How to write a MFC 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.
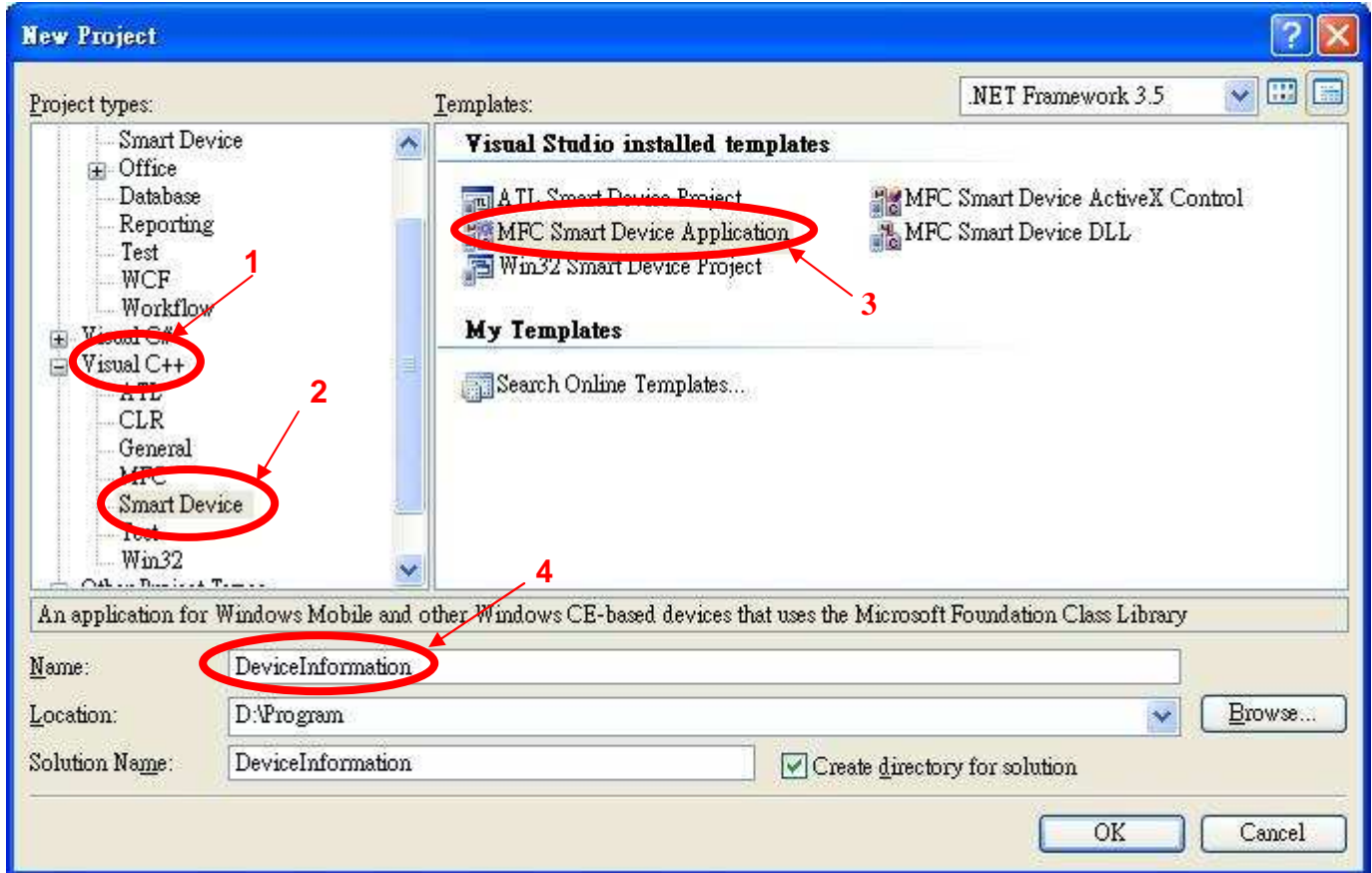
Step 1 Create a new project

