

# iPPC-6731-WES7 User Manual (WES7 Based iPPC)

Version 1.0.2 January 2016

Service and usage information for

15" TFT LCD



iPPC-6731-WES7

Written by Mac  
Edited by Anna Huang

## Warranty

---

All products manufactured by ICP DAS are under warranty regarding defective materials for a period of one year, beginning from the date of delivery to the original purchaser.

## Warning

---

ICP DAS assumes no liability for any damage resulting from the use of this product. ICP DAS reserves the right to change this manual at any time without notice. The information furnished by ICP DAS is believed to be accurate and reliable. However, no responsibility is assumed by ICP DAS for its use, not for any infringements of patents or other rights of third parties resulting from its use.

## Copyright

---

Copyright © 2016 by ICP DAS Co., Ltd. All rights are reserved.

## Trademark

---

The names used for identification only may be registered trademarks of their respective companies.

### Contact US

If you have any problem, please feel free to contact us.

You can count on us for quick response.

Email: [service@icpdas.com](mailto:service@icpdas.com)

# Contents

Contents .....	3
1. Introduction.....	6
1.1. Features .....	7
1.2. Specifications .....	8
1.3. Overview .....	10
1.4. Dimensions.....	15
1.5. Companion CD.....	16
1.6. Rescue CF Card.....	17
2. Getting Started .....	18
2.1. Mounting the Hardware .....	19
2.1.1. Inserting the I/O Modules .....	20
2.1.2. Mounting the iPPC .....	22
2.1.3. Deploying a Basic iPPC System.....	25
2.1.4. Supplying Power to iPPC .....	26
2.2. Committing EWF to Allow Settings to Be Saved .....	28
2.3. Changing the User Interface Language.....	30
2.4. Using DCON Utility Pro to Configure I/O Modules.....	33
3. Security and Risk.....	34
3.1. Creating and Managing User Accounts.....	35
3.2. Turning Firewall On or Off .....	41
3.3. Configuring the FTP Server .....	43
3.4. Configuring the EWF Manager .....	48
4. Tools and Tasks .....	54
4.1. PAC Utility .....	55
4.1.1. System Information .....	55
4.1.2. Auto Execution .....	56
4.1.3. EWF Operation.....	57
4.1.4. Multi-serial Port Module.....	58
4.1.5. Language Setting .....	59
4.2. DCON Utility Pro .....	60
5. Your First iPPC Program .....	61

5.1. Your First iPPC Program in VB.NET .....	62
5.1.1. Create a New Project .....	63
5.1.2. Specify the Path of PAC Reference .....	65
5.1.3. Add the Control to the Form .....	67
5.1.4. Add the Event Handling for the Control .....	69
5.1.5. Upload the Application to iPPC .....	71
5.1.6. Execute the Application on iPPC.....	73
5.2. Your First iPPC Program in C#.....	74
5.2.1. Create a New Project .....	75
5.2.2. Specify the Path of PAC Reference .....	77
5.2.3. Add the Control to the Form .....	79
5.2.4. Add the Event Handling for the Control .....	81
5.2.5. Upload the Application to iPPC .....	83
5.2.6. Execute the Application on iPPC.....	85
5.3. Your First iPPC Program in Visual C++.....	86
5.3.1. Create a New Project .....	87
5.3.2. Specify the Path of the PAC Reference .....	90
5.3.3. Add the Control to the Form .....	95
5.3.4. Add the Event Handling for the Control .....	97
5.3.5. Upload the Application to iPPC .....	99
5.3.6. Execute the Application on iPPC.....	100
6. I/O Modules and SDK Selection .....	101
7. APIs and Samples .....	106
7.1. PAC Standard APIs .....	107
7.1.1. VB.NET Samples for PAC Standard APIs .....	108
7.1.2. C# Samples for PAC Standard APIs .....	109
7.1.3. Visual C++ Samples for PAC Standard APIs .....	110
7.2. PAC IO APIs.....	111
7.2.1. VB.NET Samples for PAC IO APIs.....	112
7.2.2. C# Samples for PAC IO APIs.....	114
7.2.3. Visual C++ Samples for PAC IO APIs .....	116
8. Restore and Recovery .....	118
8.1. Recovering the iPPC .....	119
8.2. Restoring the Rescue CF Card .....	122
9. iPPC Updates .....	126
9.1. Updating the iPPC OS .....	127

9.2. Updating the iPPC SDK.....	128
9.3. Updating the PAC Utility .....	130
10. iPPC Download Center .....	132
Appendix .....	133
A. I-8K Modules and I-87K Modules .....	133
B. Revision History .....	134

# 1. Introduction

This chapter provides an overview of the iPPC-6731-WES7 and its components, and introduces the fundamental concepts for user familiar with the iPPC-6731-WES7.



