How to use the New SDK (PACSDK)

Applies to:

Platform	*OS Version*	*XPAC Utility Version*
WP-8x3x	*All versions (WinCE5)*	*All versions*
WP-8x4x	*All versions (WinCE5)*	*All versions*
VP-25W1	*All versions (WinCE5)*	*All versions*
VP-23W1	*All versions (WinCE5)*	*All versions*
WP-5000	*All versions (WinCE5)*	*All versions*

PACSDK API

The XPAC/WinPAC SDK has been unified and renamed PACSDK. The new PACSDK.dll provides support for two platforms, one being designed for the WinPAC series (ARM platforms) and the other for the XPAC series (x86 platforms).
PACSDK.dll (x86) is linked to C programs for the XPAC series to replace the previous SDK, XPACSDK_CE.dll, and PACSDK.dll(ARM) is linked to C programs for the WinPAC series to replace the previous SDK, WinPACSDK.dll.
The PACNET.dll is used for .Net CF programs (C#, VB) for both the XPAC and WinPAC series to replace the previous SDKs, XPACNET.dll and WinPACNET.dll.

New/Previous SDK files comparison

Items	**WinPACSDK Library**	**XPACSDK (CE6) Library**	**PACSDK Library**
Development header files	WinPacSDK.h	XPacSDK_CE.h	PACSDK.h PACSDK_PWM.h
Development library files	WinPacSDK.lib	XPacSDK_CE.lib	PACSDK.lib PACSDK_PWM.lib
Target device Native DLL files	WinPacSDK.dll	XPacSDK_CE.dll	PACSDK.dll PACSDK_PWM.dll
Target device .NET CF DLL files	WinPacNet.dll	XpacNet.dll	PACNET.dll

Benefits of the unified SDK include:

Easily migrates custom WinPAC programs to the XPAC series
Easily migrates custom XPAC programs to the WinPAC series

LRI Distribuída no Brasil pela LRI | vendas@lri.com.br | 51 2129-3000

TABLE OF CONTENTS

How to use the New SDK (PACSDK) 1
PACSDK.dll modifications and updates 3
1. pac_EnableLED 3
2. Add Registry API for XPAC series 3
3. Add I/O WDT, PowerOn/Safe Value API for pure DIO modules 4
4. Add I/O accessing API functions for the Multi-function modules 5
5. Add Misc. API function for PACSDK 6
6. Add the reserved memory section for XPAC series 6
7. Using the new SDK (PACSDK) in a C program 7
PACNET SDK modifications and updates 9
1. API function classification 9
2. API function modification 9
3. Enumerate the error codes 10
4. Using the new SDK (PACNET) in a C# or VB.net program 10
5. Show a tooltip for the classes of PACNET.dll 12
Error code modifications and updates 13
Using the Multi-function DCON module 15
On WinPAC devices 15
On XPAC devices 17
Questions related to updating the PACSDK.dll from the WinPACSDK.dll and the solutions 19
Questions related to updating to PACSDK.dll from the XPACSDK.dll and solutions 19

LRI Distribuída no Brasil pela LRI | vendas@lri.com.br | 51 2129-3000

PACSDK.dll modifications and updates

The new PACSDK.dll provides support for two platforms, one being designed for the WinPAC series (ARM platforms) and the other for the XPAC series (x86 platforms). However, there are a number of modifications and updates that are included in the new PACSDK, which are listed below.

(**Note**: Compared to the previous WinPAC/XPAC SDK, these modification and updates need to be made to the previously implemented WinPAC/XPAC programs so that it will work with the new SDK)

1. pac_EnableLED

The original **pac_EnableLED (bool bFlag)** function can be used only for the WinPAC series in the previous SDK, and the original **pac_EnableLED (int pin, bool bFlag)** function can be used only for the XPAC series in the previous SDK.

Consequently, this API function cannot be integrated to the PACSDK.dll because of the conflicting parameters.

As a result, the function in PACSDK.dll has been changed to

pac_EnableLED (bool bFlag) is been reserved and

a new API function has been added,

pac_EnableLEDs (int pin,bool bFlag) .

2. Add Registry API for XPAC series

The suite of the API functions listed below doesn't been provided in the previous SDK, XPACSDK_CE.dl, and supported in the new PACSDK.dll for the XPAC series.

(The previous SDK, WinPACSDK.dll for the WinPAC series and the new SDK, PACSDK.dll for the WinPAC series have provided the support of all the functions below)

pac_RegCountKey
pac_RegCountValue
pac_RegCreateKey
pac_RegDeleteKey
pac_RegDeleteValue
pac_RegGetDWORD
pac_RegGetKeyByIndex
pac_RegGetKeyInfo
pac_RegGetString
pac_RegGetValueByIndex
pac_RegKeyExist

LRI Distribuída no Brasil pela LRI | vendas@lri.com.br | 51 2129-3000

pac_RegSave
pac_RegSetString
pac_RegSetDWORD

3. Add I/O WDT, PowerOn/Safe Value API for pure DIO modules

The new PACSDK.dll provides the support of I/O WDT, Power On and Safe value functions for pure DIO DCON modules. (Refer to Note 1) These functions aren't supported for the previous SDK, WinPacSDK.dll and XPacSDK_CE.dll.