Click the File → New → Project in order as following.
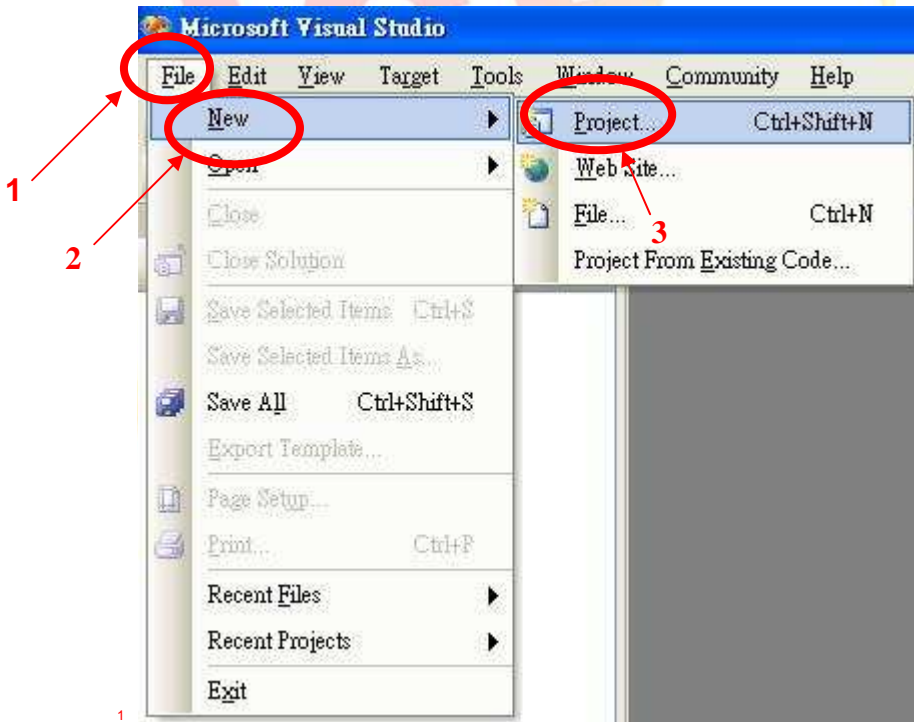
Step 2 Perform the following steps in order

Select "Visual C++" → "Smart Device"→ "MFC Smart Device Application" and input your program name.