The iPPC-6731-WES7 are WES7 based iPPC that combine computing, I/O, and operator interface into a single unit, and provide the perfect solution for integrating HMI, data acquisition and control in an individual PAC. It is equipped with an Intel Atom E3827 CPU, three I/O expansion slots, TFT LCD and a variety of connectives including dual Gigabit Ethernet, USB port, RS-232 and RS-485 interface. The operating system is pre-installed in the built-in mSATA SSD, and the storage can be expanded from a Compact Flash slot. Local I/O slots are available to use our I-8K and I-87K series I/O modules and remote I/O expansion is available to use our Ethernet I/O modules and RS-485 I/O modules. Designed for panel mount installation, the front panel is NEMA 4/IP65 rated and can withstand sprayed water, humidity and extreme dust. Designed to operate over a wide  $-10^{\circ}\text{C} \sim 60^{\circ}\text{C}$  ambient temperature range, the fan less design offers the ultimate in reliability with no moving parts.

Since WES7 has the same Win32 API as Windows 7, most popular applications on desktop can run on WES7 based iPPC.

# 1.1. Features

The iPPC-6731-WES7 offers the most comprehensive configuration and remote system upgrade solutions to meet specific application requirements. The following list shows the hardware and software features designed to simplify installation, configuration and application.

- 15" LCD with Touch Panel
- E3827 (1.75 GHz) CPU
- WES7 (Windows Embedded Standard 7)
- Support eLogger HMI (free)
- 3 I/O Expansion Slots
- NEMA 4/IP65 Compliant Front Panel
- Ultra-Rugged Construction and Reliable Design
- Operating Temperature : -10 °C to +60 °C

## 1.2. Specifications

The table below summarizes the specifications of the iPPC-6731-WES7.

System Software	
OS	Microsoft Windows Embedded Standard 7 SP1
Framework Support	.Net Framework 3.5 SP1, 4.0, 4.5
Embedded Service	IE11, FTP Server, IIS 7.0, ASP (Java Script, VB Script)
SDK Provided	Dll for Visual Studio.Net
Multilanguage Support	English, German, French, Spanish, Portuguese, Russian, Italian, Korean, Japanese, Simplified Chinese, Traditional Chinese
CPU Module	
CPU	E3827 (1.75 GHz, 64-bit dual core)
SDRAM (DDR3)	2 GB
MRAM	128 KB
Flash (SSD)	mSATA slot with one 32 GB SSD
EEPROM	16 KB
Memory Expansion	CF slot with a 8 GB CF card (support up to 32 GB)
RTC (Real Time Clock)	Provide second, minute, hour, date, day of week, month, year
64-bit Hardware Serial Number	Yes, for software copy protection
Dual Watchdog Timers	Yes (0.8 second)
Rotary Switch	Yes (0 ~ 9)
Communication Ports	
Ethernet	2 x RJ-45 10/100/1000 Base-TX
USB 2.0	3
COM 2	RS-232 (RxD, TxD, CTS, RTS, DSR, DTR, CD, RI and GND); Non-isolated
COM 3	RS-485 (Data+, Data-, GND); 2500 V <sub>DC</sub> isolated