pac_GetModuleLastOutputSource
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

Notes:

1. The each of API function is used for the DCON module which is provided with Power ON or Safe value function.
2. I-7K/I-87K series modules provided with Power ON or Safe Value function can support the API functions above. I-8K series module provide the functions is only I-8041RW。

LRI Distribuída no Brasil pela LRI | vendas@lri.com.br | 51 2129-3000

4. Add I/O accessing API functions for the Multi-function modules

The new PACSDK.dll provides the support of I/O accessing functions (including Write/Read DIO, AIO, Read DI counter and I/O WDT, Power On and Safe value function for the Multi-function DCON modules. (Refer to Note 2 regarding of the definition of Multi-function modules) These functions aren't supported for the previous SDK, WinPAC.dll and XPACSDK_CE.dll.

pac_WriteAO_MF
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
pac_ReadModuleSafeValueDO_MF
pac_WriteModuleSafeValueDO_MF

Notes:

1. The functions pac_WriteDO, pac_ReadDIO, pac_ReadDI, pac_ReadDO, pac_ReadDIO_DIBit, pac_ReadDIO_DOBit, pac_ReadDIBit, pac_ReadDOBit, pac_ReadDICNT and pac_ClearDICNT, which were supported in the previous SDK, are used to read and write the DIO channels for pure DIO DCON modules, which are defined as modules that only have DI, DO or DIO channels.
2. In addition to providing support for the API functions described above, the PACSDK also provides the support for the Multi-function API that is used to read and write the DIO channels for the Multi-function DCON modules, which are defined as modules that mainly act as AIO or Counters but are equipped with DIO channels. Such as the I-87005W/I-87016W/I-87082W/I-7016/I-7088, etc.)
3. The functions mentioned above (i.e., pac_WriteDO/ pac_ReadDIO, etc.) cannot be used to access Multi-function DCON modules. Only the pac_xxx_MF API allows access to Multi-function DCON modules.
4. In both the WinPACSDK.dll and the XPACSDK_CE.dll, PAC_IO API functions only support access to high profile I-87K/I-8K series modules and I-7K series modules. In the PACSDK.dll, the processing can be modified to send DCON commands without needing to determine the module name, which means that a the new PAC_IO API functions can support access to the I-87K/I-8K (High profile and Low profile series modules), I-7K series modules, I-8000 series modules units, tM series modules, and other OEM/ODM DCON modules.

5. The comparison table of pac_WriteAO / pac_WriteAO_MF Functions and available modules

LRI Distribuída no Brasil pela LRI | vendas@lri.com.br | 51 2129-3000

are as following:

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. Add Misc. API function for PACSDK

The new PACSDK.dll provides 2 miscellaneous API functions below.

pac_GetCurrentDirectory
pac_GetCurrentDirectoryW

6. Add the reserved memory section for XPAC series

In order to reserve some memory sections of EEPROM and SRAM for the use by the system, the reserved section of the pac_ReadMemory and pac_WriteMemory function must be changed. Thee reserved section is same with the WinPAC SDK. The definition of the items included in the reserved section is

EEPROM

0 ~0x1FFF (8KB) for users
0x2000~0x3FFF (8KB) is reserved for the system

SRAM

The size of the input range for the SRAM is only 0 ~0x6FFFF (448KB), with another 64KB of SRAM is reserved for use by the system.

In the previous XPAC SDK (XPacSDK_CE.dll), all memory space (0~0x3FFF, 16KB) of EEPROM is available for the use by the user, and all memory space (0~0x80000, 512KB) of SRAM is available for the use by the user.

LRI Distribuída no Brasil pela LRI | vendas@lri.com.br | 51 2129-3000

7. Using the new SDK (PACSDK) in a C program

To use the new PACSDK in a C-based program, some code needs to be changed in the program.

Replace the previous header file by PACSDK.h

#include "WinPacSDK.h"

Changed as

#include " PACSDK.h"

WinPacSDK.h is used for both WinPAC or ViewPAC series program and it must be replaced by PACSDK.h

#include "XPacSDK_CE.h"

Changed as

#include " PACSDK.h"

XPacSDK_CE.h is used for the XPAC series program and it must be replaced by PACSDK.h