Step 3 Program Setting

Click "Next"
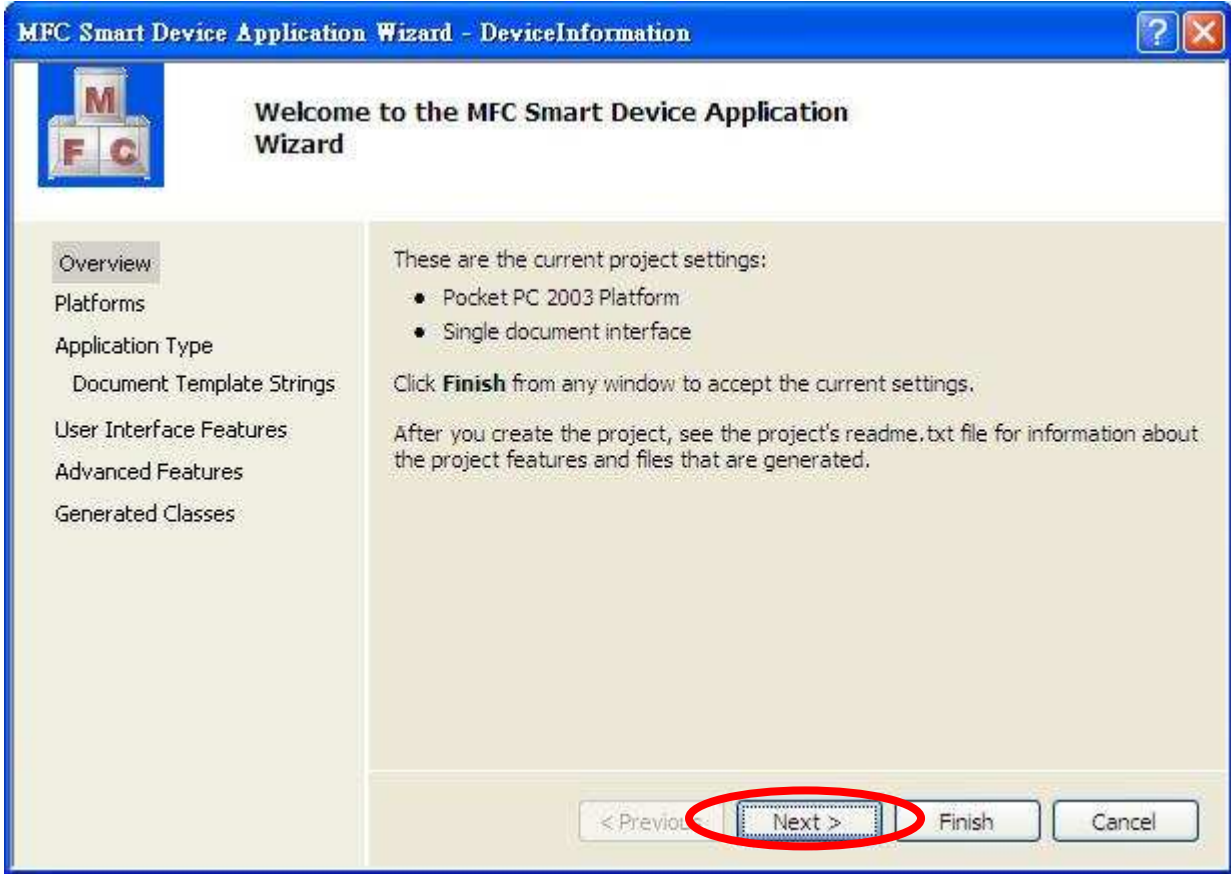
Step 4 Select SDK

Select "XPacSDK_CE" → Click ">" → Click "Next"
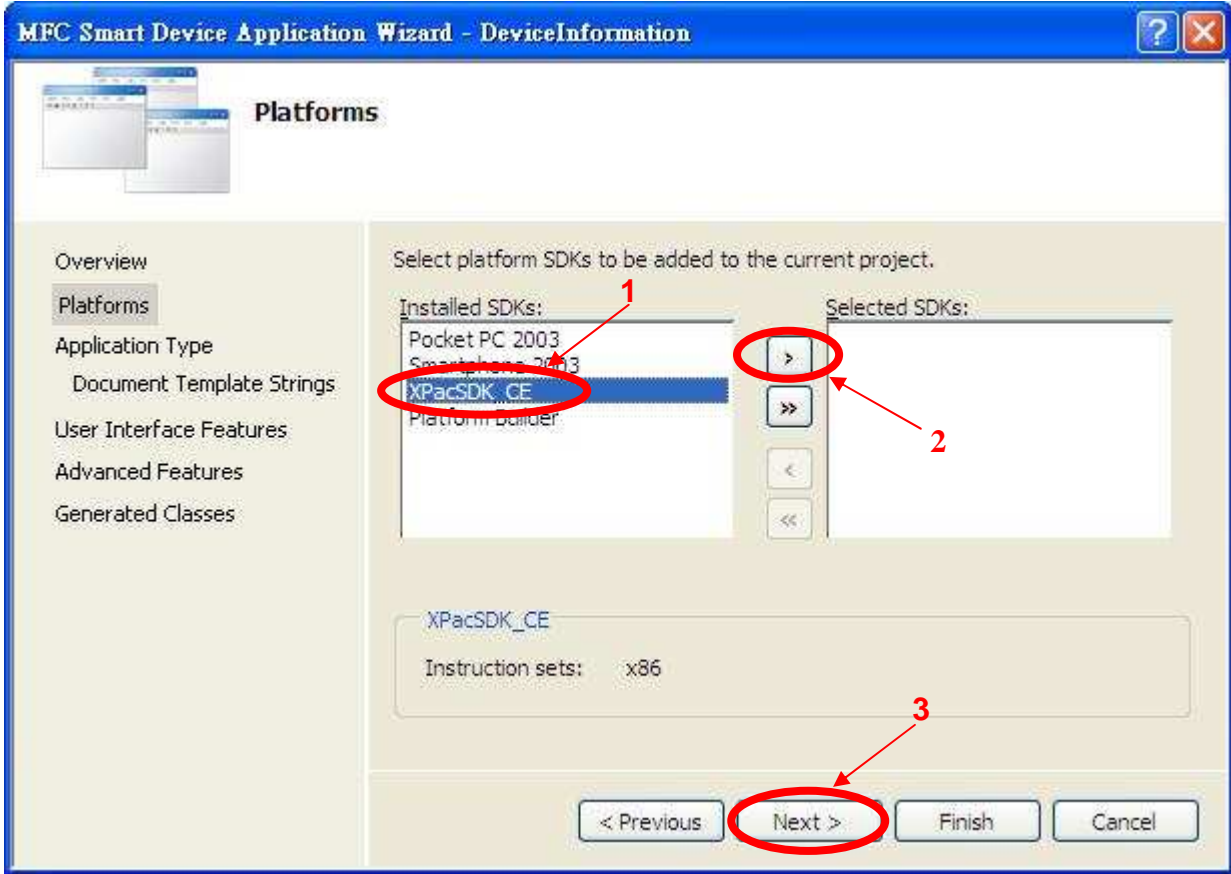
Step 5 Select Type