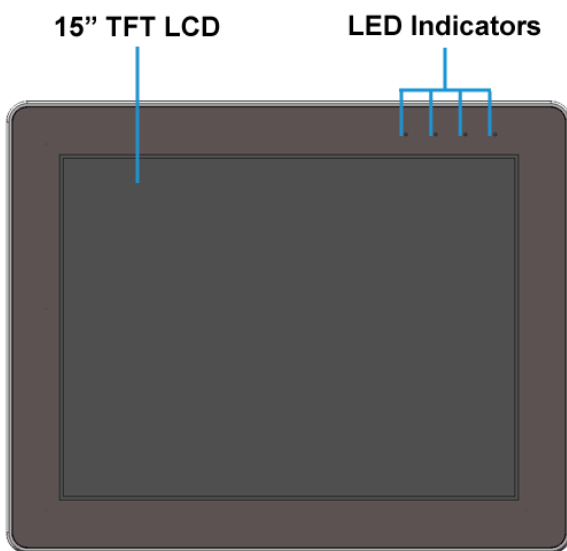


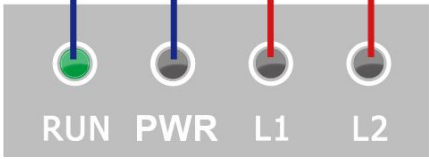
COM 4	RS-232/RS-485 (RxD, TxD, CTS, RTS and GND for RS-232, Data+ and Data- for RS-485); 2500 V <sub>DC</sub> isolated
Audio	Mic-in and Earphone-out
<b>MMI (Man Machine Interface)</b>	
LCD	15"
Resolution	1024 x 768
Brightness (cd/m <sup>2</sup> )	400
Contrast Ratio	500 : 1
LED Backlight Life (hrs)	50,000
Touch Panel	5-wire, resistive type; light transmission: 80 %
LED Indicator	4 (Run, PWR, L1, L2; L1 and L2 are user programmable)
<b>I/O Expansion Slots</b>	
Slot Number	3 (for high profile I-8K and I-87K series I/O modules)
<b>Mechanical</b>	
Dimensions (W x H x D)	381 mm x 305 mm x 88 mm
Panel Cut-Out (W x H)	366 mm x 290 mm, ±1 mm
Installation	Panel mounting
Ingress Protection	Front panel: NEMA 4/IP65
<b>Environmental</b>	
Operating Temperature	-10 °C ~ +60 °C
Storage Temperature	-20 °C ~ + 70 °C
Ambient Relative Humidity	10 % ~ 90 % RH (non-condensing)
<b>Power</b>	
Input Range	+10 V <sub>DC</sub> ~ +30 V <sub>DC</sub>
Redundant Power Inputs	Yes
Isolation	1 kV
Consumption	25 W

# 1.3. Overview

The iPPC-6731-WES7 is equipped with several interfaces and peripherals that can be integrated with external systems. Here is an overview of the components and its descriptions. The details of these items are as follows:

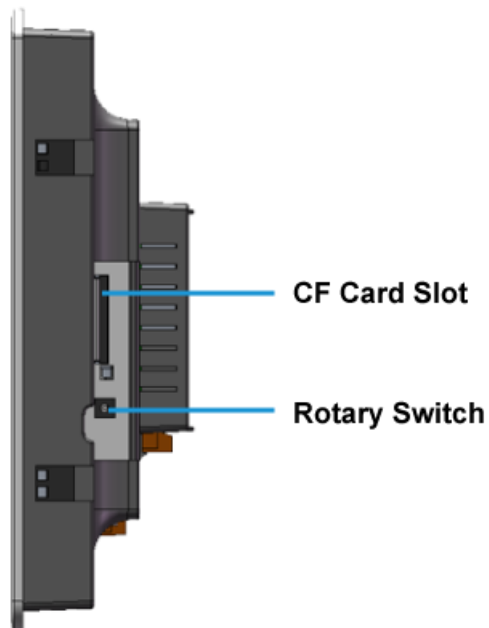
## Front View




Item	Description									
LED Indicators	<p data-bbox="515 1406 917 1440">User Programmable LED Indicators</p> <p data-bbox="515 1451 778 1485">System LED Indicators</p>  <p data-bbox="494 1713 1284 1803">The two system LED indicators display the status of the iPPC-6731-WES7. The details are shown as below.</p> <table border="1" data-bbox="494 1825 1284 1993"> <thead> <tr> <th data-bbox="494 1825 758 1881">LED Indicator</th> <th data-bbox="758 1825 1029 1881">Color (On state)</th> <th data-bbox="1029 1825 1284 1881">Meaning</th> </tr> </thead> <tbody> <tr> <td data-bbox="494 1881 758 1937">RUN</td> <td data-bbox="758 1881 1029 1937">Green</td> <td data-bbox="1029 1881 1284 1937">OS is running</td> </tr> <tr> <td data-bbox="494 1937 758 1993">PWR</td> <td data-bbox="758 1937 1029 1993">Red</td> <td data-bbox="1029 1937 1284 1993">Power is on</td> </tr> </tbody> </table>	LED Indicator	Color (On state)	Meaning	RUN	Green	OS is running	PWR	Red	Power is on
LED Indicator	Color (On state)	Meaning								
RUN	Green	OS is running								
PWR	Red	Power is on								

