XPAC Migration

 How to update a program to XP-8x3x-CE6

Written by Sean Hsu Last Modified Date: April 2017

How to update a program to XP-8x3x-CE6

Summary for updating a program to XP-8X3X-CE6

- The platform development SDK package installed on PC for developing the program for XP-8x3x-CE6 must be version 1.4.5 or later.
- The PACSDK.dll/PACSDK.lib used for XP-8x3x-CE6 must be version 4.4.0.1 or later.
- The VC program must be re-built with PACSDK.lib by Visual Studio.
- The C#/VB program aren't re-built by Visual Studio.

Platform Comparison

СРИ	LX800/Atom	R3600		
Models	XP-8x4x/ XP-8x4x-Atom	XP-8x3x		
OS Version	WinCE 6.0	WinCE 6.0		
Development Tools	VS2005/VS2008	VS2005/VS2008		
Embedded SQL Server	3.5	3.5		
.Net Compact Framework	3.5	3.5		

Development Tool Requirements

Ensure that the necessary tools and the corresponding PAC SDKs are installed in your computer. The table below provides a summary of the supported development tools and languages required for developing PAC applications.

Development To	Language	Visual Basic .Net	Visual C#	Visual C++
Visual Studio	Any version except Professional			
2005 or earlier	Professional	٧	٧	٧
Visual Studio 2008	Any version except Professional			
	Professional	٧	٧	٧
Visual Studio 2012 or later	Any version except Professional			
	Professional			

V: Supported, --: Unsupported



Visual Studio 2005/2008 Professional

Visual Studio takes full advantage of the .NET Compact Framework, which uses public Internet standards to enable integration with new and existing applications running on any platform. Supported languages include Visual C#, Visual C++ and Visual Basic .NET.

If you have an MSDN subscription, VS2005/VS2008 Pro can be downloaded from https://msdn.microsoft.com/en-us/subscriptions/securedownloads/hh442898 Otherwise, you will need to purchase the software from your regular distributor.

Installation Procedure for VS2005

Step 1: Download the necessary installation files and updates

- 1. Visual Studio 2005 Professional <u>https://www.dreamspark.com/Product/Product.aspx?productid=1</u>
- 2. Visual Studio 2005 Service Pack 1 <u>http://www.microsoft.com/en-us/download/details.aspx?id=10986</u>

Installation Procedure for VS2008

Step 1: Download the necessary installation files and updates

- Visual Studio 2008 Professional https://www.dreamspark.com/Product/Product.aspx?productid=1
- Visual Studio 2008 Service Pack 1
 <u>http://www.microsoft.com/en-us/download/details.aspx?id=10986</u>

Step 2: Install the platform development SDK package

The PACSDK_CE_X.X.X_VS2008 msi file contains the C header files, the C libraries and the software development kit used to develop applications for the PAC series on the PC.

The SDK package can be found in the CD:\SDK\PlatformSDK\ folder on the companion CD that was provided with the package, or the latest version can be downloaded from the ICP DAS web site. http://ftp.icpdas.com/pub/cd/xp-8x3x-ce6/sdk/platformsdk/

File name: PACSDK_CE_X.X.X_VS2008.msi, where X.X.X is the released version of the platform sdk. The platform sdk of version 1.4.5 or later to support XP-8X3X-CE6.

Step 3: Execute the PACSDK_CE_X.X.X_VS2008.msi file (for VS2008) or PACSDK_CE_X.X.X_VS2005.msi file (for VS2005)

Step 4: Follow the prompts until the installation process is complete

Creating a new project

The Visual C++ project template is a composite control that is used in this example to create a new project.

