How to write a VB.net application with XPAC SDK in visual studio 2008

Applies to:

Platform	OS version	XPAC utility version
XPAC series	All versions (WinCE6)	All versions

The XPAC SDKs provides a complete solution to integrate with XPAC and compatible with Visual C#, Visual Basic .net and C++. In order to use a component in your application, you must first add a reference to it.

Step 1 Get the XPacNet.dll and copy it to the project folder

The XPacNet.dll can be obtained from any VB.net demo program that has been provided on the CD or by downloading the latest version from ICP DAS web site. You can obtain by following list.
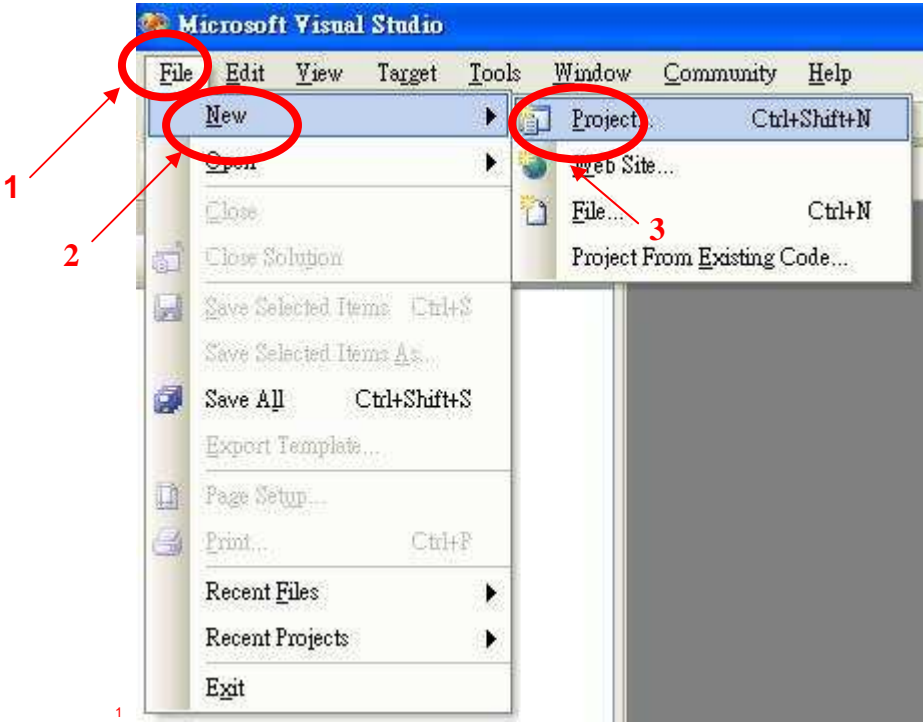
1. CD:\SDK\XPacNET

2. ftp://ftp.icpdas.com/pub/cd/xp-8000-ce6/sdk/xpacnet/

3. ftp://ftp.icpdas.com/pub/cd/XP-8000-Atom-CE6/SDK/XPacNet/

Step 2 Create a new project

Click the File ->New->Project in order as following.