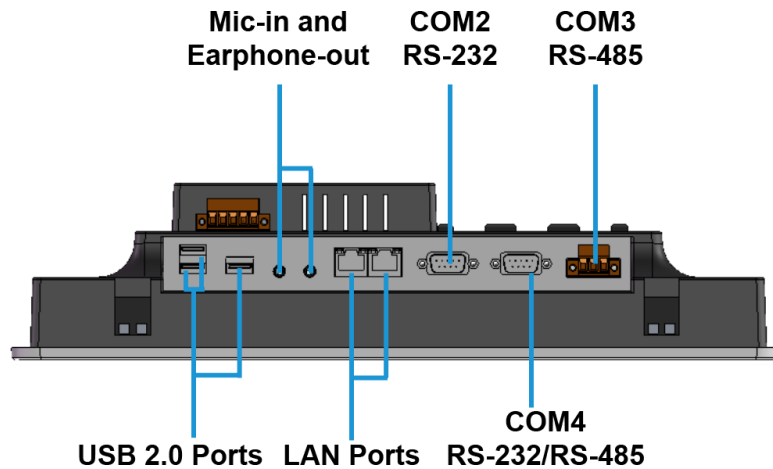
## Right Side View

---



Item	Description
CF Card Slot	The CF card can be used to restore the iPPC-6731-WES7 system and expand the memory up to 32 GB.
Rotary Switch 	Rotary Switch is an operating mode selector switch which provides seven functions related to the selection of the operating mode for iPPC-6731-WES7.

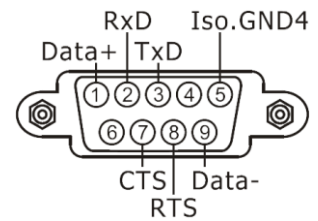
## Bottom View



Item	Description
USB 2.0 Port	The iPPC-6731-WES7 has three USB ports that allow support for the USB devices such as mouse, keyboard or an external USB hard drive.
Mic-in and Earphone-out	The iPPC-6731-WES7 has a microphone-in and an earphone-out that can be used to process the input and the output of sound.
LAN Port	The iPPC-6731-WES7 has two Ethernet ports that can be connected to a computer or device via an Ethernet cable. One port connects directly to the network or server, and the other port can be connected to another device server or another Ethernet device.
<b>COM2 (RS-232)</b>	
<p>The COM2 port is a 9-pins RS-232 connector. The details of the COM2 port specifications are shown to the side.</p>	
<p><b>Port Type:</b> Male</p> <p><b>Baud Rate:</b> 115200, 57600, 38400, 19200, 9600, 4800, 2400, 1200 bps</p> <p><b>Data Bits:</b> 5, 6, 7, 8</p> <p><b>Parity:</b> None, Even, Odd, Mark (Always 1), Space (Always 0)</p> <p><b>Stop Bits:</b> 1, 2</p> <p><b>FIFO:</b> 16 bytes</p>	

## COM4 (RS-232/RS-485)

The COM4 port is a 9-pins RS-232/RS-485 connector. The details of the COM4 port specifications are shown to the side.



**Port Type:** Male

**Baud Rate:** 115200, 57600, 38400, 19200, 9600, 4800, 2400, 1200 bps

**Data Bits:** 5, 6, 7, 8

**Parity:** None, Even, Odd, Mark (Always 1), Space (Always 0)

COM4 can be configured as either RS-232 or RS-485, that only can select one at a time and its configuration depends on the pin connections as follows:

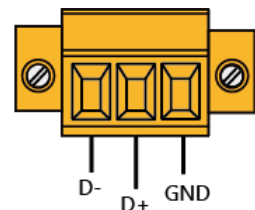
**RS-232** (RXD, TXD, CTS, RTS and GND)