Step 1: Start Visual Studio 2008



Step 2: On the File menu, point to New, and then click Project to open the New Project dialog



Step 3: In the Project types pane, expand the Visual C++ node and then select Smart Device

Step 4: In the list of Templates, select the MFC Smart Device Application item

Step 5: Specify a name and a location for the application, and then click OK

New Project			? 🗙			
Project types:		<u>T</u> emplates:	.NET Framework 3.5 💌 🖽 🔚			
Visual Basic Visual C# ✓Visual C++ ··· A TL ··· CLR ··· General ··· MFC ··· Smart Device		Visual Studio installed templates ATL Smart Device Project MFC Smart Device Application Win32 Smart Device Project My Templates				
 Test Win32 Other Project Types Test Projects 		🗊 Search Online Templates				
An application for	Windows Mobile and c	other Windows CE-based devices that uses th	ne Microsoft Foundation Class Library			
<u>N</u> ame:	SDKInformation					
Location:	C:\Documents and Settings\Windows\My Documents\WM_Windows4 My Documents VB_Rows					
Solution:	Create new Solution					
Solution Na <u>m</u> e:	Solution Name: SDKInformation					
			OK Cancel			

Step 6: When the following windows appears, click Next

FC Smart Device Application	ı Wizard - Read_IO
Welcome F C Wizard	e to the MFC Smart Device Application
Overview Platforms Application Type Document Template Strings User Interface Features Advanced Features Generated Classes	 These are the current project settings: Pocket PC 2003 Platform Single document interface Click Finish from any window to accept the current settings. After you create the project, see the project's readme.txt file for information about the project features and files that are generated.
	< Previous Next > Finish Cancel

Step 7: On the Platforms screen, select the platform to be added to the project, XPacSDK_CE and then click Next

IFC Smart Device Application	ı Wizard - Read_IO	? 🔀
Platform	s	
Overview	Select platform SDKs to be added to the current project.	
Platforms		
Application Type	Installed SDKs: Selected SDKs:	
Document Template Strings	Pocket PC 2003	
Advanced Features		
Generated Classes		
	XPacSDK_CE	
	Instruction sets: x86	
	< Previous Next > Finish Ca	ancel

Step 8: On the Application Type screen, select the Dialog based option, and then click Next

MFC Smart Device Application	Wizard - Read_IO		? 🔀
Applicati	on Type		
Overview Platforms Application Type Document Template Strings User Interface Features Advanced Features Generated Classes	Application type: Single document Single document with DocList Document/ <u>Wew architecture support</u> Resource language:	Use of MFC: Use MFC in a shared DLL Use MFC in a static library Use MFC in a static library Ext > Finish Canc	:el

Step 9: Click Next on the User Interface Features screen

IFC Smart Device Application	Wizard - SDKInformation	? 🔀
User Inte	erface Features	
Platforms Application Type Document Template Strings User Interface Features Advanced Features Generated Classes	Menus and buttons Status bar Dialog title: SDKInformation	
	< Previous Next > Finish C	ancel

Step 10: Click Next on the Advanced Features screen

IFC Smart Device Application Wizard - Read_IO					
Advance	ed Features				
Overview Platforms Application Type Document Template Strings User Interface Features Advanced Features Generated Classes	Advanced features: Windows Hglp Printing and print preview ActiveX controls Windows sockets Number of files on recent file list: Image: Control of files on recent file list: <th>ncel</th>	ncel			

Step 11: Click Finish on the Generated Classed screen

Generate	ed Classes			
Overview	<u>G</u> enerated dasses:			
Platforms Application Type Document Template Strings	CSDKInformationApp CSDKInformationDlg			
User Interface Features	Class name	h filor		
Advanced Features	CSDKInformationApp	SDKInformation.h		
Generated Classes	Base dass:	.cpg file:		
	CWinApp	SDKInformation.cpp		
	< Previous	lext > Finish Cancel		

Configuring the Platform

When developing applications using Visual C++, the Platform must be configured to indicate which platform and device the application will be downloaded to. Before deploying the project, ensure that the platform is corrected.

On the Debug configuration toolbar, select the Release option and then select XPacSDK_CE (x86), as illustrated below.



Specifying the PAC SDK Libraries

The PACSDK libraries for XP-8000-CE6 series module are provided in the PACSDK, and are compatible with C++. In order to use a component in an VC application, It is must to link PACSDK.lib in the project first.