Replace the previous library file by PACSDK.lib

WinpacSDK.lib *// WinPAC or ViewPAC series*

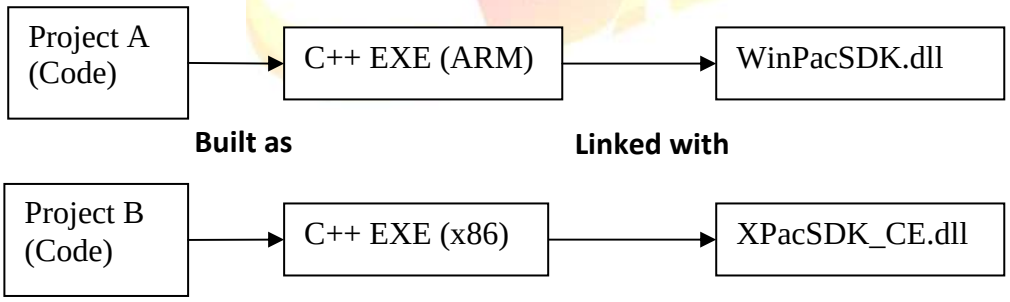
XPacSDK_CE.lib *// XPAC series*

Changed as

PACSDK.lib

WinPacSDK.lib used for WinPAC or ViewPAC series and XPacSDK_CE.lib used for XPAC series are replaced by PACSDK.lib

The original flowchart for a C program that is calling the previous SDK is illustrated below



Even if Project A applied to WinPAC series modules and Project B applied to XPAC series modules are functionally identical. The source code using the previous SDK cannot be exactly the same because of using the different header file and the few function names and error code defined in the previous SDK are different. So Project A and Project B are regarded as separate programs, cannot share the source code

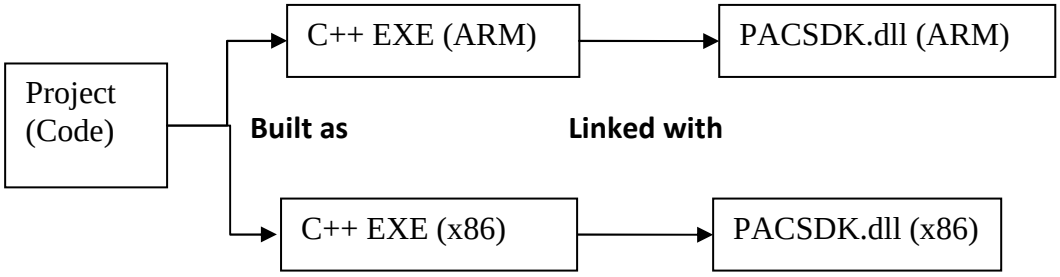
The results of the above are

Project A is built as an ARM-based executable program and it must be run with WinPacSDK.dll.

Project B is built as an x86-based executable program and it must be run with XPacSDK_CE.dll.

LRI Distribuída no Brasil pela LRI | vendas@lri.com.br | 51 2129-3000

The flowchart for a C program that is now calling the new SDK (PACSDK.dll) is as follows:



The benefits of using the new SDK:
A program applied to WinPAC series modules and the other program applied to XPAC series modules are functionally identical, because using the same header file and the API functions and error code on the library are exactly the same, the source code can be shared for two programs.
The Project with the shared source code can be built as two different platform executable programs selecting the different Platform settings in the development environment while build the project.
The results of the above are
Project is built as an ARM-based executable program which runs with the ARM-based PaCSDK.dll and it's also built as an x86-based executable program which runs with x86-based PaCSDK.dll.

Distribuída no Brasil pela LRI | vendas@lri.com.br | 51 2129-3000

PACNET SDK modifications and updates

The .NET Compact Framework environment allows multiple high-level languages (C#, VB) to be used on different platforms without needing to be rewritten for specific architectures. The new PACNET.dll replaces the previous .NETCF SDK, WinPacNet.dll and XPacNet.dll files which means that NET CF programs linking to the PACNET.dll on a WinPAC device can be migrated to a XPAC device without needing to rewrite the code or rebuild the project and vice versa.

1. API function classification

All API functions for the WinPacNet.dll or the XPacNet.dll are placed in a single WinPacNet.WinPAC.xxx/XPacNET.XPac.xxx class, but the API functions for the PACNET.dll are classified as PACNET.sys, PACNET.Memory, and PACNET.Interrupt, etc.
The classifications applied to the API functions for the PACNET.dll as defined in the API user manual are as follows.

Classification in the API manual	Class Name in PACNET.dll
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

2. API function modification

LED control API function (pac_EnableLED)

Refer to “pac_EnableLED” reference of PACSDK.dll modifications and updates for more details.
The modification in PACNET SDK, XPacNet.XPac.pac_EnableLED(pin, bFlag) function defined in XPacNet.dll has been changed as PACNET.SysInfo.pac_EnableLEDs(pin, bFlag) PACNET.dll.

Add Registry API for XPAC series

Refer to “Add Registry API for XPAC series” reference of PACSDK.dll modifications and updates for more details.
The suite of the Registry API functions is placed in PACNET.PAC_Reg class.

Add I/O WDT, PowerOn/Safe Value API for pure DIO modules

Refer to “Add I/O WDT, PowerOn/Safe Value API for pure DIO modules” reference of PACSDK.dll

LRI Distribuída no Brasil pela LRI | vendas@lri.com.br | 51 2129-3000

modifications and updates for more details.
The suite of the I/O WDT, PowerOn/Safe Value API functions for pure DIO modules is placed in PACNET.PAC_IO class.