Select "Dialog based" → "Next"
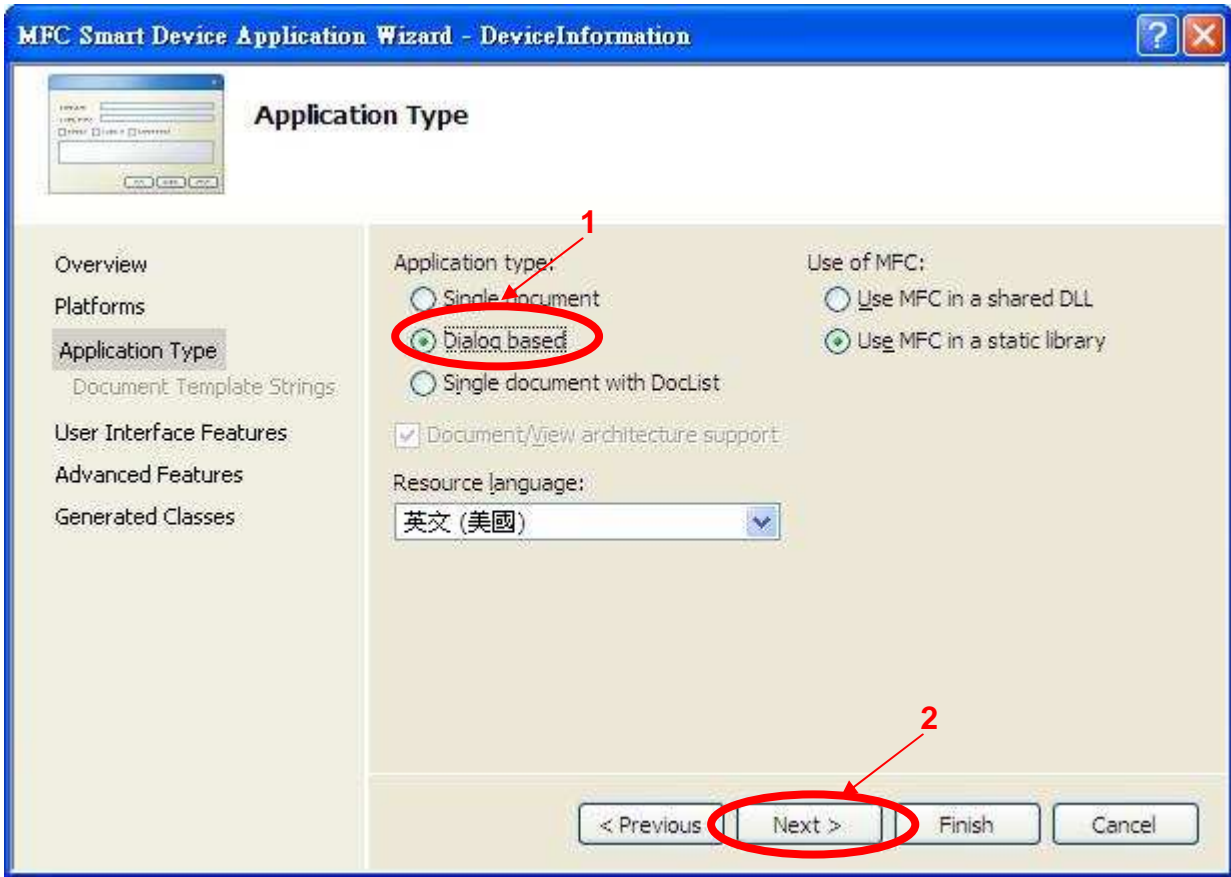
Step 6 Application interface

Click "Next"
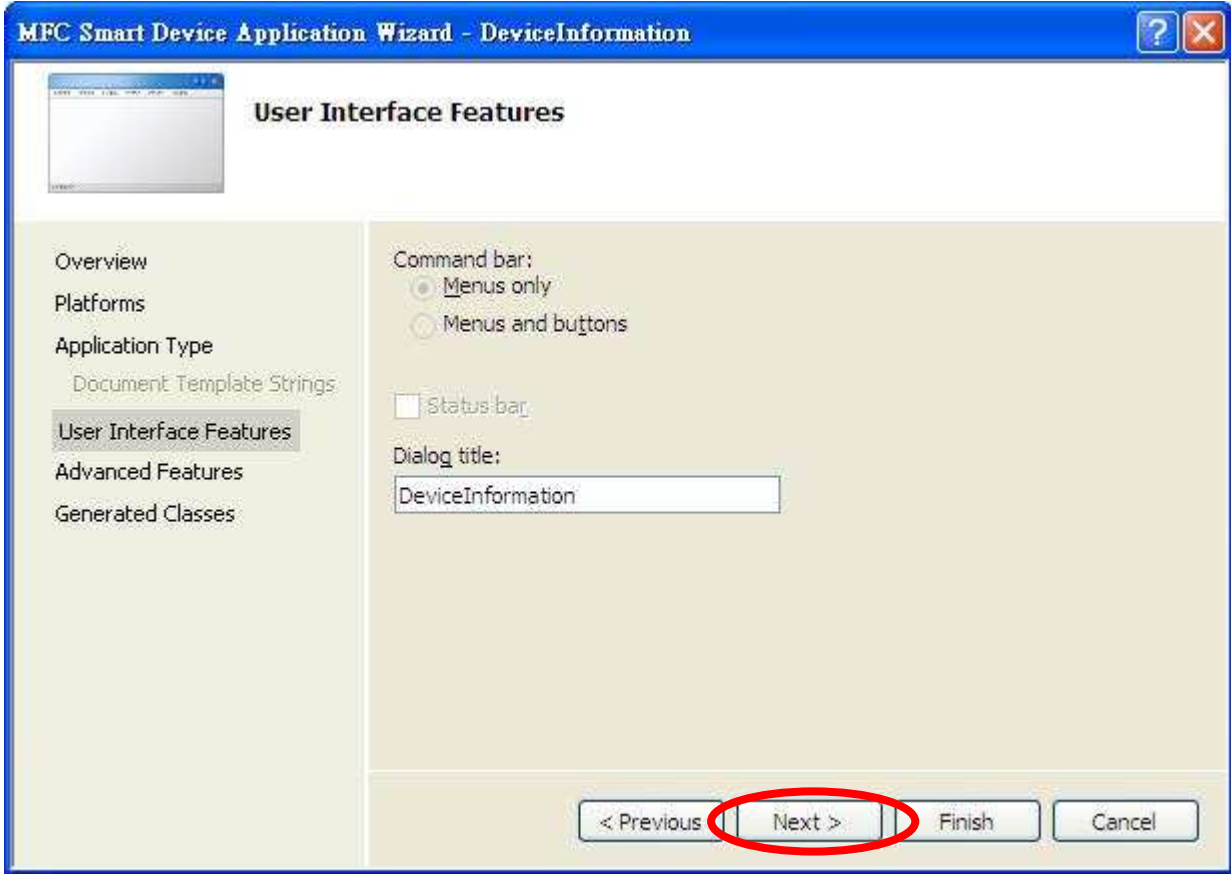
Step 7 Application Features