Step 3 Perform the following steps in order

Select "Visual Basic" → "Smart Device" → "Smart Device Project" and input your program name.
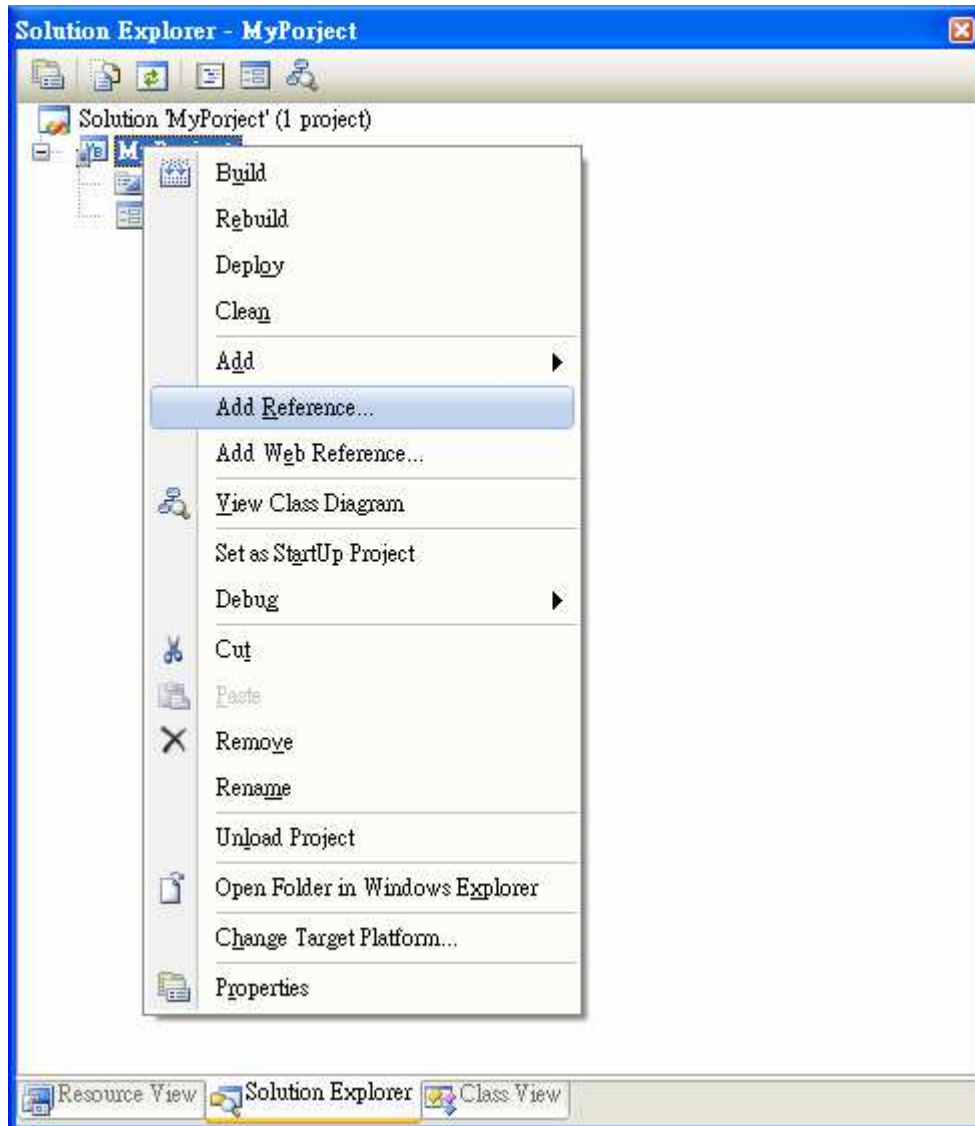
Step 4 Perform the following steps in order