**RS-485** (Data+ and Data-)

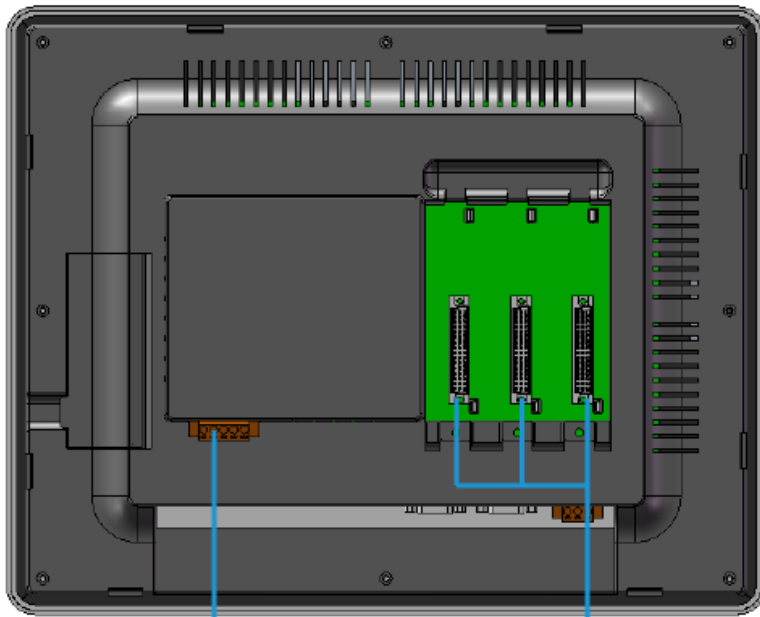
No software configuration or hardware jumper needed

## COM3 (3-wire RS-485)

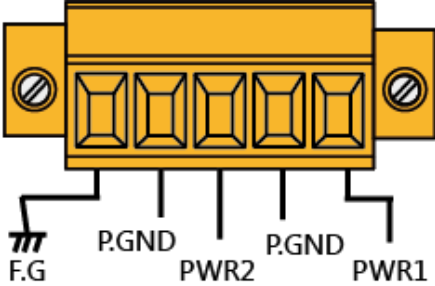
The COM3 port is a 3-wire RS-485 connector. The details of the COM3 specifications are shown to the side.