Step 1: On the View menu, click the Property Pages option



Step 2: In left pane of the Property Pages dialog, expand the Linker node, and then click Input

Step 3: In the right pane of the Property Pages dialog, enter PACSDK.lib in the

Additional Dependencies item

SDKInformation Property Pages	? 🗵
Configuration: Active(Debug)	Platform: Active(XPacSDK_CE (x86)) Configuration Manager
 Common Properties Configuration Properties General Deblogging Deployment C/C++ Linker General Input Manifest File Deblogging System Optimization Embedded IDL Advanced Command Line Resources XML Document Generator Browse Information Build Events Custom Build Step Authenticode Signing 	Additional Dependencies PACSDK lib Ignore All Default Libraries No Ignore Specific Library Module Definition File Add Module to Assembly Embed Managed Resource File Force Symbol References Delay Loaded DLLs Assembly Link Resource Module To Specific Library Force Symbol References Delay Loaded DLLs Assembly Link Resource Additional Dependencies Specifies additional items to add to the link line (ex: kernel32 lib); configuration specific.
	確定 取消 要用 (A)

Step 4: Click the OK button to exit the Property Pages dialog

Adding a control to the form

A variety of controls can be dragged from the Toolbox and placed onto the form. These controls, however, are not really "live"; they are simply images that can be used as a convenient method of moving an object to a precise location on the form.

After adding a control to the form, use the Properties dialog to set the parameter values for the control, such as the background color and the default text, etc. The values that are specified in the Properties dialog are the initial values that will be assigned to that property when the control is created at run time.

Step 1: On the View menu, click the Resource View option

File	Edit	View	Project	Build	Debug	Tools	Test	Wi	ndow	Help
		¥	Code			Ctrl	+Alt+0			
		-2	Solution E	xplorer		Ctrl	+Alt+L			
			Bookmark	Windo	w	Ctrl+K, C	Ctrl+W	'		
		<u> </u>	Class Viev	v		Ctrl+S	hift+C	:		
			Code Defi	nition V	Vindow	Ctrl+S	hift+V	'		
		<u> 1</u>	Object Bro	owser		Ctrl	+Alt+J			
			Output				Alt+2			
			Property I	Manage	er					
		2	Resource	View		Ctrl+S	Shift+E			
		R	Toolbox			Ctrl+	+Alt+X			
			Find Resu	lts				►		
			Other Windows				►			
			Toolbars					►		
			Full Scree	n	9	Shift+Alt	+Enter			
		E	Navigate	Backwa	rd		Ctrl+-			
		E,	Navigate	Forward	d	Ctrl+9	Shift+-			
			Next Task							
			Previous 1	Task						
		(CO)	Property P	Pages						

Step 2: In the Resource View Panel, expand the [Project name].rc file node, expand the Dialog node, and then click the plug-in dialog item

Resource View - SDK_Info 🛛 🛨 🗶	
⊡@ SDK_Info	
iaia SDK_Info.rc	
	SDK_Info.rc - IDD_S Dialog 🛛 🗸 🗙
	SDK_Info
	TODO: Place dialog controls here.
4 III •	
Resource 🔆 Toolbox 🗔 Property	

Step 3: Drag a Button control from the Toolbox panel onto the form



Step 4: On the Properties panel, type Check the SDK version in the Caption field

Properties 🔹 🕂 🗙		
IDC_BUTTON1 (Button Control) ICeButtonEc •		
2↓ 🔲 🖋 ⊨ 🖾		
Appearance		
	Caption	Check the SDK version
	Client Edge	False
	Horizontal Alignmer	Default
	Modal Frame	False
	Multiline	False
	Notify	False
	Static Edge	False
	Vertical Alignment	Default
Ξ	Behavior	
	Default Button	False
	Disabled	False
	Owner Draw	False
	Visible	True
Ξ	Misc	
	(Name)	IDC_BUTTON1 (Button C
horoup many more		

Step 5: In the Properties window, type Check the SDK version, and press ENTER to set the Text property



Adding the event handling for the control

Once the design stage for the application is complete, code can be added to provide the functionality for the program.





Step 2: Insert the following code

char sdk_version[32];

TCHAR buf[32];

pac_GetSDKVersion(sdk_version);

pac_AnsiToWideString(sdk_version, buf);

MessageBox(buf,0,MB_OK);



Step 3: Insert the following code into the header section

#include "PACSDK.h"