Add I/O WDT, PowerOn/Safe Value API for the Multi-function modules

Refer to "Add I/O WDT, PowerOn/Safe Value API for Multi-function modules" reference of PACSDK.dll modifications and updates for more details.
The suite of the I/O WDT, PowerOn/Safe Value API functions for Multi-function modules is also placed in PACNET.PAC_IO class.

Add Misc. API function for PACSDK

Refer to "Add Misc. API function for PACSDK" reference of PACSDK.dll modifications and updates for more details.
The suite of misc. API function is placed in PACNET.MISC class.

3. Enumerate the error codes

Add a function to enumerate all the error codes for PACSDK
The code snippet is as follows (The code is applicable to every C#/VB demo file)

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

The sample code is used to show the error code number and its e***numerated definition.***
If the last error code, 0x10001 is happened on the user's program.
The message box with "PAC_ERR_UNKNOWN Error Code:0x10001" caption will be shown.

4. Using the new SDK (PACNET) in a C# or VB.net program

To use the new PACNET in a C# or VB.net program, some code needs to be changes in the program.

In a C# program

Modify the code for using XPAC series devices, "using XPacNET" to "using PACNET".

```
using XPacNet;
```

Changed as

```
using PACNET;
```

Modify the code for using XPAC series devices, "using WinPacNet" to "using PACNET".

```
using WinPacNet;
```

Changed as

```
using PACNET;
```

In a VB.net program

Modify the code for using XPAC series devices, "Imports XPacNET" to "Imports PACNET".

```
Imports XpacNet
```

Changed as

```
Imports PACNET
```


LRI Distribuída no Brasil pela LRI | vendas@lri.com.br | 51 2129-3000

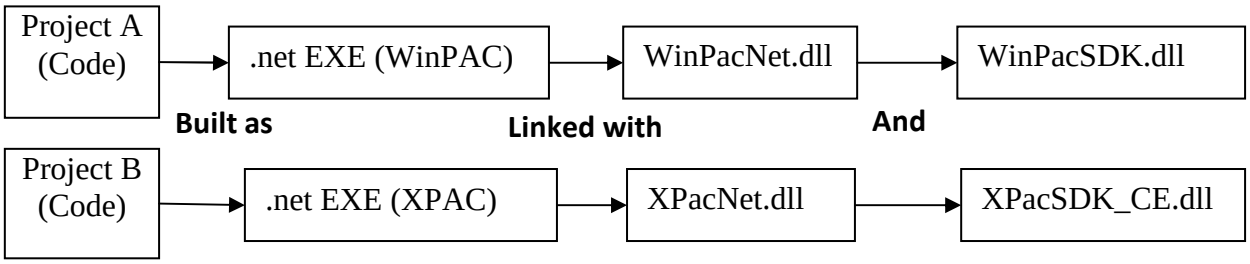
Modify the code for using XPAC series devices, “Imports WinPacNet” to “Imports PACNET”.

```
Imports WinPacNet
```

Changed as

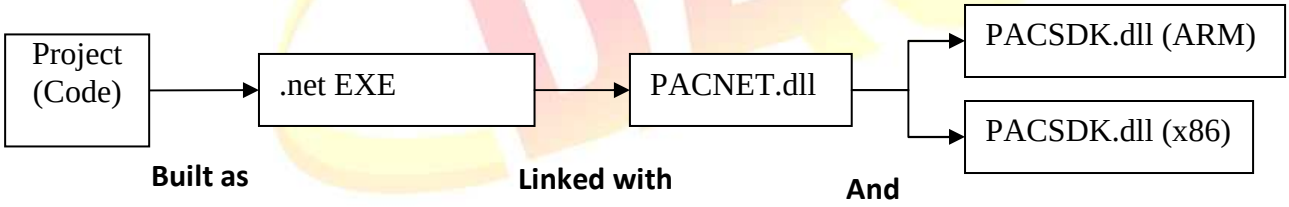
```
Imports PACNET
```

With the previous .NETCF library (WinPacNet.dll or XPacNet.dll), the flowchart was as follows:



Project A applied to WinPAC series modules and Project B applied to XPAC series modules are functionally identical, but the source code cannot be exactly the same because of using the different .NET CF library and few function name and error code are different. So Project A and Project B are regarded as separate programs, no relevance.
Project A for WinPAC series is built as an executable program which must be run with WinPacNet.dll and WinPacSDK.dll.
Project B for XPAC is built as an executable program which must be run with XpacNet.dll and XPacSDK_CE.dll.