## Back View

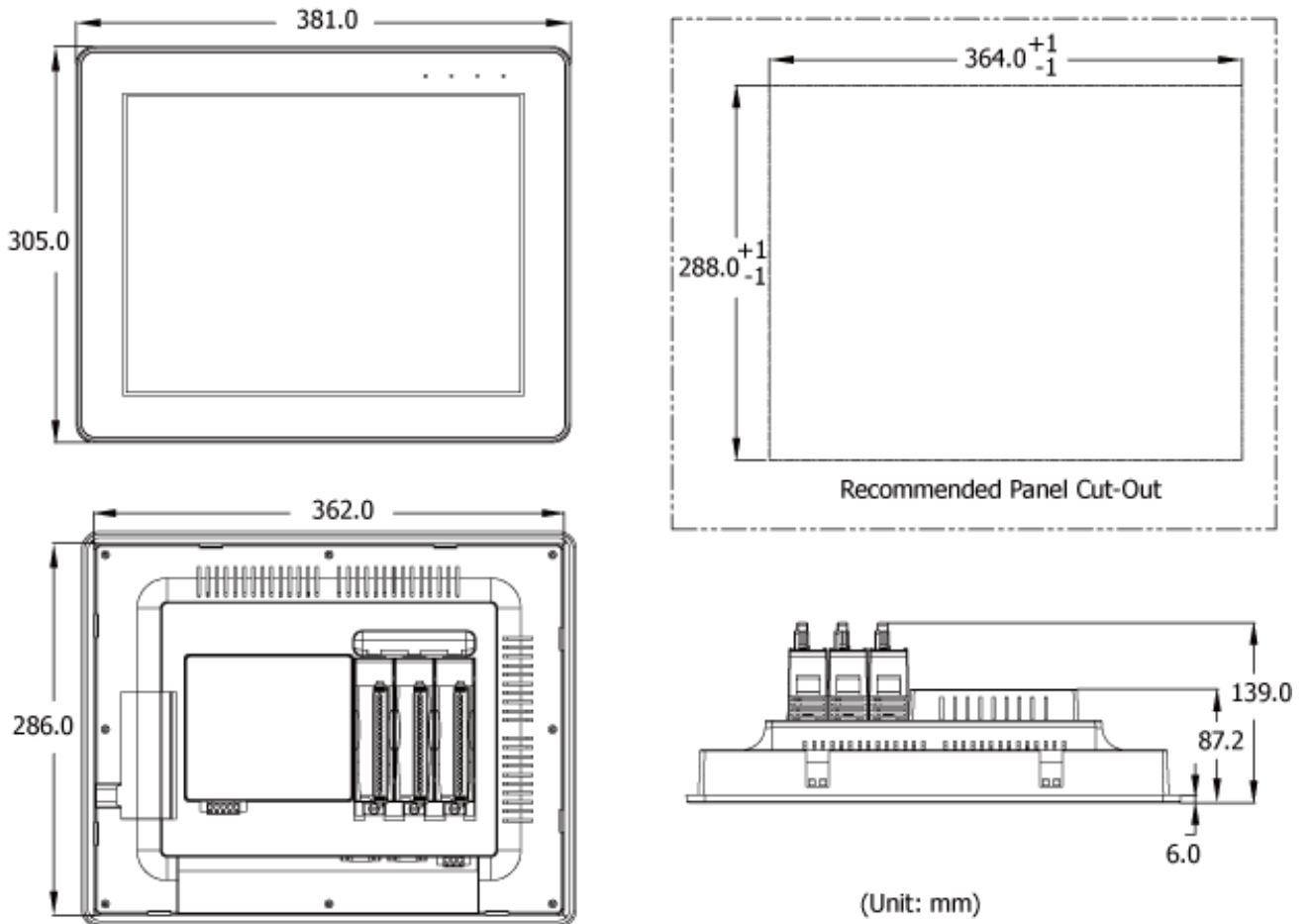


**Redundant Power and Ground**      **Expansion I/O Slots**

Item	Description
Redundant Power and Ground	<p>The iPPC-6731-WES7 has a terminal with 5-wire; there are 4-wire for redundant power input and a wire for frame ground, the details are shown as below.</p>  <p>The diagram shows a yellow terminal block with five terminals. From left to right, the terminals are labeled: F.G (Frame Ground), P.GND (Power Ground), PWR2 (Power), P.GND (Power Ground), and PWR1 (Power).</p>
Expansion I/O Slot	The expansion I/O slots support the high profile I-8K and the I-87K series I/O modules

