

Classification	XPAC SDK FAQ				No.	6-009-00	
Author	Sean	Version	1.0.1	Date	2013/12/26	Page	1/18

如何使用新的SDK (PACSDK)

適用於:

平台	OS 版本	XPAC utility 版本
XPAC 系列 (CE6)	所有版本	所有版本

PAC SDK 整合

原本 WinPAC/ViewPAC(WinCE5.0)平台上使用的 C SDK 函式庫為 WinPacSDK.dll, 而 XPAC(WinCE6.0)平台使用的 C 函式庫為 XPacSDK_CE.dll。為了增進 WinPAC/XPAC 軟體相容性，將 WinPacSDK 及 XPacSDK_CE.dll 整合為一，命名為 PACSDK.dll。而.NET CF 函式庫，將原先 WinPacNet.dll 及 XPacNet.dll 整合為 PACNET.dll。

新/舊 SDK 檔名比較表

項目	WinPAC SDK	XPAC SDK (CE6)	PACSDK
開發使用的 header 檔	WinPacSDK.h	XPacSDK_CE.h	PACSDK.h PACSDK_PWM.h
開發使用的 library 檔	WinPacSDK.lib	XPacSDK_CE.lib	PACSDK.lib PACSDK_PWM.lib
目標裝置使用的原生 DLL 檔	WinPacSDK.dll	XPacSDK_CE.dll	PACSDK.dll PACSDK_PWM.dll
目標裝置使用的.NET CF DLL 檔	WinPacNet.dll	XpacNet.dll	PACNET.dll

整合的優點為

SDK 整合為一的對使用者而言，編寫的應用程式易於轉移。無論由 WinPAC 平台轉移至 XPAC 平台，或由 XPAC 平台轉移至 WinPAC 平台皆可。

Classification	XPAC SDK FAQ				No.	6-009-00	
Author	Sean	Version	1.0.1	Date	2013/12/26	Page	- 2 -/18

目錄

如何使用新的 SDK (PACSDK)	1
目錄	2
PACSDK.dll 變更及新增說明	3
1. LED 控制函式 (pac_EnableLED LED)	3
2. XPAC 新增 Registry API 函式組	3
3. PAC_IO 新增 DIO DCON 模組使用的 I/O WDT, Power On/Safe Value 函式	3
4. PAC_IO 新增對應 multi-function 模組存取的 API 函式	4
5. MISC 新增 API 函式	5
6. XPAC 增加記憶體系統保留區段	5
7. 於 C/C++的程式使用新版的 SDK(PACSDK)	6
PACNET SDK 變更及新增說明	8
1. API 函式的分類	8
2. API 函式新增及修改	8
3. 列舉錯誤碼的文字說明	9
4. 於 C#或 VB.net 的程式使用新版的 SDK(PACNET)	9
5. 顯示 Show a tooltip for the classes of PACNET.dll	10
錯誤碼的變更與新增	12
多功能(Multi-Function) DCON 模組的使用	14
WinPAC 使用者	14
XPAC 使用者	16
WinPACSDK.dll 昇級至 PACSDK.dll 會遇到的問題及解決方式	18
XPACSDK.dll 昇級至 PACSDK.dll 會遇到的問題及解決方式	18

Classification	XPAC SDK FAQ				No.	6-009-00	
Author	Sean	Version	1.0.1	Date	2013/12/26	Page	- 3 -/18

PACSDK.dll 變更及新增說明

新的 PACSDK.dll 有分 ARM 跟 x86 兩種平台,Arm 平台是給 WinPAC 系列模組使用, x86 平台 是給 XPAC (CE6)系列模組使用。以下有幾個項目是 PACSDK 變更及新增說明。(之前聯結 WinPAC/XPAC SDK 的程式，如果要改使用 PACSDK.dll，需要依以下說明來變更程式碼。如此，編譯後的程式才可以正確的執行)

1. LED 控制函式 (pac_EnableLED LED)

原來的 pac_EnableLED (bool bFlag)函式只能在 WinPAC 系列模組的程式中呼叫 WinPacSDK.dll 才可以使用。而 pac_EnableLED (int pin, bool bFlag)函式只能在 XPAC 系列模組的程式中呼叫 XPacSDK_CE.dll。

然而, 這個 API 函式不能被整合至 PACSDK.dll，因為有函式參數衝突的問題。

因此，在 PACSDK.dll 中將函式改爲

保留 pac_EnableLED (bool bFlag)

且增加一個新函式

pac_EnableLEDs (int pin,bool bFlag) .

2. XPAC 新增 Registry API 函式組

原本 XPAC 上使用的舊函式庫, XPACSDK_CE.dll 是不支援 Registry API 函式組。而新的 PACSDK 才支援此函式組。

(而 WinPAC 系列使用的舊函式庫, WinPACSDK.dll 及新的函式庫, PACSDK.dll 都有提供這一個函式組)函式組提供的所有函式，如下。

pac_RegCountKey
pac_RegCountValue
pac_RegCreateKey
pac_RegDeleteKey
pac_RegDeleteValue
pac_RegGetDWORD
pac_RegGetKeyByIndex
pac_RegGetKeyInfo
pac_RegGetString
pac_RegGetValueByIndex
pac_RegKeyExist
pac_RegSave
pac_RegSetString
pac_RegSetDWORD

3. PAC_IO 新增 DIO DCON 模組使用的 I/O WDT, Power On/Safe Value 函式

新的 PACSDK.dll 增加 I/O WDT, Power On and Safe value 函式提供給只有 DIO 通道的 DCON 模組使用。(參考 Note 1) 舊的函式庫 WinPacSDK.dll 及 XPacSDK_CE.dll 並不支援這些函式。

pac_GetModuleLastOutputSource

Classification	XPAC SDK FAQ				No.	6-009-00	
Author	Sean	Version	1.0.1	Date	2013/12/26	Page	- 4 -/18

pac_GetModuleWDTStatus
pac_GetModuleWDTConfig
pac_SetModuleWDTConfig
pac_ResetModuleWDT
pac_RefreshModuleWDT

pac_InitModuleWDTInterrupt
pac_SetModuleWDTInterruptStatus
pac_GetModuleWDTInterruptStatus

pac_ReadModuleSafeValueDO
pac_WriteModuleSafeValueDO

pac_ReadModuleSafeValueAO
pac_WriteModuleSafeValueAO

pac_ReadModulePowerOnValueDO
pac_WriteModulePowerOnValueDO

pac_ReadModulePowerOnValueAO
pac_WriteModulePowerOnValueAO

Note:

1. 以上的函式提供給具有 PowerON/ Safevalue 功能之 DCON 模組使用。
2. I-7K/I-87K 系統模組具有 PowerON/Safe Value 都可以使用以上函式, I-8K 模組目前提供 PowrON/SafeValue 為 I-8041RW。

4. PAC_IO 新增對應 multi-function 模組存取的 API 函式

新的 PACSDK.dll 增加 I/O 存取函式(包括讀寫 DIO/AIO 通道, 讀取 DI counter 及 I/O WDT, Power On and Safe value 函式)提供給多功能的 DCON 模組使用。(請參考 Note 2 了解多功能模組的定義) 舊的函式庫(WinPacSDK.dll 及 XPacSDK_CE.dll)並不支援這些函式。

pac_WriteAO_MF (Note 4)
pac_WriteModulePowerOnValueAO_MF
pac_WriteModuleSafeValueAO_MF
pac_WriteDO_MF
pac_ReadDIO_MF
pac_ReadDI_MF
pac_ReadDO_MF
pac_ReadDIO_DIBit_MF
pac_ReadDIO_DOBit_MF
pac_ReadDIBit_MF
pac_ReadDOBit_MF
pac_ReadDICNT_MF
pac_ClearDICNT_MF
pac_ReadModulePowerOnValueDO_MF
pac_WriteModulePowerOnValueDO_MF

Classification	XPAC SDK FAQ				No.	6-009-00	
Author	Sean	Version	1.0.1	Date	2013/12/26	Page	- 5 -/18

pac_ReadModuleSafeValueDO_MF
pac_WriteModuleSafeValueDO_MF

Note:

1. 原本舊的函式庫提供的 pac_WriteDO/ pac_ReadDIO/ pac_ReadDI/ pac_ReadDO/ pac_ReadDIO_DIBit/ pac_ReadDIO_DOBit/ pac_ReadDIBit/ac_ReadDOBit/ pac_ReadDICNT/ pac_ClearDICNT 等函式是給只具有 DI 或 DO 通道之 DCON 模組使用。
2. 後來新增不少多功能 DCON 模組(多功能模組是指 AI/AO 或 Counter 模組具有 DI 或 DO 通道), 如 I-87005W/I-87016W/I-87082W/I-7016/I-7088 等模組, 原來純 DIO 函式已經不敷使用。考量函式執行效能及簡單性, 另外提供以上幾組 pac_xxx_MF 函式, 供程式讀取多功能之 DCON 模組時使用。
3. 之前 WinPACSDK.dll 及 XPACSDK_CE.dll 提供的 pac_WriteDO...等 DIO 模組存取函式只支援存取 I-87K(高卡)/I-8K/I-7K 模組。PACSDK.dll 內所有 DIO 模組存取函式及所有多功能模組存取函式, 因程式處理的方式修改, 已經不用判斷模組名稱來變更送出的命令(即不受限於模組名稱)。所以 PACSDK.dll 的所有 IO API 函式, 可應用於存取 I-87K(高, 低卡)/I-8K/I-7K, i-8000 DCON, tM 模組等, 以及也可使用於之後新的模組或 OEM 模組(新的模組或 OEM 模組需符合規範的 DCON protocol)
4. pac_WriteAO/pac_WriteAO_MF 函式, 適用於這兩個函式的模組如下。

Since November 1, 2012

pac_WriteAO	Pac_WriteAO_MF
I-87024W/CW/DW/RW, I-87024	I-87026PW
I-87028CW/UW	
I-87022	
I-87026	
I-7021, I-7021P	
I-7022	
I-7024, I-7024R	

5. MISC 新增 API 函式

pac_GetCurrentDirectory
pac_GetCurrentDirectoryW

6. XPAC 增加記憶體系統保留區段

爲了保留 EEPROM 及 SRAM 部份區段記憶體給系統使用, PACSDK 函式庫的 pac_ReadMemory 及 pac_WriteMemory 函式有更改過。保留區段與 WinPAC SDK 相同。

保留區段的定義如下

Classification	XPAC SDK FAQ				No.	6-009-00	
Author	Sean	Version	1.0.1	Date	2013/12/26	Page	- 6 -/18

EEPROM

0 ~0x1FFF (8KB) 使用者使用

0x2000~0x3FFF (8KB) 保留給系統使用

SRAM

使用者可用的範圍為0 ~0x6FFFF (448KB), 最後一個64KB空間是保留給系統使用

在舊的XPAC SDK (XPacSDK_CE.dll), EEPROM的所有空間(0~0x3FFF, 16KB) 使用者都可以使用。而SRAM的所有空間(0~0x80000, 512KB) 也是提供給使用者使用。

7. 於C/C++的程式使用新版的SDK(PACSDK)

原本使用 C/C++開發的程式，在使用新的 PACSDK 函式庫時，需修改部份程式碼，如下：

-Header file 的部分

將#include "WinPacSDK.h"

改成

#include "PACSDK.h"

Note:

WinPacSDK.h 為 WinPAC/ViewPAC 系列開發 C/C++程式所使用的 header file，

將#include "XPacSDK_CE.h"

改成

#include "PACSDK.h"

Note:

XPacSDK_CE.h 為 XPAC 系列開發 C/C++程式所使用的 header file

-Lib 的部分

將 WinpacSDK.lib

XPacSDK_CE.lib

改成

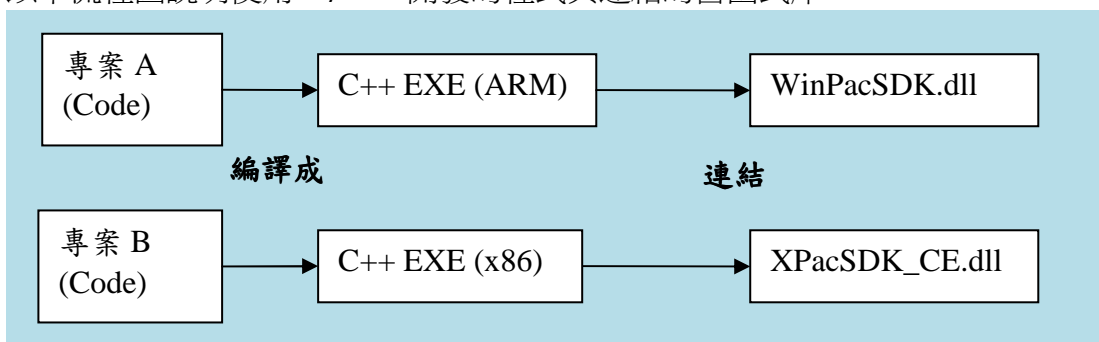
PACSDK.lib

Note:

WinpacSDK.lib 為 WinPAC 系列開發 C/C++程式所使用的靜態函式庫

XPacSDK_CE.lib 為 XPAC 系列開發 C/C++程式所使用的靜態函式庫

以下流程圖說明使用 C/C++開發的程式與連結的舊函式庫



專案 A 程式碼是開發應用於 WinPAC 系列模組，而專案 B 程式碼是開發應用於 XPAC 系列模

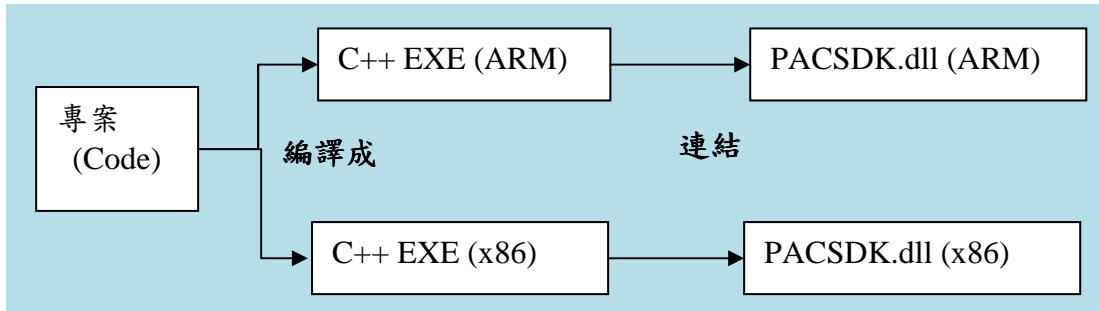
Classification	XPAC SDK FAQ				No.	6-009-00	
Author	Sean	Version	1.0.1	Date	2013/12/26	Page	- 7 -/18

組。即使兩個程式的功能完全相同，但使用舊的函式庫，其程式碼內使用不同的 header file 而且使用的一些函式名稱及錯誤碼名稱定義也不完全相同，所以整個程式碼是不同的。因此，上面圖示的專案 A 及專案 B 被視為兩支不同的程式，不能共同程式碼。

專案 A 會被編輯成一支 ARM 平台的執行程式，它必須連接 WinPacSDK.dll 才能運作正常。

專案 B 會被編輯成一支 x86 平台的執行程式，它必須連接 XPacSDK_CE.dll 才能運作正常。

以下流程圖說明使用 C/C++開發的程式與連結的新函式庫



連結新函式庫的好處：

一支應用於 WinPAC 系列模組的程式與另一支應用於 XPAC 系列模組的程式，如果功能完全相同。連結新的函式庫，使用相同的 Header file 而且函式庫的 API 函式及錯誤碼定義皆相同，所以這兩支程式可以共用同一個程式碼。

共同程式碼的專案在編譯時，於開發環境中選擇不同的平台設定後，可以編譯出兩支不同平台的執行程式。一支程式需連結 ARM-based 的 PACSDK.dll 而另一支程式需連結 x86-based 的 PACSDK.dll。

Classification	XPAC SDK FAQ				No.	6-009-00	
Author	Sean	Version	1.0.1	Date	2013/12/26	Page	- 8 -/18

PACNET SDK 變更及新增說明

在.NET Compact Framework 架構下，多種高階程式(C#, VB)在不用重新編譯過而仍可以在各種平台下執行，具有跨平台的優點。新的.NET CF 函式庫(PACNET.dll)取代舊的函式庫 (WinPacNet.dll and XPacNet.dll)後，原本在 WinPAC 系列模組上連結 PACNET.dll 的.NET CF 程式 可以移植至 XPAC 系列模組上執行，而且不用重寫程式碼或重新編譯專案。XPAC 的程式也是可以移植至 WinPAC 上。

1. API 函式的分類

在 WinPacNet.dll 或 XPacNet.dll 的所有 API 函式，都只被分類在一個類別(Class)內 (WinPacNet.WinPAC.xxx/XPacNET.XPac.xxx class),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
System Information API	Sys
Watchdog API	Sys.WDT
Buzzer API	Sys.Buzz
Interrupt API	Interrupt
Memory Access API	Memory
Registry API	PAC_Reg
UART API	UART
PAC_IO API	PAC_IO
Backplane Timer API	BPTimer
Error Handling API	ErrHandling
MISC API	MISC

2. API 函式新增及修改

LED 控制函式 (pac_EnableLED LED)

請參考上一節 PACSDK.dll 變更及新增說明 “LED 控制函式 (pac_EnableLED LED)”節有詳細的變更說明。

在 PACNET SDK 中,原來定義於 XPacNet.dll 的 XPacNet.XPac.pac_EnableLED(pin, bFlag) 函式變更為 PACNET.SysInfo.pac_EnableLEDs(pin, bFlag)。

XPAC 新增 Registry API 函式組

請參考上一節 PACSDK.dll 變更及新增說明 “XPAC 新增 Registry API 函式組”節有詳細的變更說明。

Registry API 函式組被歸類在 PACNET.PAC_Reg 類別中

PAC_IO 新增 DIO DCON 模組使用的 I/O WDT, Power On/Safe Value 函式

請參考上一節 PACSDK.dll 變更及新增說明“PAC_IO 新增 DIO DCON 模組使用的 I/O WDT, Power On/Safe Value 函式”節有詳細的變更說明。

I/O WDT, PowerOn/Safe Value API 函式組被歸類在 PACNET.IO 類別中

Classification	XPAC SDK FAQ				No.	6-009-00	
Author	Sean	Version	1.0.1	Date	2013/12/26	Page	- 9 -/18