Select "Windows CE" → Select ".NET Compact Framework Version 3.5" → Select "Device Application" and click "OK" button.
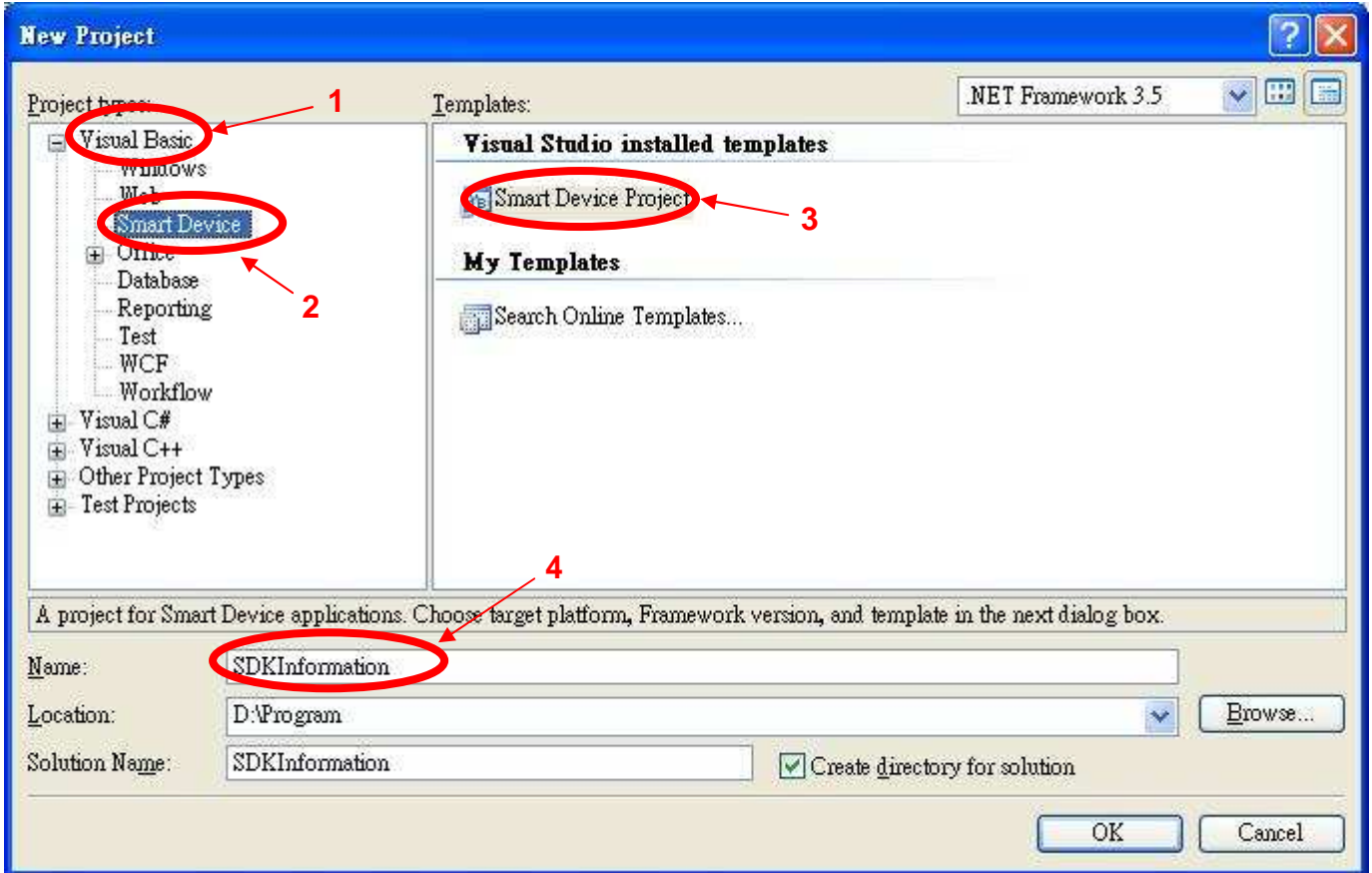