Click "Next", If your program have using network. You must select "Windows sockets".
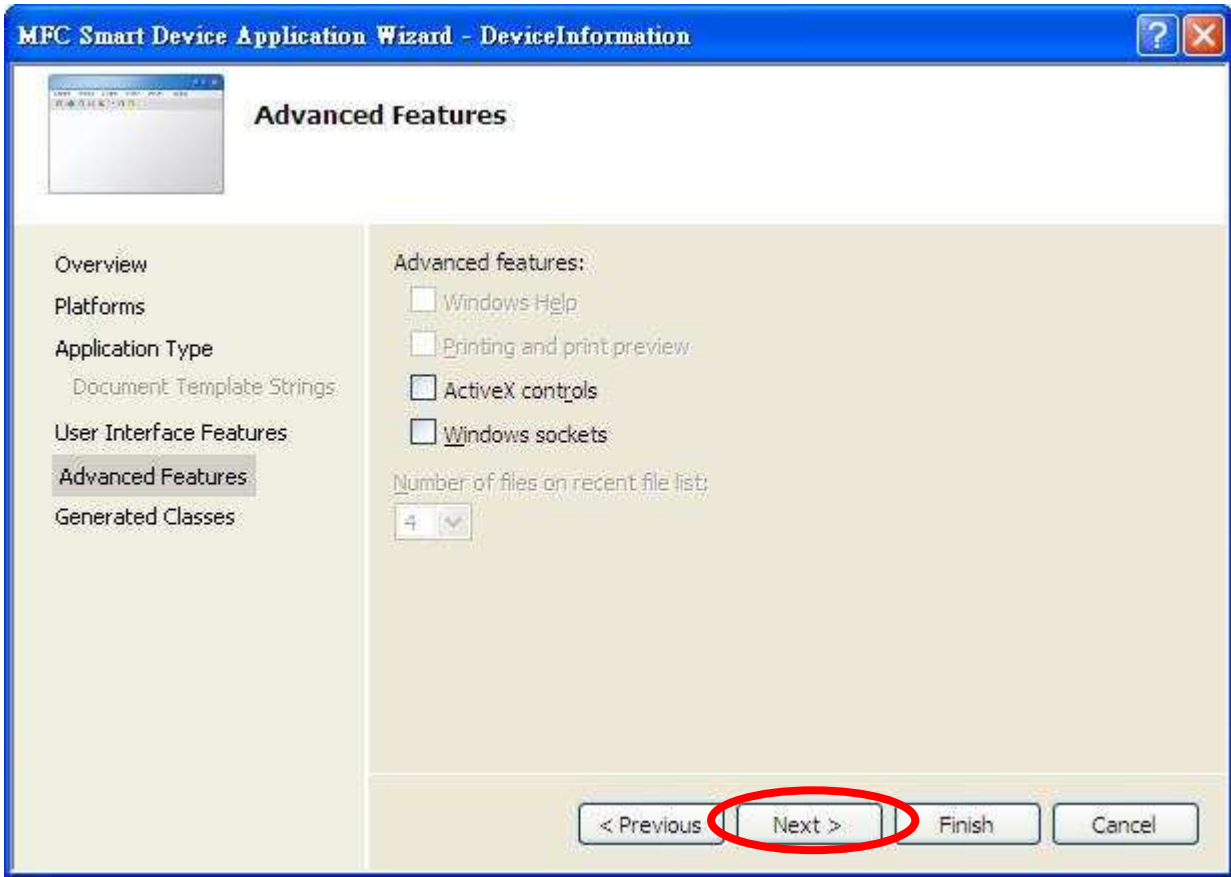
Step 8 Generated Classes