PAC_IO 新增對應 multi-function 模組存取的 API 函式

請參考上一節 PACSDK.dll 變更及新增說明“PAC_IO 新增對應 multi-function 模組存取的 API 函式”節有詳細的變更說明。

存取 multi-function 模組 API 函式組被歸類在 PACNET.IO 類別中

MISC 新增 API 函式

請參考上一節 PACSDK.dll 變更及新增說明“MISC 新增 API 函式”節有詳細的變更說明。

MISC API 函式組被歸類在 PACNET.MISC 類別中

3. 列舉錯誤碼的文字說明

PACSDK 新增功能可以列舉所有的錯誤碼及它的文字說明

程式碼片斷如下(此程式碼可適用於每個 C#/VB demo)

```
uint ec = PACNET.ErrHandling.pac_GetLastError();
MessageBox.Show(((PACNET.ErrCode)ec).ToString() + "\nError Code: 0x" +
ec.ToString("X"));
```

以上程式碼加入使用者的.NET CF 程式內。

假如使用者的程式發生的最後一個錯誤碼為 0x10001 (0x10001 表示 “Slot registered error”, “Slot registered error 定義在 PACNET.XML 內)

程式將出一個訊息視窗顯示“Slot registered error Error Code:0x10001”。

Note: PACNET.dll 和 PACNET.XML 必須與.NET CF 執行檔置於相同路徑內，如此，.NET CF 執行檔才能正常的運作。

4. 於 C# 或 VB.net 的程式使用新版的 SDK (PACNET)

原本使用 C# 或 VB.net 開發的程式，在使用新的 PACSDK 函式庫時，需修改部份程式碼，如下

C# 程式

使用的裝置為 XPAC 系列，修改程式碼將“using XPacNET” 改為 “using PACNET”。如

```
using XPacNet;
```

修改為

```
using PACNET;
```

使用的裝置為 WinPAC 系列，修改程式碼將 “using WinPacNet” to “using PACNET”。

```
using WinPacNet;
```

修改為

```
using PACNET;
```

VB.net 程式

使用的裝置為 XPAC 系列，修改程式碼將“Imports XPacNET” 改為 “Imports PACNET”。

```
Imports XpacNet
```

修改為

```
Imports PACNET
```

Classification	XPAC SDK FAQ				No.	6-009-00	
Author	Sean	Version	1.0.1	Date	2013/12/26	Page	- 10 -/18

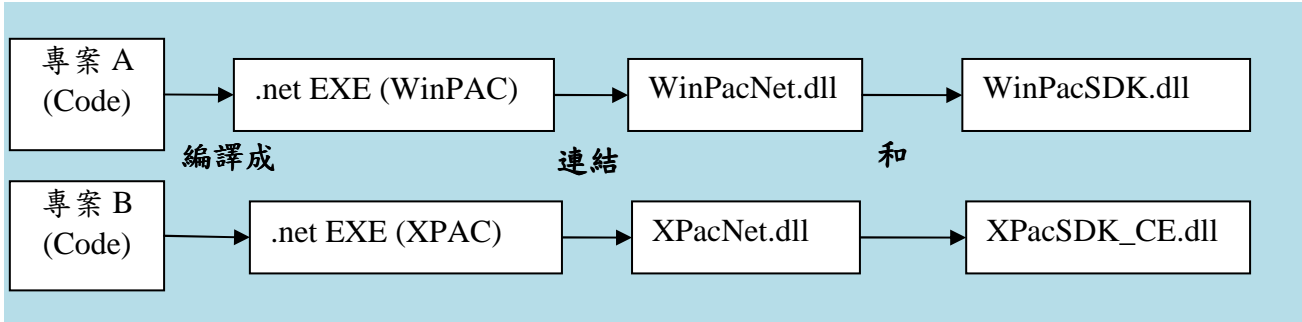
使用的裝置為 WinPAC 系列，修改程式碼將“Imports WinPacNet” 改為 “Imports PACNET”。

`Imports WinPacNet`

修改為

`Imports PACNET`

以下流程圖說明使用 .NET CF 開發的程式與連結的舊函式庫 (WinPacNet.dll 或 XPacNet.dll)

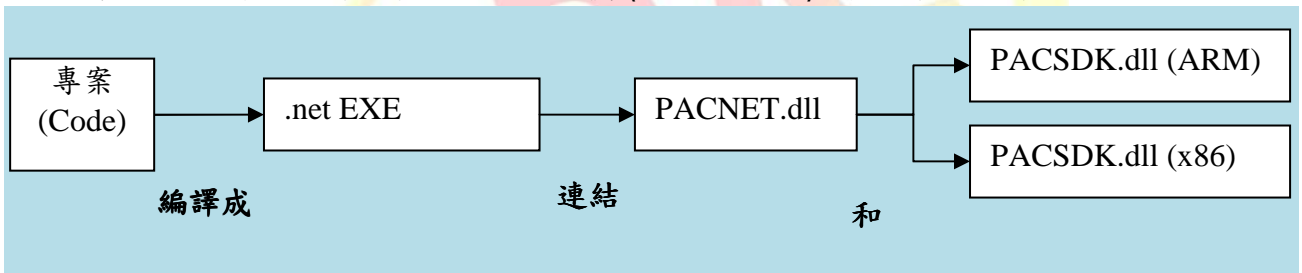


專案 A 程式碼是開發應用於 WinPAC 系列模組，而專案 B 程式碼是開發應用於 XPAC 系列模組。即使兩個程式的功能完全相同，但使用舊的函式庫，其程式碼內使用不同的 .NET CF 庫式庫而且使用的一些函式名稱及錯誤碼名稱定義也不完全相同，所以整個程式碼是不同的。因此，上面圖示的專案 A 及專案 B 被視為兩支不同的程式，不能共同程式碼。

