ftp.icpdas.com/pub/cd/winpac/napdos/wp-8x4x_ce50/sdk/winpacnet/



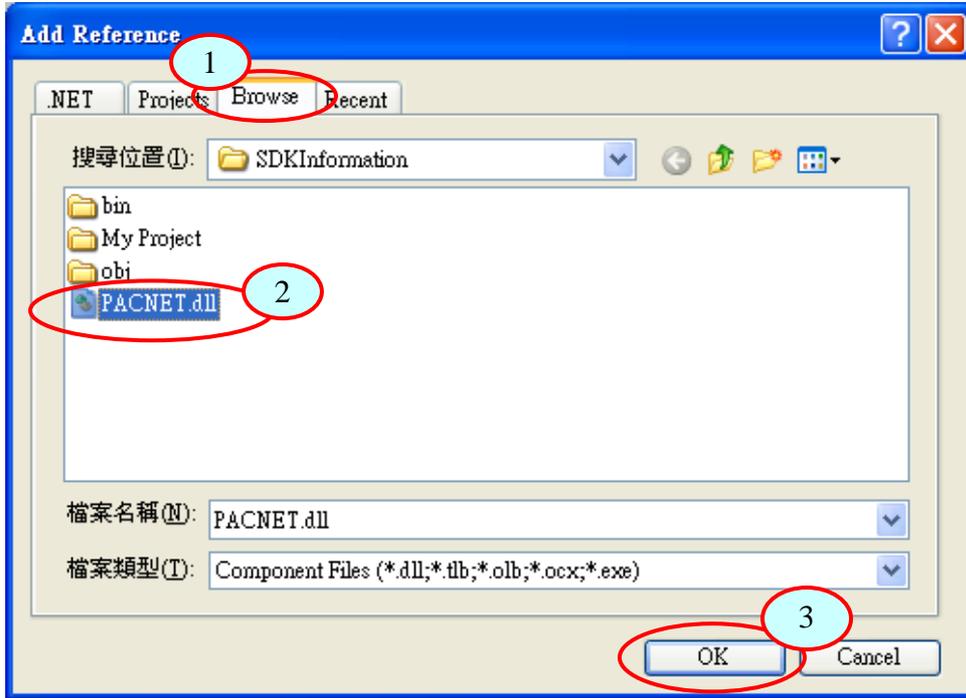
1-2 開啟你的 WinPAC 系列模組使用的 **Visual Studio 2005 or 2008 C#/VB.net** 專案 (*.sln)

1-3 右鍵點擊 Solution explorer 視窗上的專案名稱，接著點選“Add Reference...”項目 (如下圖，此例子專案名稱為 SDKInformation)



Classification	WinPAC SDK FAQ				No.	6-010-00	
Author	Sean Hsu	Version	1.0.1	Date	2012/10/18	Page	10/16