With the new .NETCF library (PACNET.dll) and the flowchart becomes:



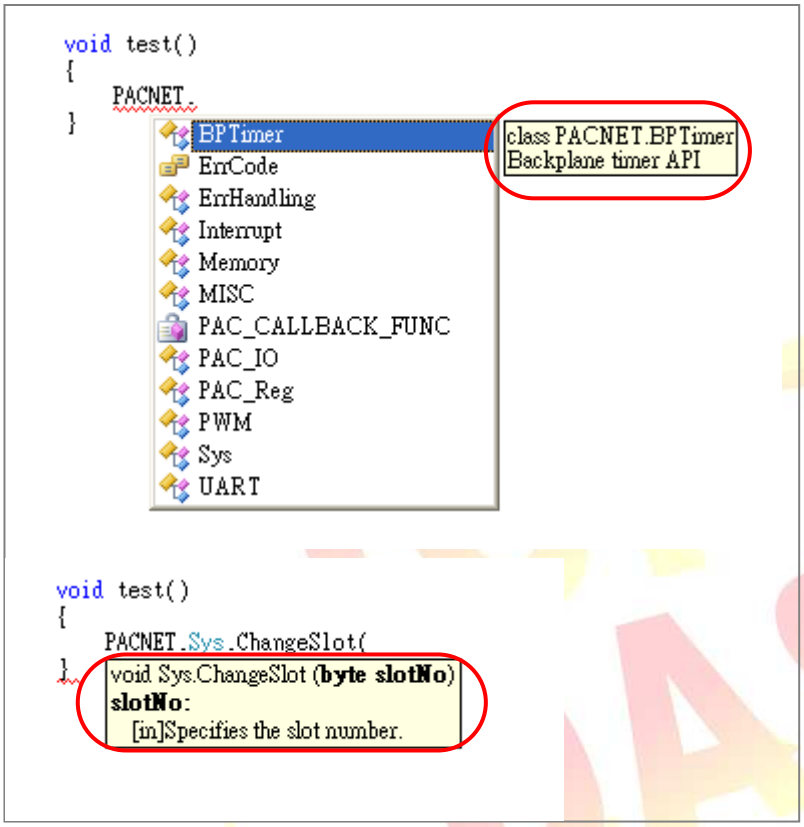
The benefits of using the new SDK:
A program applied to WinPAC series modules and the other program applied to XPAC series modules are functionally identical, because of using the same .NET CF library and the API functions and error code on the library are exactly the same, the source code can be shared for two programs. One shared source code can be built as an executable programs and link the same .NET CF library (PACNET.dll). The only change is that links different platform native SDK. (PACSDK.dll (ARM) is used on WinPAC series and PACSDK.dll(x86) is used for XPAC series)

Notes:
PACNET.dll has been developed using the .Net CF V2.0 environment and can be used on all XPAC and WinPAC series devices.

LRI Distribuída no Brasil pela LRI | vendas@lri.com.br | 51 2129-3000

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

When developing the programs in VS2005/VS2008 IDE, typing a reference to a system class or namespace or roll over class, the tooltips pop up on your cursor line giving not only the parameters and variables of methods, but also some descriptions for these methods, classes and namespaces. Those description of tooltips are same on the PAC API manual. (Refer to the following figure)



Note*:*

The PACNET.dll referred to the project and PACNET.xml must be placed at the same folder and the tooltip will show in the Visual studio IDE well.



LRI Distribuída no Brasil pela LRI | vendas@lri.com.br | 51 2129-3000

Error code modifications and updates

For WinPAC series

Modify

The error code, PAC_ERR_EEP_ACCESS_RESTRICTION and PAC_ERR_SRAM_INVALID_TYPE defined in WinPacSDK.h are modified as PAC_ERR_EEP_INVALID_ADDRESS and PAC_ERR_MEMORY_INVALID_TYPE defined in PACSDK.h.

Error code (PAC_ERR_MEMORY_BASE + 1)
PAC_ERR_EEP_ACCESS_RESTRICTION
Changed to
PAC_ERR_EEP_INVALID_ADDRESS

Error code (PAC_ERR_MEMORY_BASE + 3)
PAC_ERR_SRAM_INVALID_TYPE
Changed to
PAC_ERR_MEMORY_INVALID_TYPE

Add

```
//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)
```


Distribuída no Brasil pela LRI | vendas@lri.com.br | 51 2129-3000

For XPAC series

Modify

The error code,PAC_ERR_INTR_CRATE_EVENT_FAILURE defined in XPacSDK_CE.h is misspelled, and it is corrected in PACSDK.h as PAC_ERR_INTR_CREATE_EVENT_FAILURE

//Interrup
Error code (PAC_ERR_INTR_BASE + 1)
PAC_ERR_INTR_CRATE_EVENT_FAILURE
Changed to
PAC_ERR_INTR_CREATE_EVENT_FAILURE

//Basic
PAC_ERR_MODULE_UNEXISTS
Original Errorcode: PAC_ERR_BASE + 4
Changed to
PAC_ERR_BASE + 7

Add

//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)

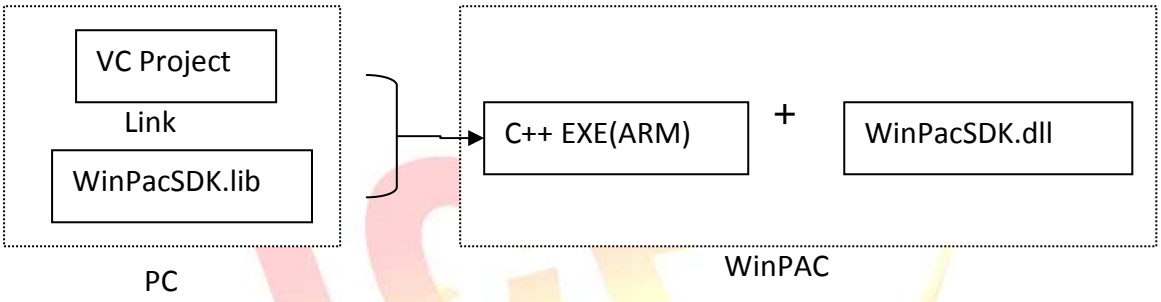
Distribuída no Brasil pela LRI | vendas@lri.com.br | 51 2129-3000