專案 A 會被編輯成一支 WinPAC 系列模組使用的執行程式，它必須連接 WinPacNet.dll 及 WinPacSDK.dll 才能運作正常。

專案 B 會被編輯成一支 XPAC 系列模組使用的執行程式，它必須連接 XpacNet.dll 及 XPacSDK_CE.dll 才能運作正常。

用 C# 或 VB.net 開發的程式連結的新函式庫 (PACNET.dll)，其流程為以下圖示



連結新函式庫的好處：

一支應用於 WinPAC 系列模組的程式與另一支應用於 XPAC 系列模組的程式，如果功能完全相同。連結新的函式庫，使用相同的 .NET CF 函式庫而且函式庫的 API 函式及錯誤碼定義皆相同，所以這兩支程式可以共用同一個程式碼。

共同程式碼的專案可以編譯成一支跨平台的執行程式，連結的 .NET CF 函式庫也是同一個，PACNET.dll。唯一要改變的是 WinPAC 系列模組連接的原生 DLL 需為 ARM 平台的 PACSDK.dll，而於 XPAC 系統模組連接的原生 DLL 需為 x86 平台的 PACSDK.dll。

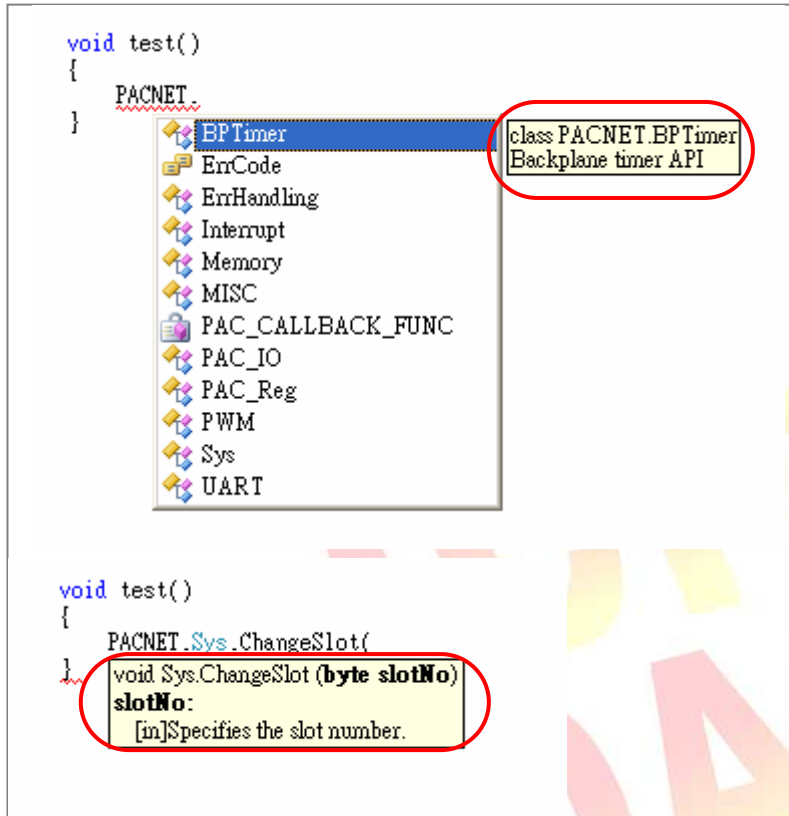
Notes:

PACNET.dll 是於 .NET CF V2.0 環境編譯產生的檔案，目前適用於所有 XPAC (CE6) 及 WinPAC 系列裝。

5. 顯示 Show a tooltip for the classes of PACNET.dll

Classification	XPAC SDK FAQ				No.	6-009-00	
Author	Sean	Version	1.0.1	Date	2013/12/26	Page	- 11 -/18

於 VS2005/VS2008 IDE 上開發程式。程式編輯視窗上，輸入一個系統類別(Class)/namespace 的方法，屬性等或滑鼠移過這個類別時，會有一個快顯視窗(tooltip)出現，這個視窗除了顯示使用方法的參數及變數外，還會顯示方法，類別，namespace 的說明。這些敘述與 PAC API 手冊上是相同的。(參考以下圖示)



Note:

程式參照的 PACNET.dll 及 PACNET.xml 兩個檔案需與置放於專案使用的同一個目錄內，如此，tooltip 才能在 Visual Studio 上正確顯示。



PACNET.dll



PACNET.XML

Classification	XPAC SDK FAQ				No.	6-009-00	
Author	Sean	Version	1.0.1	Date	2013/12/26	Page	- 12 -/18

錯誤碼的變更與新增 對 WinPAC 使用者而言 變更

原本定義於 WinPacSDK.h 的錯誤碼(PAC_ERR_EEP_ACCESS_RESTRICTION 及 PAC_ERR_SRAM_INVALID_TYPE)，在 PACSDK.h 被修改為 PAC_ERR_EEP_INVALID_ADDRESS 及 PAC_ERR_MEMORY_INVALID_TYPE

錯誤碼(PAC_ERR_MEMORY_BASE + 1)
PAC_ERR_EEP_ACCESS_RESTRICTION
改成
PAC_ERR_EEP_INVALID_ADDRESS

錯誤碼(PAC_ERR_MEMORY_BASE + 3)
PAC_ERR_SRAM_INVALID_TYPE
改成
PAC_ERR_MEMORY_INVALID_TYPE

新增

//Basic

PAC_ERR_MODULE_UNEXISTS (PAC_ERR_BASE + 7)
PAC_ERR_INVALID_SLOT_NUMBER (PAC_ERR_BASE + 8)

//Interrupt

PAC_ERR_INTR_BASE 0x13000
PAC_ERR_INTR_CREATE_EVENT_FAILURE (PAC_ERR_INTR_BASE + 1)

//UART

PAC_ERR_UART_INTERNAL_BUFFER_OVERFLOW (PAC_ERR_UART_BASE+9)

//IO

PAC_ERR_IO_DO_CANNOT_OVERWRITE (PAC_ERR_IO_BASE+10)
PAC_ERR_IO_AO_CANNOT_OVERWRITE (PAC_ERR_IO_BASE+11)

Classification	XPAC SDK FAQ				No.	6-009-00	
Author	Sean	Version	1.0.1	Date	2013/12/26	Page	- 13 -/18

對 XPAC 使用者而言

變更

在 XPacSDK_CE.h 定義的錯誤碼(PAC_ERR_INTR_CRATE_EVENT_FAILURE)在 PACSDK.h 被更正為 PAC_ERR_INTR_CREATE_EVENT_FAILURE

//Interrupt

錯誤碼(PAC_ERR_INTR_BASE + 1)