Click "Finish"
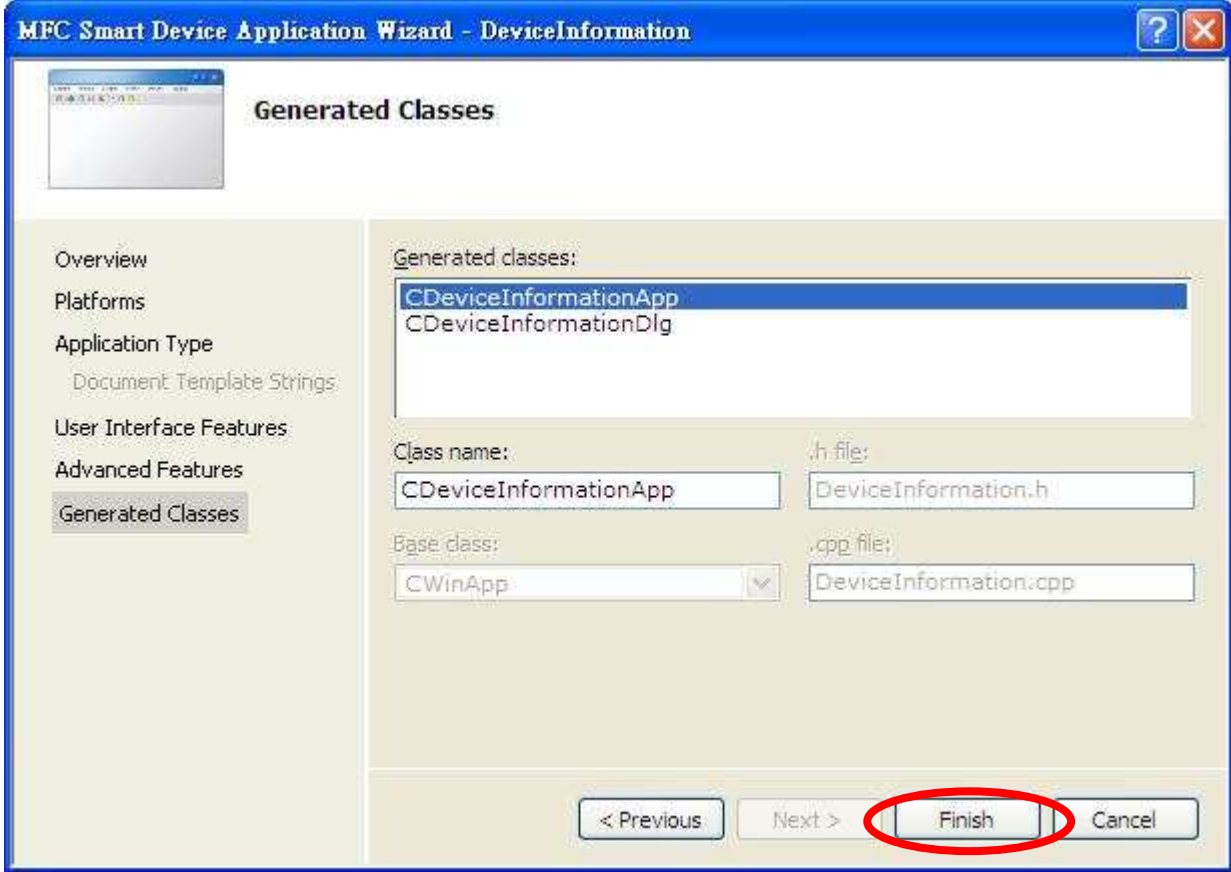
Step 9 Add "XPacSDK_ce.h"

Select "DeviceInformationDlg.cpp", and then add #include "XPacSDK_ce.h"
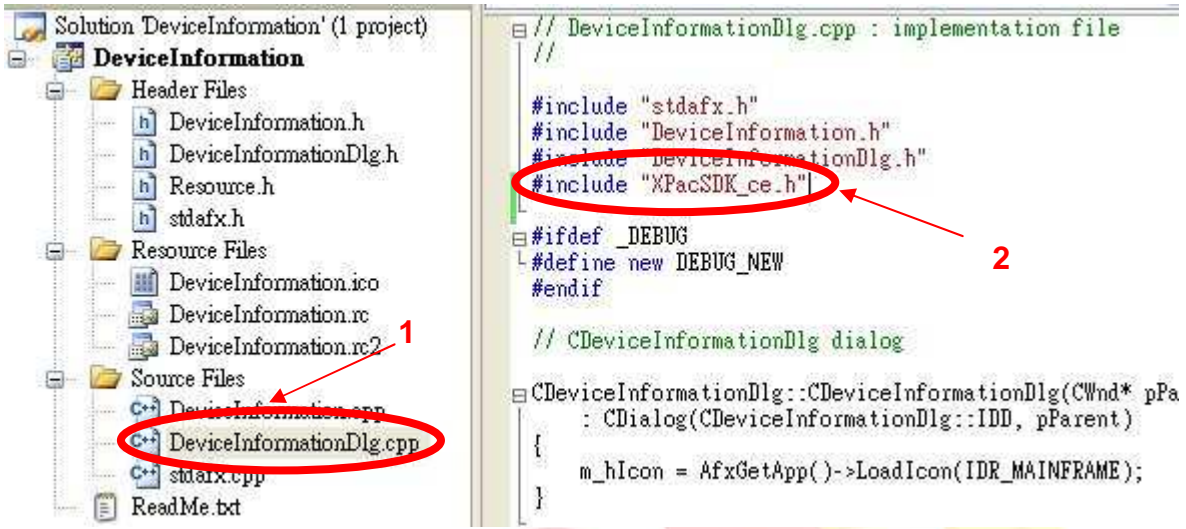
Adding the Control to the Form

You can drag various controls from the Toolbox onto the dialog. 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 1 Open Dialog