Using the Multi-function DCON module

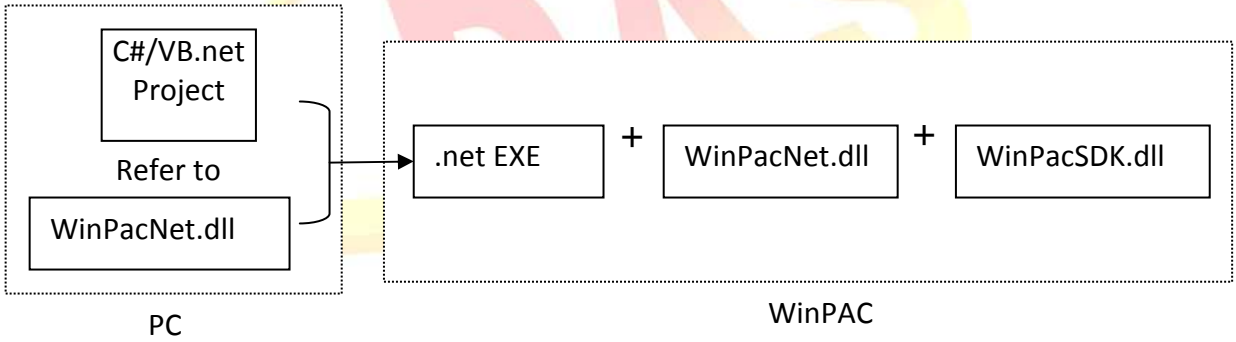
On WinPAC devices

1. The users have used WinPAC series devices and their programs is based on the old SDK (WinPacSDK.dll/WinPacNet.dll) working with the old DCON modules (***Note 2***) on WinPAC device and without using multi-function DCON modules (***Note 1***).
   The user's program can continue to use the old library without needing to be modified. (The Old SDK will continue to maintain (Fix the bugs) and released regularly, but will not add new features)

   Use the old SDK as following flowchart:
   The VC project required to link WinPacSDK.lib while building, and the built executable file placed in the WinPAC series device must work with WinPacSDK.dll



   The C#/VB.net project required to refer to WinPacNet.dll while building, and the built executable file placed in the WinPAC series device must work with WinPacNet.dll and WinPacSDK.dll.



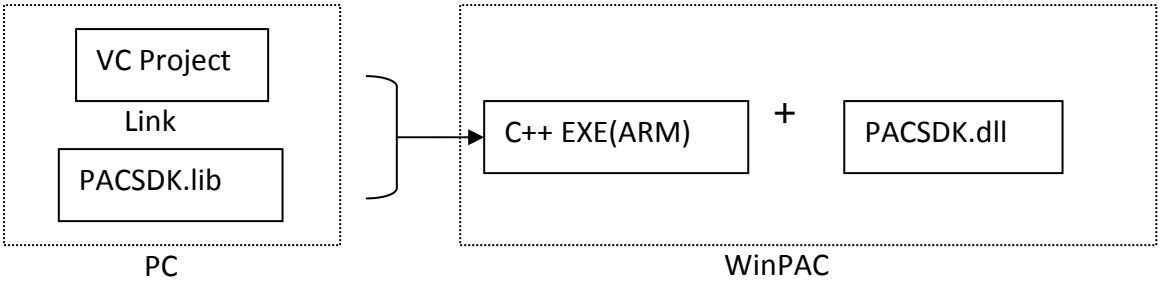
2. The users have used WinPAC series devices and their programs is based on the old SDK (WinPacSDK.dll) working with the old DCON modules and multi-function DCON modules on WinPAC device. The new PACSDK.dll provides pac_xxx_MF API functions that allow access to Multi-function modules, **so the code must be updated in order to use the new PACSDK.dll in the program.**
   (Refer to How-to document, w6-10_how_to_update_to_pacsdk_library_from_winpacsdk_library_en.pdf for more details)

3. The users have never used WinPAC series devices. Their program will be based on the new SDK working with an old DCON module or a Multi-function module. Our API Manual give instructions for the PACSDK.dll and the demo programs included on the shipped CD/FTP are linked with the new PACSDK.dll, so users should refer to the demo programs and follow the