PAC_ERR_INTR_CRATE_EVENT_FAILURE

改成

PAC_ERR_INTR_CREATE_EVENT_FAILURE

//Basic

PAC_ERR_MODULE_UNEXISTS

原錯誤碼 PAC_ERR_BASE + 4 改成錯誤碼 PAC_ERR_BASE + 7

新增

//Basic

PAC_ERR_INVALID_MAC (PAC_ERR_BASE + 4)

PAC_ERR_INVALID_COMPORT_NUMBER (PAC_ERR_BASE + 5)

PAC_ERR_FUNCTION_NOT_SUPPORT (PAC_ERR_BASE + 6)

PAC_ERR_INVALID_SLOT_NUMBER (PAC_ERR_BASE + 8)

//Memory Access

PAC_ERR_NVRAM_INVALID_ADDRESS (PAC_ERR_MEMORY_BASE + 4)

PAC_ERR_EEP_WRITE_PROTECT (PAC_ERR_MEMORY_BASE + 5)

PAC_ERR_EEP_WRITE_FAIL (PAC_ERR_MEMORY_BASE + 6)

PAC_ERR_EEP_READ_FAIL
(PAC_ERR_MEMORY_BASE + 7)

//UART

PAC_ERR_UART_INTERNAL_BUFFER_OVERFLOW (PAC_ERR_UART_BASE+9)

//IO

PAC_ERR_IO_DO_CANNOT_OVERWRITE (PAC_ERR_IO_BASE+10)

PAC_ERR_IO_AO_CANNOT_OVERWRITE (PAC_ERR_IO_BASE+11)

Classification	XPAC SDK FAQ				No.	6-009-00	
Author	Sean	Version	1.0.1	Date	2013/12/26	Page	- 14 -/18

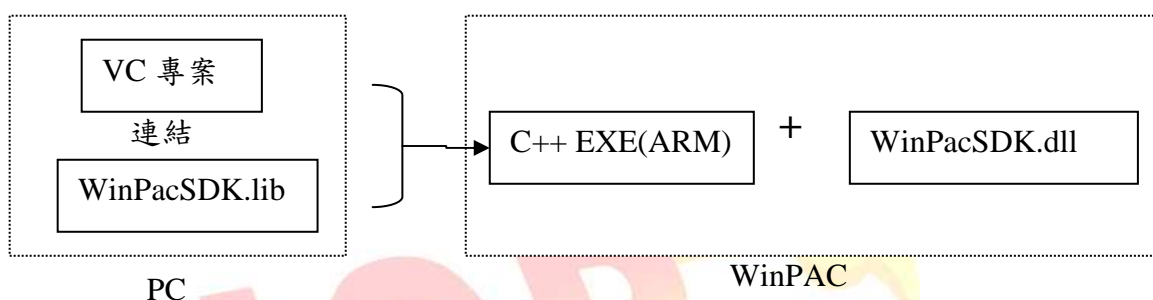
多功能(Multi-Function) DCON 模組的使用

WinPAC 使用者

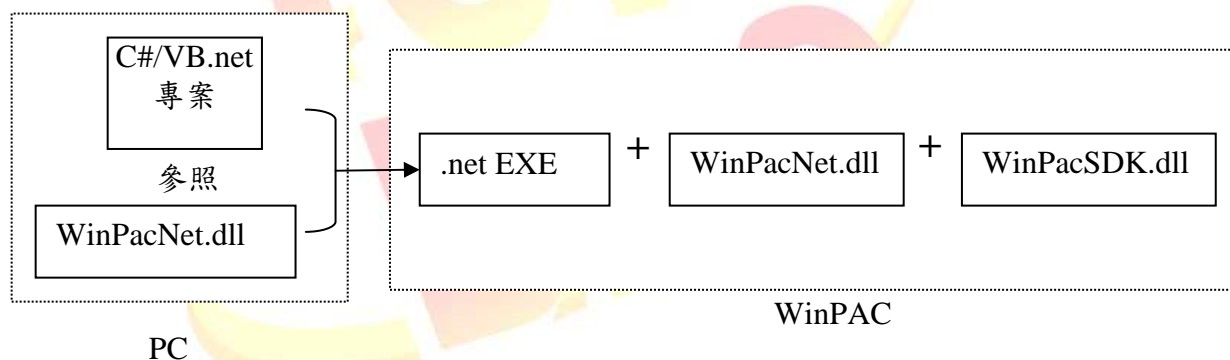
1. 使用舊的函式庫(WinPacSDK.dll/WinPacNet.dll)開發的程式，而搭配的模組仍是舊的 DCON 模組(**Note 2**)，沒有使用到新的 Multi-Function DCON 模組(**Note 1**)。原本使用舊的函式庫的程式，不需要做任何修改。(舊版的 SDK 仍會繼續維護修正 Bug 並定期發佈，但不會再加入新功能)

使用舊的 SDK,方式如下:

VC 專案於編譯時需連結 WinPacSDK.lib。編譯產生的執行檔，置於 WinPAC 系列裝置時需連結 WinPacSDK.dll



C#/VB.net 專案於編譯時需參照 WinPacNet.dll，編譯產生的執行檔，置於 WinPAC 系列裝置時需連結 WinPacSDK.dll/WinPacNet.dll。