Click "Resource View" → "Dialog" → "IDD_DEVICEINFORMATION_DIALOG"
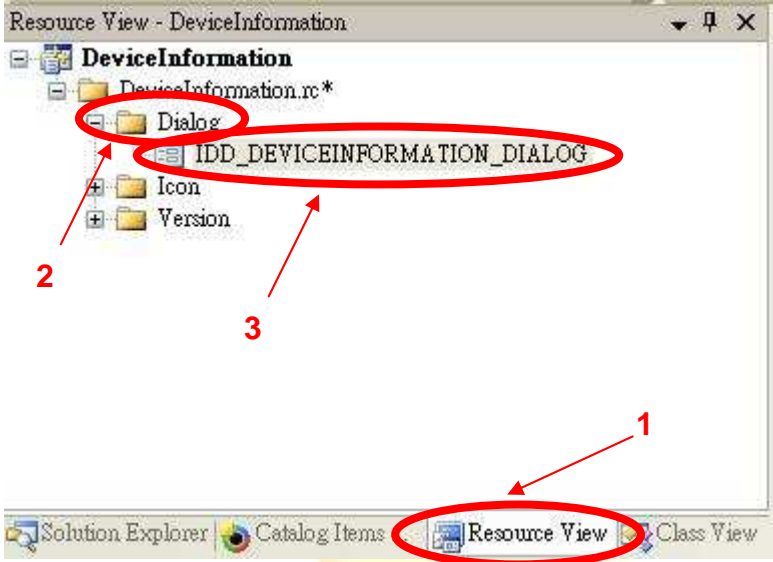
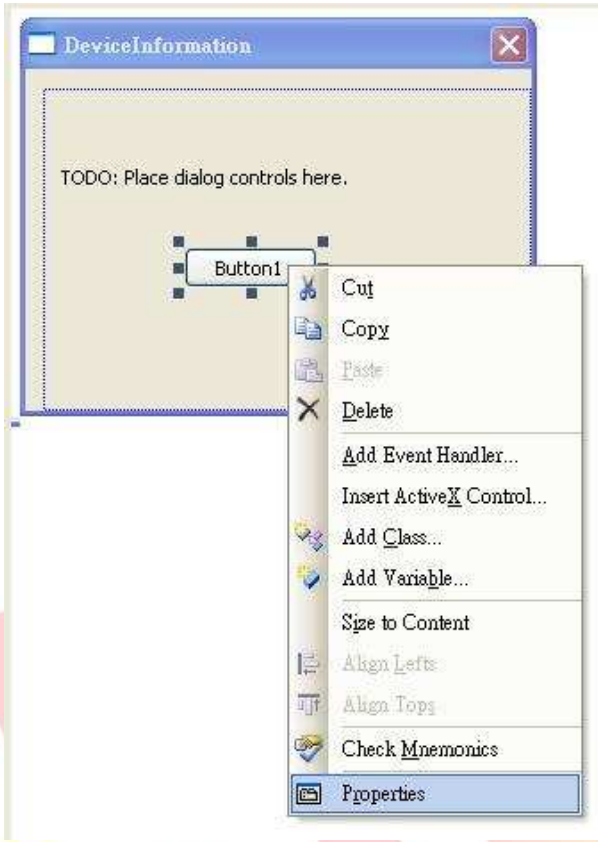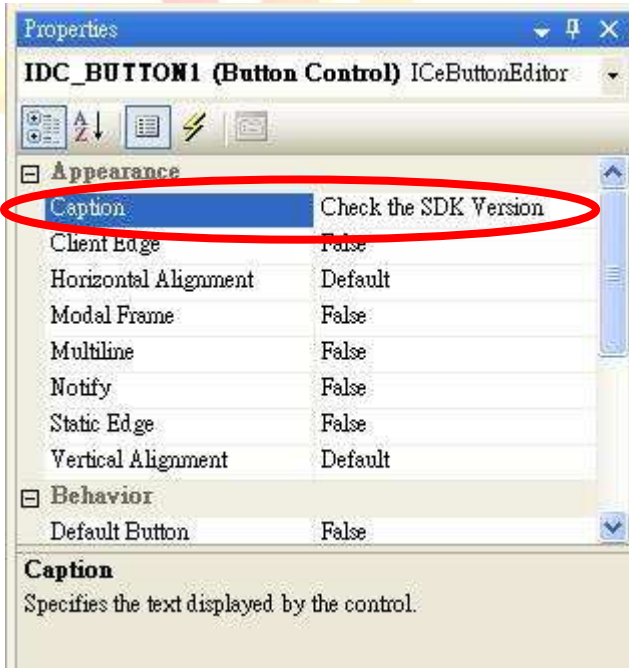## Step 2 From the Toolbox, drag a Button control onto the Dialog
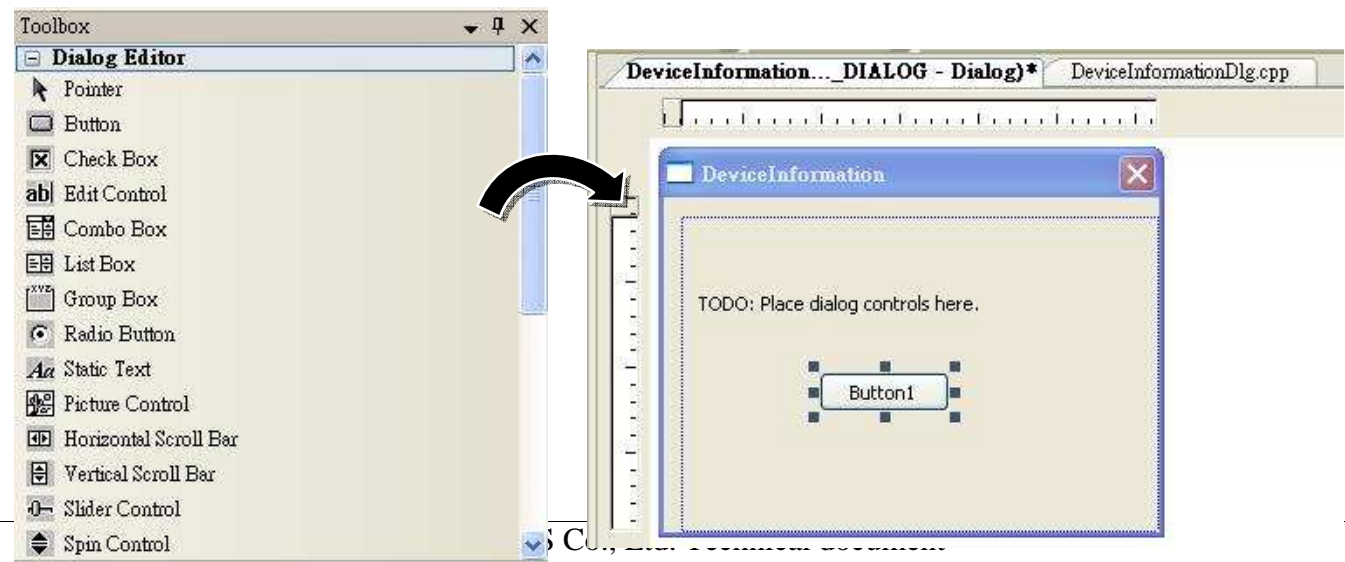
Step 3 Right-click the Button control, and then click Properties