1-4 於“Add Reference”視窗上，選“Browse”標籤及點選 PACNET.dll 加入此專案。



步驟 2: 參考以下說明並確認你程式碼是否有以下的項目，如果有請修改你的程式碼。

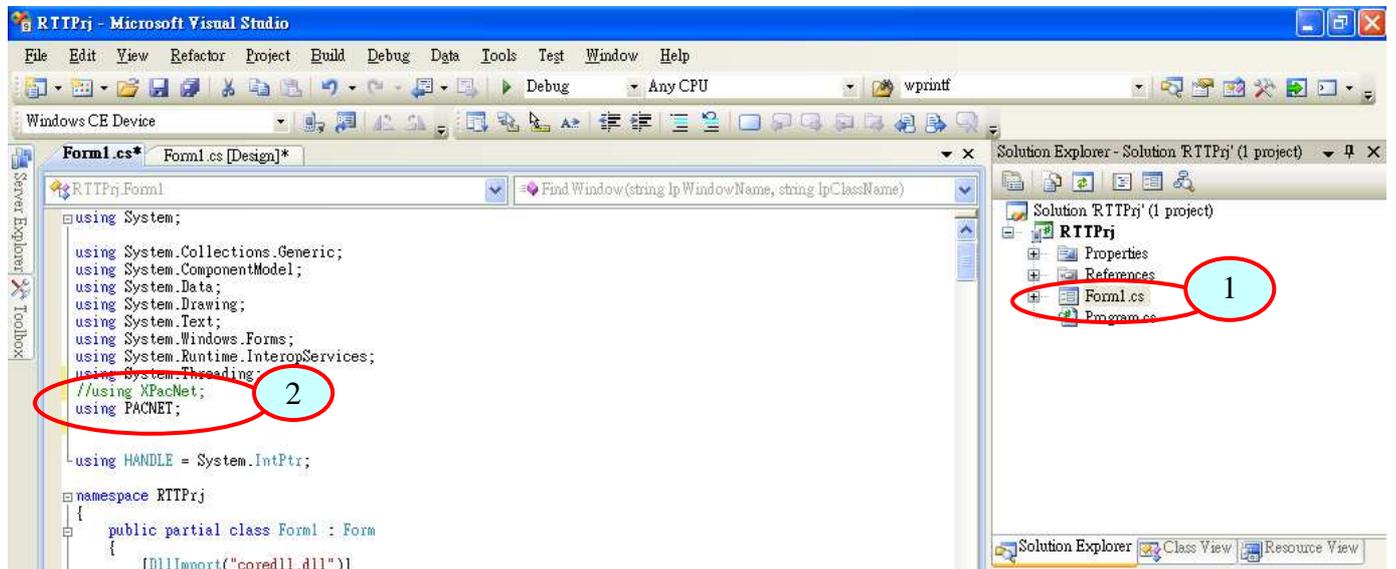
1. 修改程式碼

於 C# 程式將“using WinPacNET” 改為 “using PACNET”。

```
// using WinPacNet;
using PACNET;
```

於 VB 程式將“Imports WinPacNET” 改為 “Imports PACNET”。

```
//Imports WinPacNet
Imports PACNET
```



Classification	WinPAC SDK FAQ				No.	6-010-00	
Author	Sean Hsu	Version	1.0.1	Date	2012/10/18	Page	11/16

2. DllImport 修改

於 C# or VB program 使用 WinPAC SDK 的 API 函式，需使用 DllImport 命令

於 C#程式

```
//[DllImport("WinPacSDK.dll")]
```

```
[DllImport("PACSDK.dll")]
```

於 C#程中，使用 `DllImport("PACSDK.dll")` 取代 `DllImport("WinPacSDK.dll")`

於 VB 程式

```
//<DllImport("WinPacSDK.dll")>
```

```
<DllImport("PACSDK.dll")>
```

於 VB.net 程中，使用 `DllImport<"PACSDK.dll">` 取代 `DllImport<"WinPacSDK.dll">`

3. 類別(Class)名稱修改

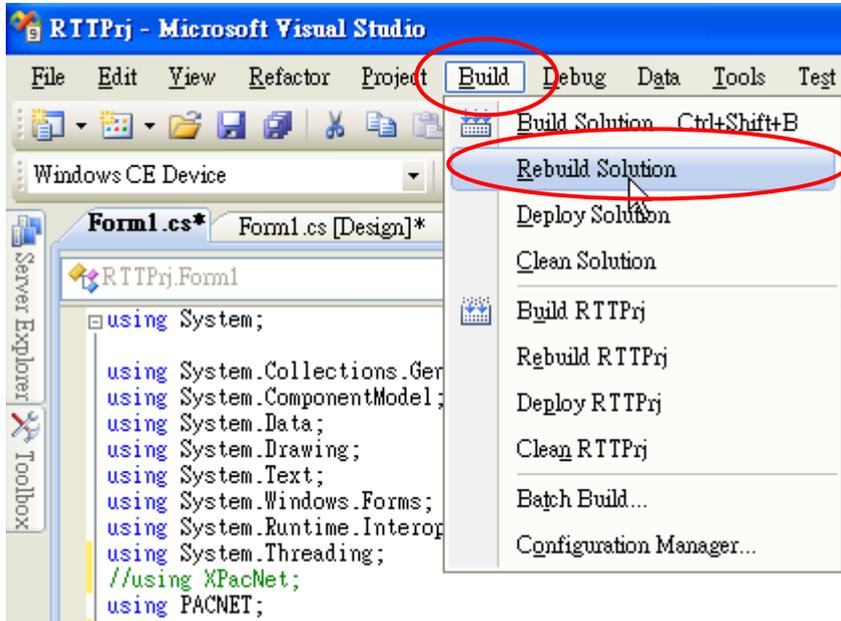
在 WinPacNet.dll 的所有 API 函式，都只被分類在一個類別(Class)內(WinPacNet.WinPAC.xxx), 在新的 PACNET.dll 中，API 函式被分類在不同的類別(Class)內，如 PACNET.Sys, PACNET.Memory, and PACNET.Interrupt, 等。

在 PACNET.dll 所有 API 函式 Class 分類，在 API 手冊有詳細定義說明，而主要分類對照表如下

API 手冊的分類名稱	PACNET.dll Class 分類名稱
2.1 System Information API	Sys
2.1 Backplane API	Sys
2.1 Buzzer API	Sys.Buzzer
2.2 Interrupt API	Interrupt
2.3 Memory Access API	Memory
2.4 Watchdog API	Sys.WDT
2.5 Registry API	PAC_Reg
2.6 UART API	UART
2.7 PAC_IO API	PAC_IO
2.8. PWM API	PWM
2.9. Backplane Timer API	BPTimer
2.10. Error Handling API	ErrHandling
2.11 MISC API	MISC