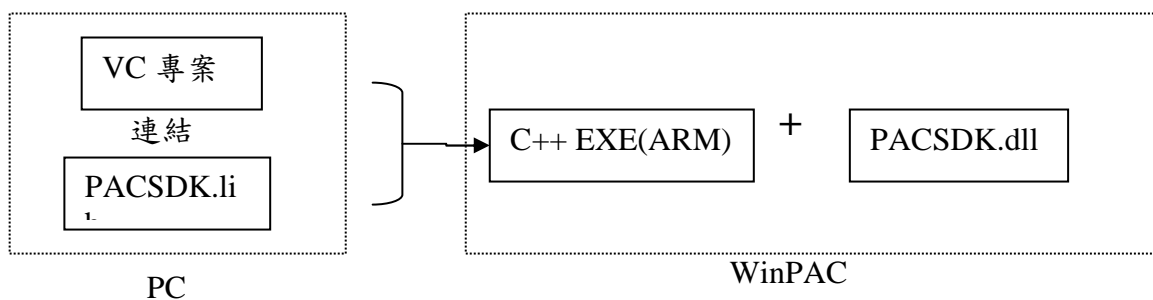


2. 使用舊的函式庫(WinPacSDK.dll/WinPacNet.dll)開發的程式，而搭配模組是舊的 DCON 模組及 Multi-Function DCON 模組。由於只有新的函式庫 (PACSDK.dll/PACNET.dll)才有提供 pac_xxx_MF API 函式去存取 Multi-Function DCON 模組，所以使用者的程式必需修改。
(詳細修改方式請參考 How-to document, w6-10_how_to_update_to_pacsdk_library_from_winpacsdk_library_tc.pdf)
3. 新的使用者-於 WinPAC 系列模組上使用舊的 DCON 模組及 Multi-Function DCON 模組。
我們的 API 手冊只會列出 PACSDK.dll 的使用方式及其 API 函式說明，且出貨 CD/FTP 上的 demo 也是提供連接新的函式庫(PACSDK.dll/PACNET.dll)。沒有對舊的函式庫做說明。建議新的使用者使用新的函式庫(PACSDK.dll/PACNET.dll)來開發程式。

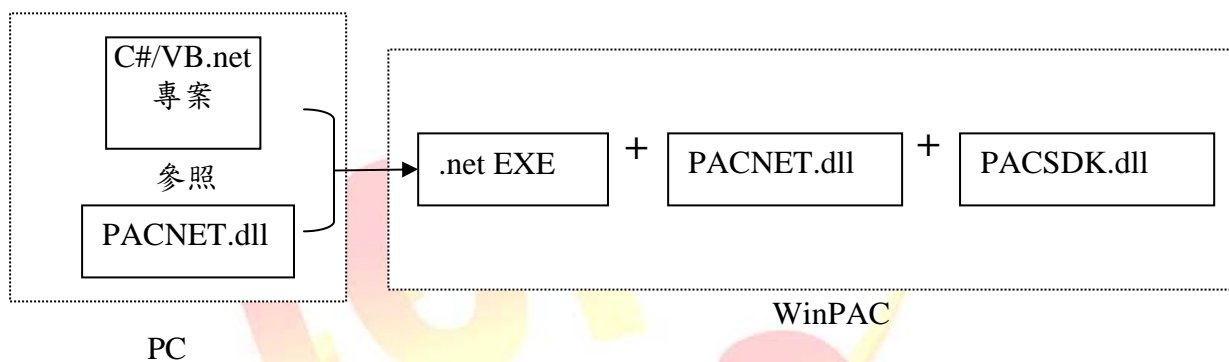
新的使用方式如下:

Classification	XPAC SDK FAQ				No.	6-009-00	
Author	Sean	Version	1.0.1	Date	2013/12/26	Page	- 15 -/18

VC 專案於編譯時需連結 PACSDK.lib。編譯產生的執行檔，置於 WinPAC 系列裝置時需連結 PACSSDK.dll



C#/VB.net 專案於編譯時需參照 PACNET.dll，編譯產生的執行檔，置於 WinPAC 系列裝置時需連結 PACSDK.dll/PACNET.dll。



Notes:

1. 多功能(Multi-Function)DCON 模組是指 AI/AO 或 Counter 模組具有 DI 或 DO 通道，如 I-87005W/I-87016W/I-87082W/I-7016/I-7088 等模組。
2. 舊的 DCON 模組定義:非多功能 DCON 模組即為舊的 DCON 模組。

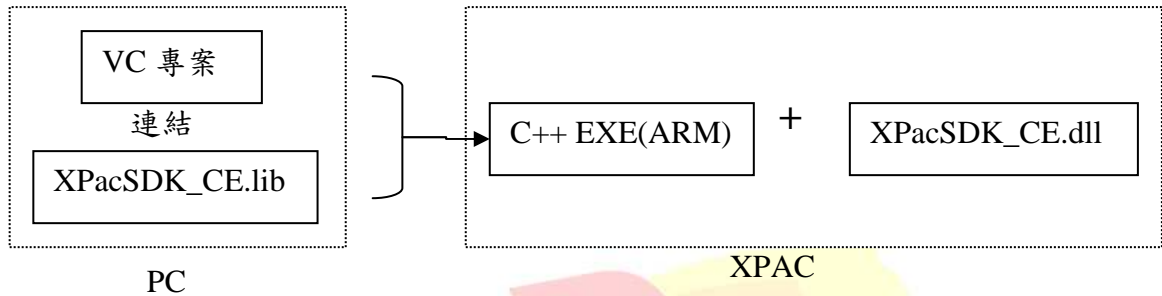
Classification	XPAC SDK FAQ				No.	6-009-00	
Author	Sean	Version	1.0.1	Date	2013/12/26	Page	- 16 -/18

XPAC 使用者

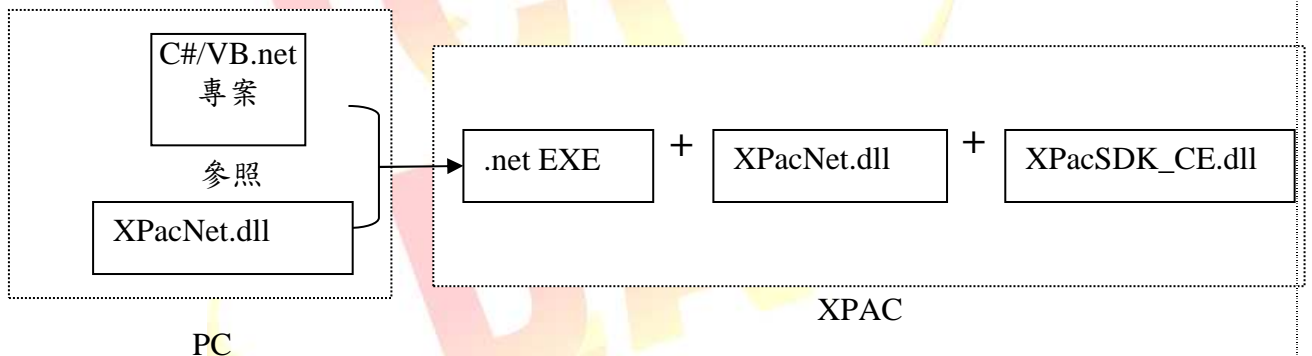
1. 使用舊的函式庫(XPacSDK_CE.dll/XPacNet.dll)開發的程式，而搭配的模組仍是舊的 DCON 模組，沒有使用到新的 Multi-Function DCON 模組。[原本使用舊函式庫的程式，不需要做任何修改。\(舊版的 SDK 仍會繼續維護修正 Bug 並定期發佈，但不會再加入新功能\)](#)