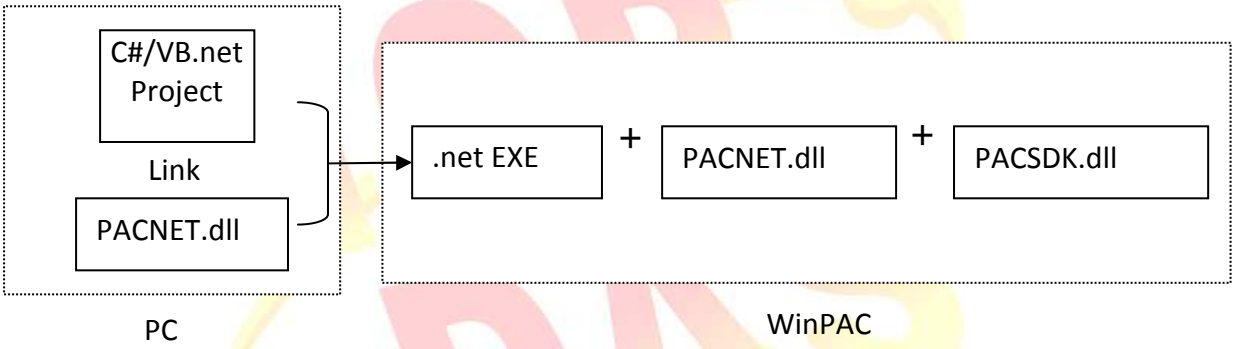
LRI Distribuída no Brasil pela LRI | vendas@lri.com.br | 51 2129-3000

API instructions when developing new programs based on the new PACSDK.dll, rather than those for the WinPACSDK.

Use the new SDK as following flowchart:
The VC project required to link PACSDK.lib while building, and the built executable file placed in the WinPAC series device must work with PACSDK.dll



The C#/VB.net project required to refer to PACNET.dll while building, and the built executable file placed in the WinPAC series device must work with PACNET.dll and PACSDK.dll.



Notes:

1. Multi-function DCON modules are defined as modules that mainly act as AIO or Counters but are equipped with DIO channels. Such as the I-87005W/I-87016W/I-87082W/I-7016/I-7088, etc.)
2. Old DCON module definition: Non multi-function DCON modules are defined as Old DCON modules.

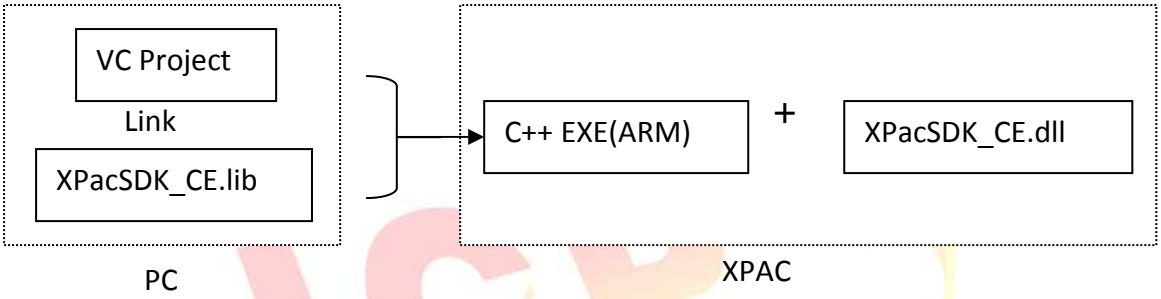
LRI Distribuída no Brasil pela LRI | vendas@lri.com.br | 51 2129-3000

On XPAC devices

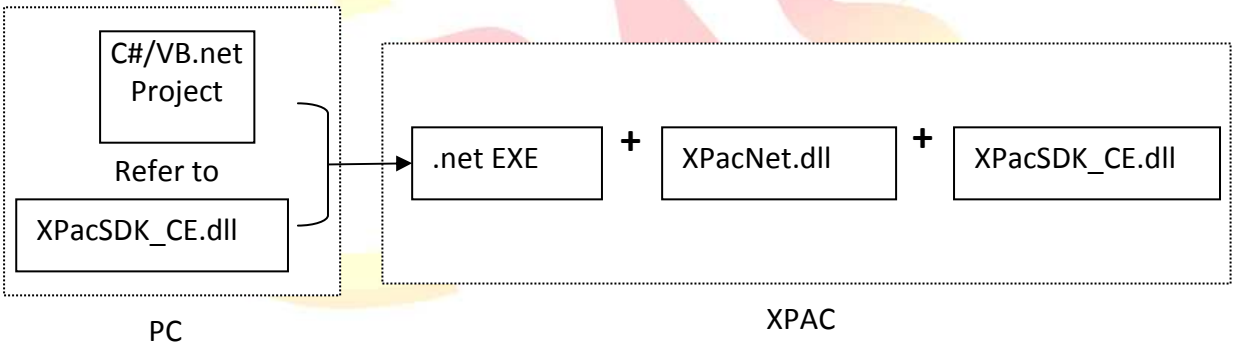
1. The users have used XPAC series devices and their programs is based on the old SDK (XPacSDK_CE.dll/XPacNet.dll) working with the old DCON modules (*Note 2*) on XPAC device and without using multi-function DCON modules (*Note 1*).