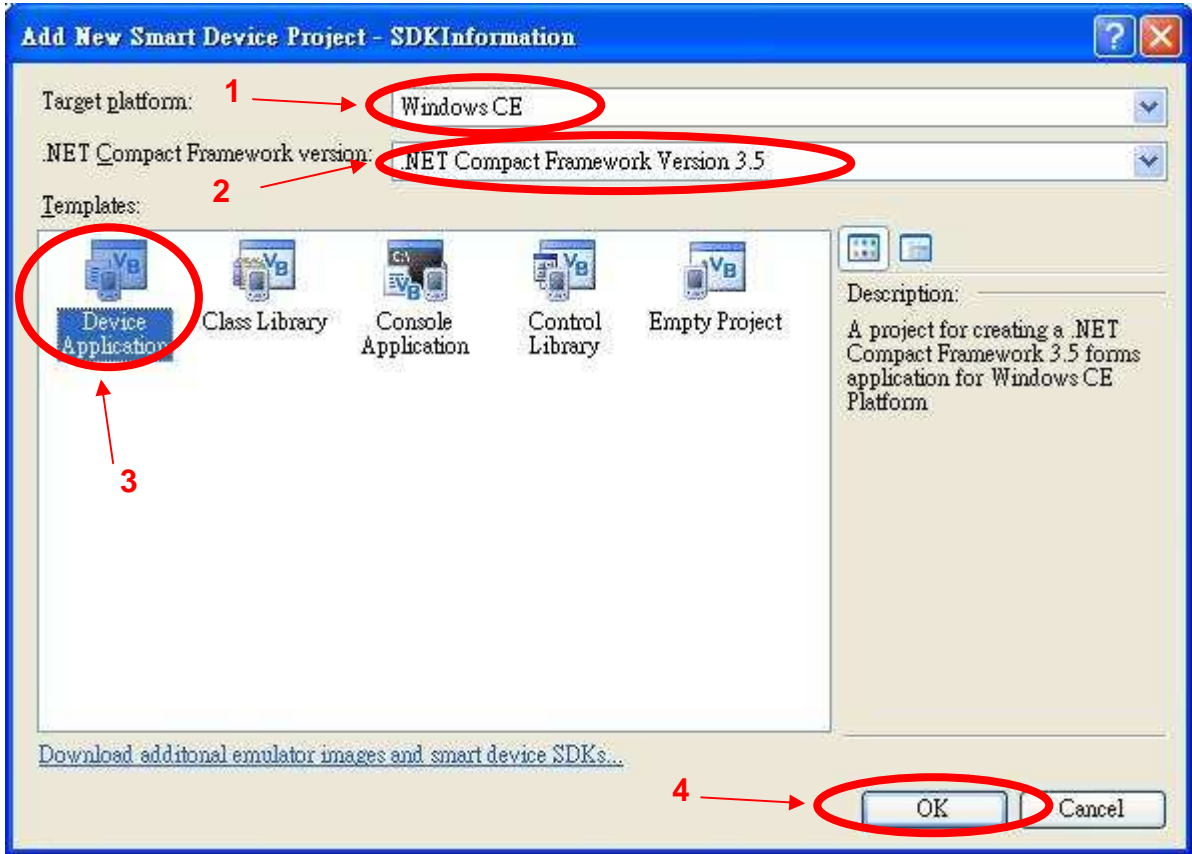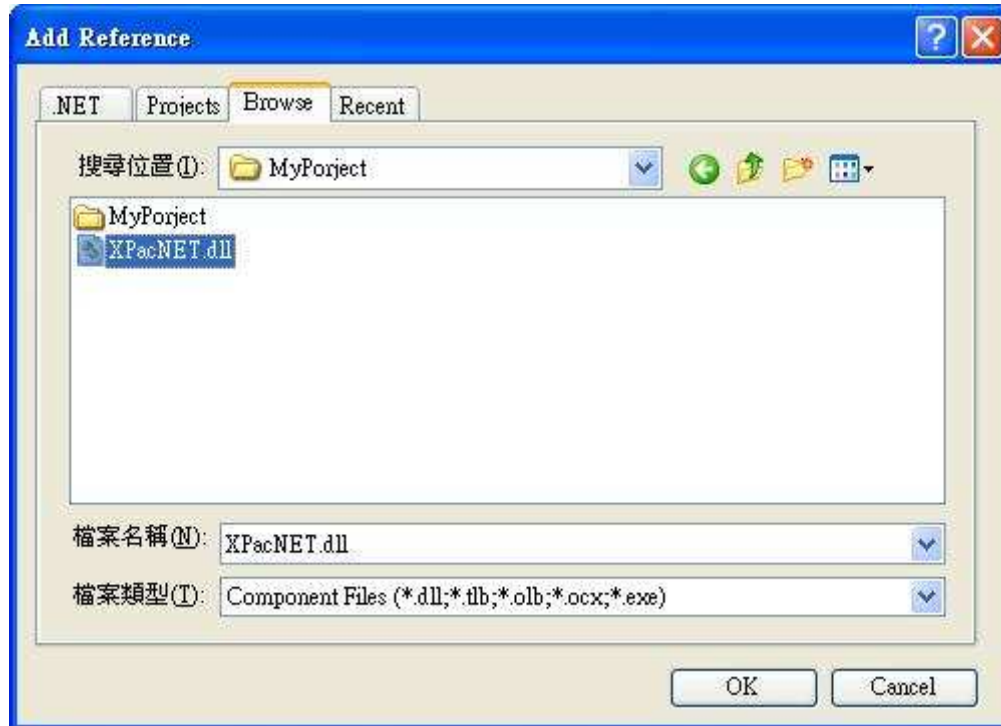
Classification	Development FAQ on XPAC					No.	5-032-00
Author	WeiKai	Version	1.0.0	Date	2011/4/11	Page	5/12