Step 4 In the Properties window, type Check the SDK version, and press ENTER to set the Caption 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 5 Double-click the button on the Dialog

## Step 6 Inserting the following code

```
char temp[100];
TCHAR SDK[100];
pac_GetSDKVersion(temp);
wsprintf(SDK,L"%S",temp);

::MessageBox(NULL,SDK,NULL,MB_OK);
```

```
void CDeviceInformationDlg::OnBnClickedButton1()
{
    // TODO: Add your control notification handler code here
    char temp[100];
    TCHAR SDK[100];
    pac_GetSDKVersion(temp);
    wsprintf(SDK,L"%S",temp);
    ::MessageBox(NULL,SDK,NULL,MB_OK);

}
```
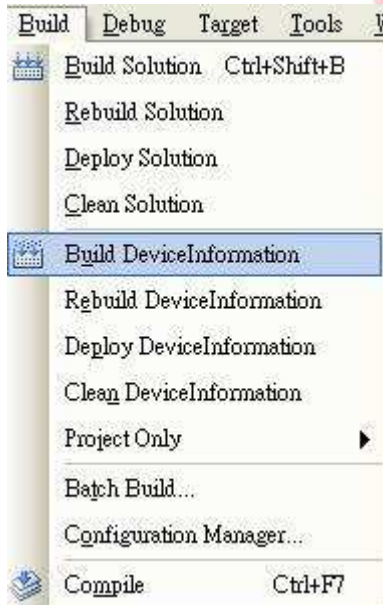
Uploading the Application to XPAC

XPAC series provide FTP server service and you can upload files to XPAC or download files from a public FTP server.
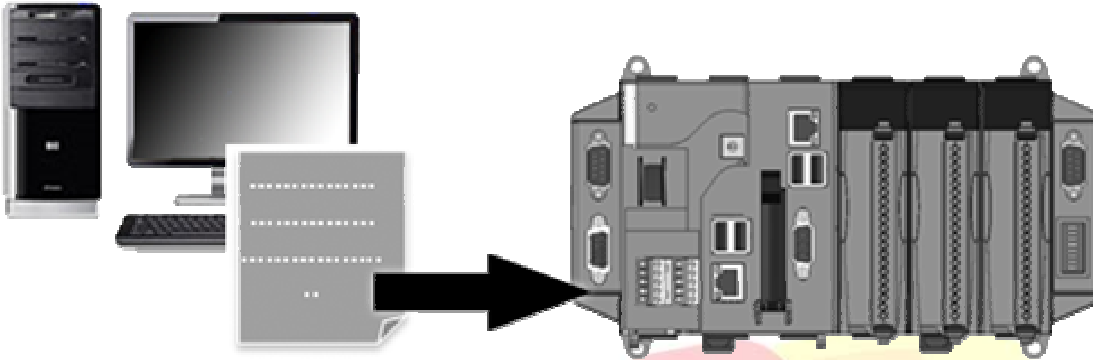
Step 7 **On the Build menu, click Build DeviceInformation**

Step 8 **Open your ftp software and type the IP address of XPAC.**

Step 9 **Upload the SDKInformation.exe application and the corresponding XPacSDK_CE.dll file to the XPAC**