使用舊的 SDK,方式如下:

VC 專案於編譯時需連結 XPacSDK_CE.lib。編譯產生的執行檔，置於 XPAC 系列裝置時需連結 XPacSDK_CE.dll

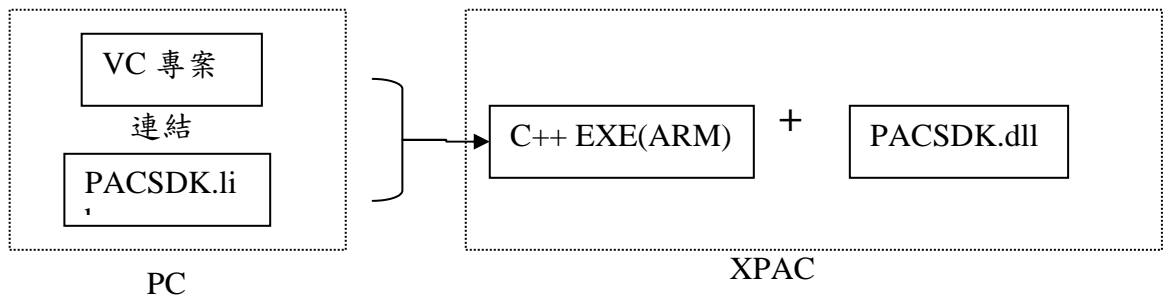


C#/VB.net 專案於編譯時需參照 XPacNet.dll，編譯產生的執行檔，置於 XPAC 系列裝置時需連結 XPacSDK_CE.dll/XPacNet.dll。

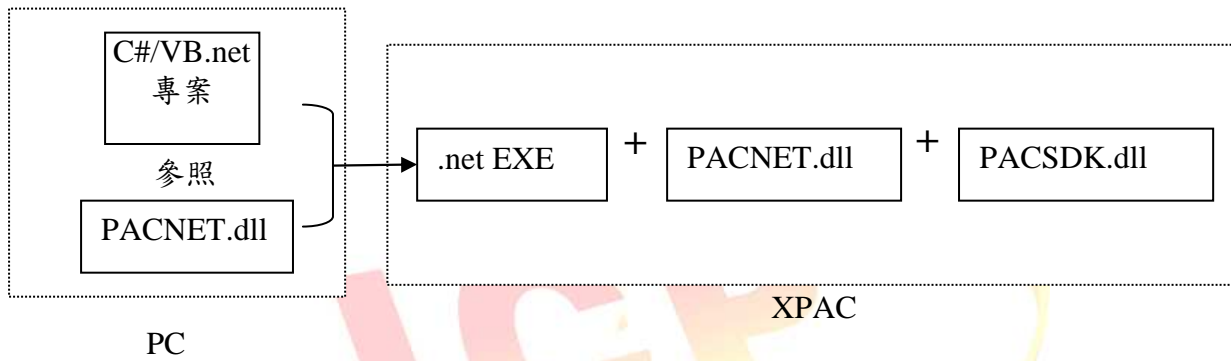


1. 使用舊的函式庫(XPacSDK_CE.dll/XPacNet.dll)開發的程式，而搭配模組是舊的 DCON 模組及 Multi-Function DCON 模組。[由於只有新的函式庫 \(PACSDK.dll/PACNET.dll\)才有提供 pac_xxx_MF API 函式去存取 Multi-Function DCON 模組，所以使用者的程式必需修改。](#)
(詳細修改方式請參考 [How-to document, x6-10_how_to_update_to_pacsdk_library_from_xpacsdk_library_tc.pdf](#))
2. 新的使用者，使用是舊的 DCON 模組及 Multi-Function DCON 模組。
我們的 API 手冊只會列出 PACSDK.dll 的使用方式及其 API 函式說明，且出貨 CD/FTP 上的 demo 也是提供連接新的函式庫(PACSDK.dll/PACNET.dll)。沒有對舊的函式庫做說明。建議新的使用者使用新的函式庫(PACSDK.dll/PACNET.dll)
新的使用方式如下:
VC 專案於編譯時需連結 PACSDK.lib。編譯產生的執行檔，置於 XPAC 系列裝置時需連結 PACSDK.dll

Classification	XPAC SDK FAQ				No.	6-009-00	
Author	Sean	Version	1.0.1	Date	2013/12/26	Page	- 17 -/18



C#/VB.net 專案於編譯時需參照 PACNET.dll，編譯產生的執行檔，置於 XPAC 系列裝置時需連結 PACSDK.dll/PACNET.dll。



Classification	XPAC SDK FAQ				No.	6-009-00	
Author	Sean	Version	1.0.1	Date	2013/12/26	Page	- 18 -/18

WinPACSDK 函式庫昇級至 PACSDK 函式庫會遇到的問題及解決方式

http://ftp.icpdas.com/pub/cd/winpac/napdos/wp-8x4x_ce50/document/faq/sdk/w6-010_how_to_update_to_pacsdk_library_from_winpacsdk_library_en.pdf

http://ftp.icpdas.com/pub/cd/winpac/napdos/wp-8x4x_ce50/document/faq/sdk/w6-010_how_to_update_to_pacsdk_library_from_winpacsdk_library_tc.pdf

XPACSDK 函式庫昇級至 PACSDK 函式庫會遇到的問題及解決方式

http://ftp.icpdas.com/pub/cd/xpac-atom-ce6/document/faq/sdk/x6-10_how_to_update_to_pacsdk_library_from_xpacsdk_library_tc.pdf

http://ftp.icpdas.com/pub/cd/xpac-atom-ce6/document/faq/sdk/x6-10_how_to_update_to_pacsdk_library_from_xpacsdk_library_en.pdf