Classification	WinPAC SDK FAQ				No.	6-010-00	
Author	Sean Hsu	Version	1.0.1	Date	2012/10/18	Page	12/16

步驟 3: 在 Visual studio IDE 視窗上，點選 Build 選單的“Rebuild Solution”項目，重新編譯你的專案。



Classification	WinPAC SDK FAQ				No.	6-010-00	
Author	Sean Hsu	Version	1.0.1	Date	2012/10/18	Page	13/16

使用更新工具來更新 SDK

EasyUpgradeSDK.exe

EasyUpgradeSDK 是一支程式轉換工具，最主要的功能是将連結 XPacSDK 函式庫及 WinPacSDK 函式庫的 VC 程式轉換使用 PACSDK 函式庫。以及將連結 XPacNet 函式庫及 WinPacNet 函式庫的 C#.VB.net 程式轉換使用 PACNet 函式庫
此應用程式必須運作於一般 Windows PC 上。

EasyUpgradeSDK.exe 最新版本是置於 ICPDAS 網站:

http://ftp.icpdas.com/pub/cd/winpac/napdos/wp-8x4x_ce50/pc_tools/UpgradeSDK

EasyUpgradeSDK 只允許轉換檔案的副檔名為:

- *.cpp
- *.cs
- *.vb
- *.vcproj
- *.csproj
- *.vbproj
- *.vcp

這個工具的運作規則為:在使用者選取轉換的檔案後，程式會以關鍵字搜尋整個檔案內容，搜尋完成後並以新的關鍵字取代舊的。而且程式會自動尋找使用者選的目錄內是否有 Winpacsdk.h/Xpacsdk.h, Winpacsdk.lib/Xpacsdk.lib 或 WinPacNet.dll/XPacNet.dll 檔案存在。如果有會將 PACSDK.h, PACSDK.lib or PACNET.dll 拷貝至此目錄。(原來的.h/.lib/.dll 會保留不會刪除)

依以下步驟使用 EasyUpgradeSDK.exe 去更新 SDK

步驟 1: 確認檔案目錄架構是否如以下所示，將這個目錄拷貝至 Windows PC 上

```

EasyUpgradeSDK
├── EasyUpgradeSDK.exe
├── PACNET.dll
├── release_arm
│   ├── PACSDK.H
│   └── PACSDK.lib
├── release_x86
│   ├── PACSDK.H
│   └── PACSDK.lib

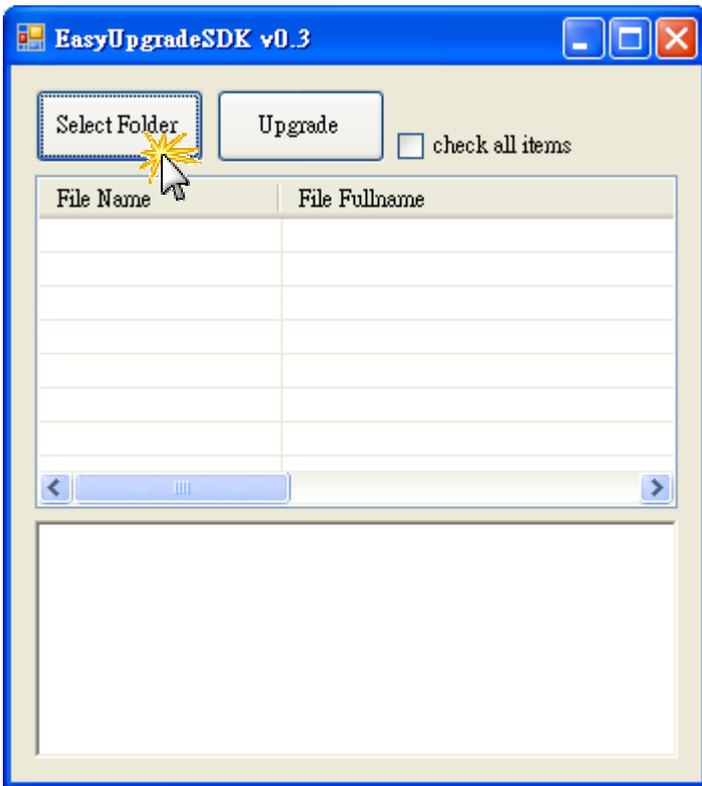
```

步驟 2: 執行 EasyUpgradeSDK.exe

步驟 3: 選擇一個目錄，專案的所有相關檔案都需在此目錄內。(這個專案必需是 WinPAC/XPAC 系列模組使用，且這個專案必須為 C#, VB.net, VC or eVC 所建立)