Step 5 **In Solution Explorer, right-click the References node, and then click Add Reference….**

Step 6 Select browse and add the XPacNet.dll

Adding the Control to the Form

You can drag various controls from the Toolbox onto the form. These controls are not really "live"; they are just images that are convenient to move around on the form into a precise location.
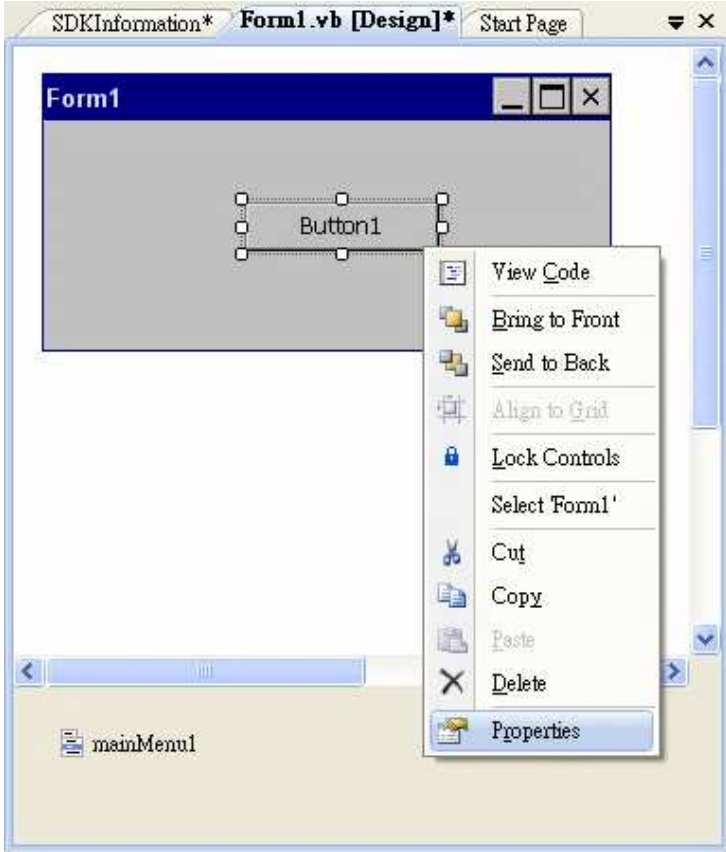
After you add a control to your form, you can use the Properties window to set its properties, such as background color and default text. The values that you specify in the Properties window are the initial values that will be assigned to that property when the control is created at run time.