   The user's program can continue to use the old library without needing to be modified. (The Old SDK will continue to maintain (Fix the bugs) and released regularly, but will not add new features)

   Use the old SDK as following flowchart:
   The VC project required to link XPacSDK_CE.lib while building, and the built executable file placed in the XPAC series device must work with XPacSDK_CE.dll



   The C#/VB.net project required to refer to XPacNet.dll while building, and the built executable file placed in the XPAC series device must work with XPacNet.dll and XPacSDK_CE.dll.

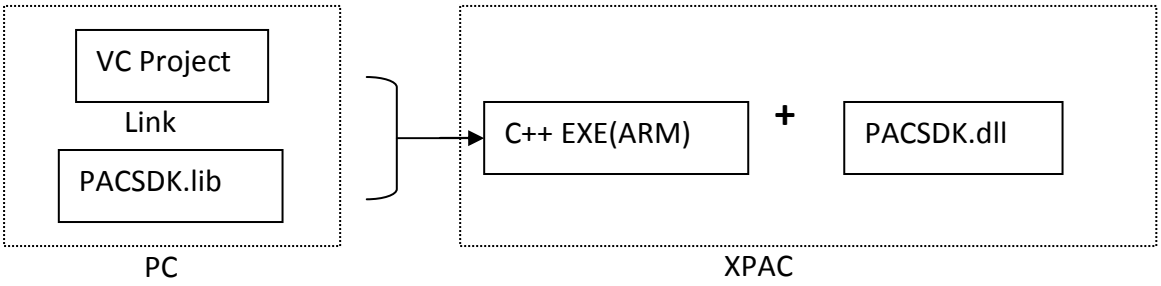


2. The users have used XPAC series devices and their programs is based on the old SDK (XPacSDK_CE.dll) working with the old DCON modules and multi-function DCON modules on XPAC device. The new PACSDK.dll provides pac_xxx_MF API functions that allow access to Multi-function modules, **so the code must be updated in order to use the new PACSDK.dll in the program.** (Refer to How-to document, x6-10_how_to_update_to_pacsdk_library_from_xpacsdk_library_tc.pdf for more details)

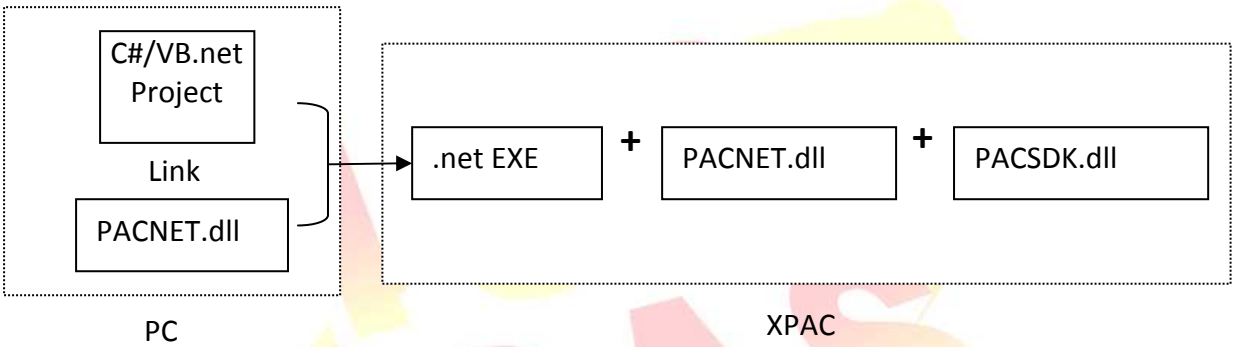
3. The users have never used XPAC series devices. Their program will be based on the new SDK working with an old DCON module or a Multi-function module. Our API Manual give instructions for the PACSDK.dll and the demo programs included on the shipped CD/FTP are linked with the new PACSDK.dll, so users should refer to the demo programs and follow the API instructions when developing new programs based on the new PACSDK.dll, rather than those for the XPACSDK.

LRI Distribuída no Brasil pela LRI | vendas@lri.com.br | 51 2129-3000

Use the new SDK as following flowchart:
The VC project required to link PACSDK.lib while building, and the built executable file placed in the XPAC series device must work with PACSDK.dll



The C#/VB.net project required to refer to PACNET.dll while building, and the built executable file placed in the XPAC series device must work with PACNET.dll and PACSDK.dll.



LRI Distribuída no Brasil pela LRI | vendas@lri.com.br | 51 2129-3000

Questions related to updating the PACSDK library from the WinPACSDK library and the solutions

Refer to links below to update the PACSDK library from WinPacSDK library.
http://ftp.icpdas.com/pub/cd/winpac/napdos/wp-8x4x_ce50/document/faq/sdk/w6-10_how_to_update_to_pacsdk_library_from_winpacsdk_library_en.pdf

Questions related to updating to PACSDK library from the XPACSDK library and solutions

Refer to links below to update the PACSDK library from XPacSDK library.
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

LRI Distribuída no Brasil pela LRI | vendas@lri.com.br | 51 2129-3000