3-1: 點選 "Select Folder" 鍵(如下圖)

Classification	WinPAC SDK FAQ				No.	6-010-00	
Author	Sean Hsu	Version	1.0.1	Date	2012/10/18	Page	14/16

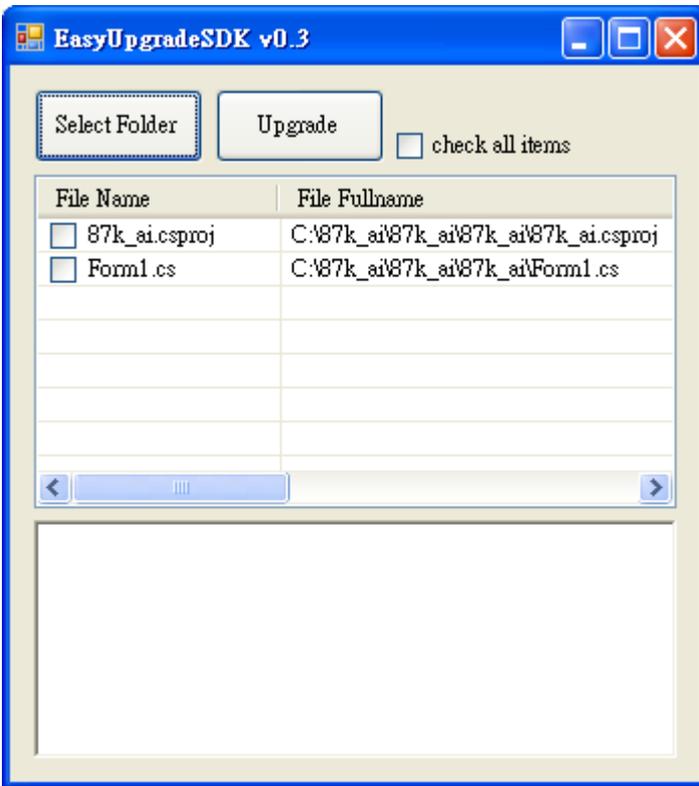


3-2: 點選一個目錄並按下“OK” 鈕。
(此例子，我們是選 C# 87k_dio demo)

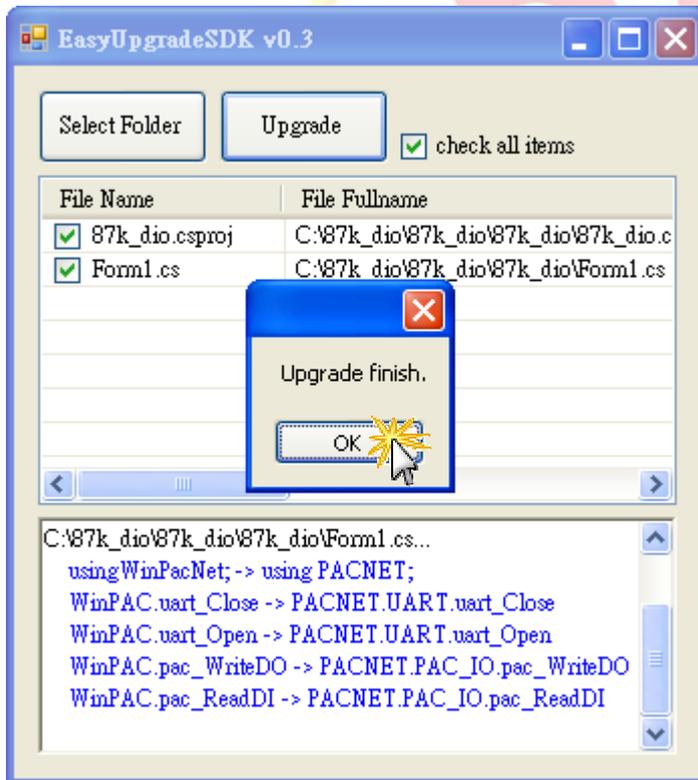


Classification	WinPAC SDK FAQ				No.	6-010-00	
Author	Sean Hsu	Version	1.0.1	Date	2012/10/18	Page	15/16

3-3: 點選“check all items” checkbox，接著按下“Upgrade” 鈕。



3-4: 等待，直到“Upgrade finish” messagebox 顯示在畫面(如下圖)



Classification	WinPAC SDK FAQ				No.	6-010-00	
Author	Sean Hsu	Version	1.0.1	Date	2012/10/18	Page	16/16

步驟 4: 完成轉換工作後，此轉換的程式碼仍需由開發程式重新編輯後才能使用。(參考手動更新 SDK 來重新編輯程式)