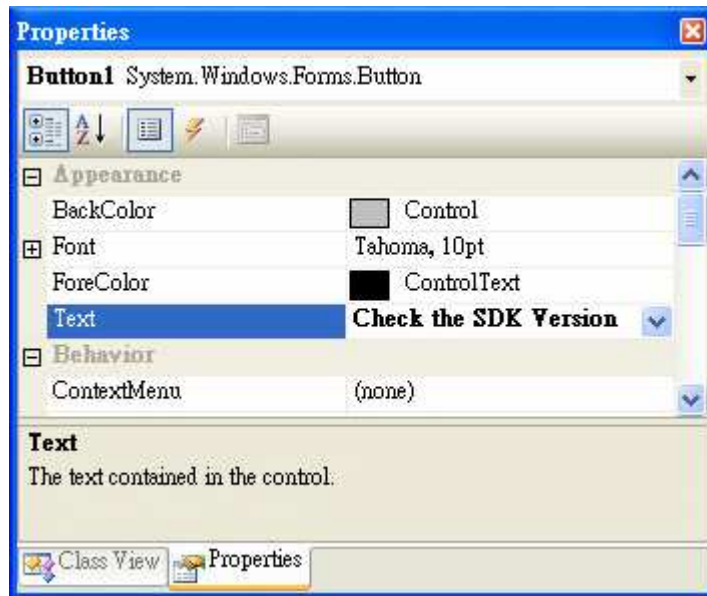## Step 7 From the Toolbox, drag a Button control onto the form

Step 8 Right-click the Button control, and then click Properties

Step 9 In the Properties window, type Check the SDK version, and press ENTER to set the Text property

Adding the Event Handling to the Control

You have finished the design stage of your application and are at the point when you can start adding some code to provide the program's functionality.

Step 10 Double-click the button on the form
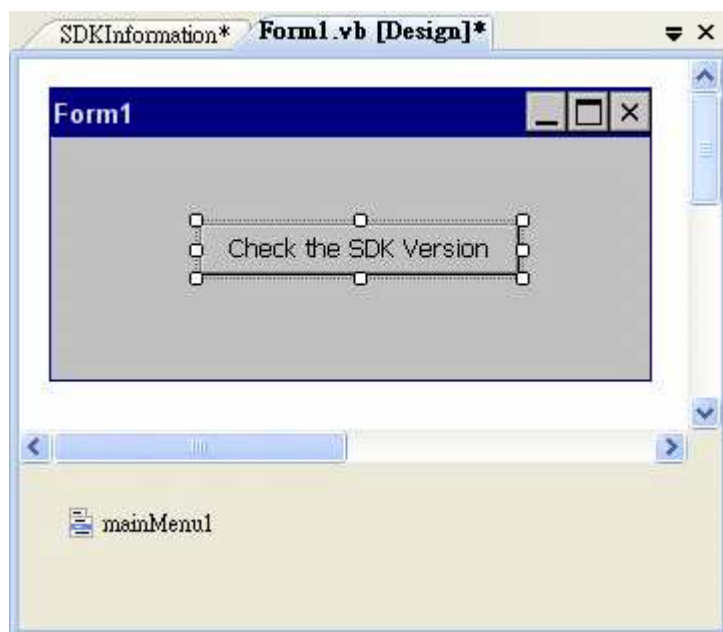
Step 11 Inserting the following code

MessageBox.Show(XPacNET.XPac.pac_GetXPacSDKVersion())

```
Public Class Form1

    Private Sub Button1_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles Button1.Click
        MessageBox.Show(XPacNET.XPac.pac_GetXPacSDKVersion())
    End Sub
End Class
```
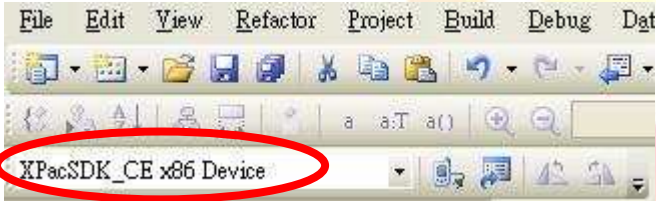
Uploading the Application to XPAC

XPAC series provide FTP server service and you can upload files to XPAC or download files from a public FTP server.
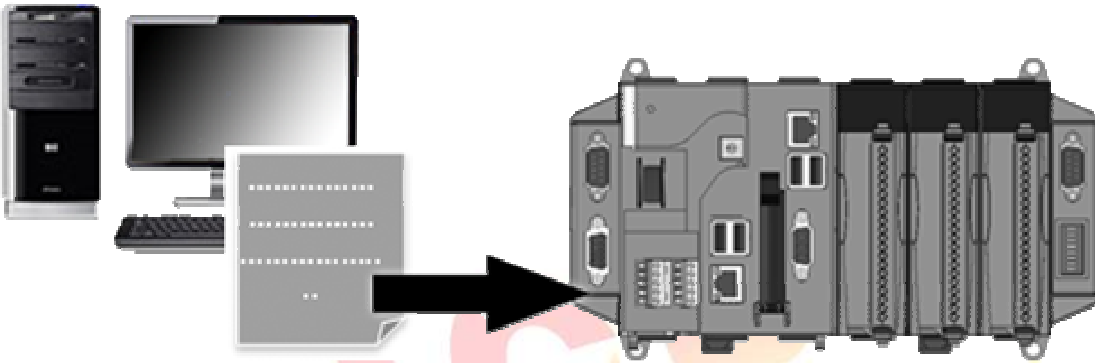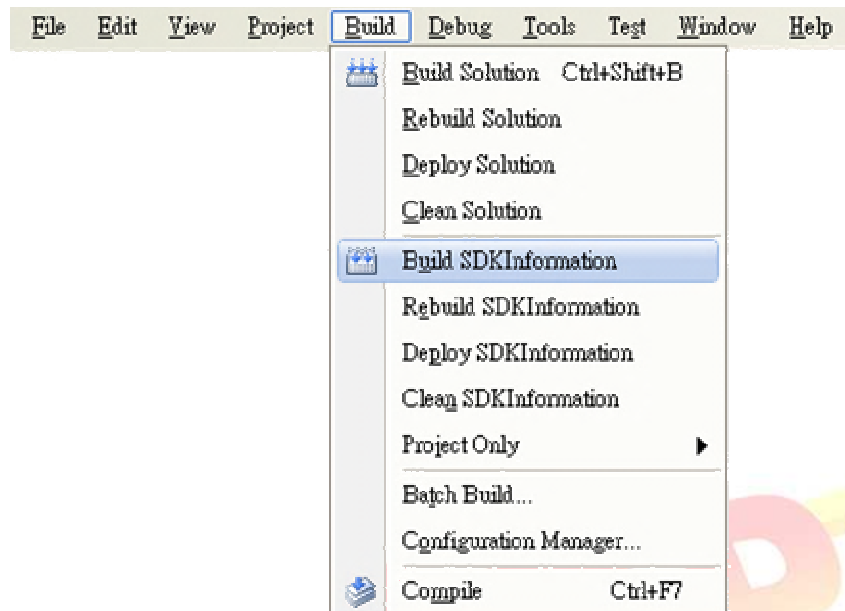
Step 12 **Select Platform device**

Select "XPacSDK_CE x86 Device"

Step 13 On the Build menu, click Build SDKInformation

File   Edit   View   Project   **Build**   Debug   Tools   Test   Window   Help

- Build Solution   Ctrl+Shift+B
- Rebuild Solution
- Deploy Solution
- Clean Solution
- **Build SDKInformation**
- Rebuild SDKInformation
- Deploy SDKInformation
- Clean SDKInformation
- Project Only   ▶
- Batch Build...
- Configuration Manager...
- Compile   Ctrl+F7

Step 14 Open your ftp software and type the IP address of XPAC.

Step 15 Upload the SDKInformation.exe application and the corresponding
XPacSDK_CE.dll file to the XPAC

Tips & Warnings

⚠ For applications programming in C# and VB.net with .net framework, when executing these application on the XPAC, the corresponding "XPacSDK_CE.dll" and "XPacNET.dll" file must be in the same directory as the .exe file.

File   Edit   View   Go

SDKInformation   XPacNET   XPacSDK_CE