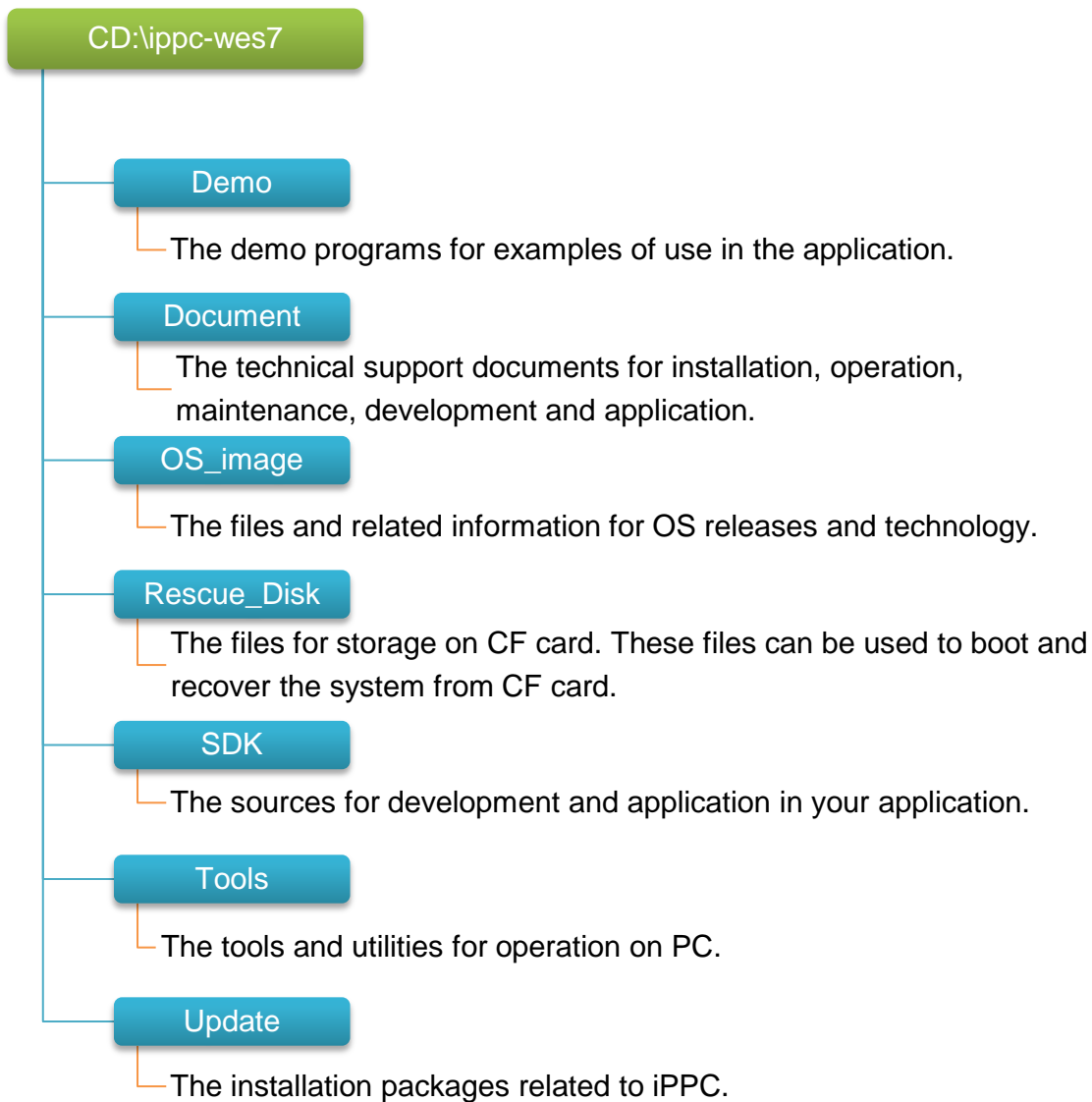
# 1.4. Dimensions

The diagrams below provide the dimensions and the panel cut-out to use in defining your enclosure specifications. Remember to leave room for potential expansion if you are using other components in your system. All dimensions are in millimeters.



## 1.5. Companion CD

This package comes with a CD that provides a collection of the software utility, documentation, drivers, demo program and application. The CD contains several subdirectories. All of them are listed below.

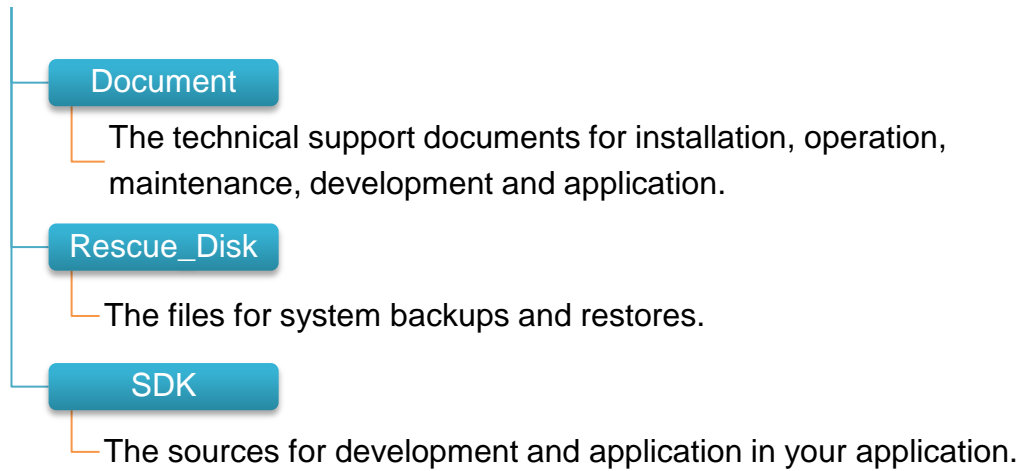




## 1.6. Rescue CF Card

The iPPC-6731-WES7 comes with a rescue compact flash card that not only supports rescue mechanism but also contains full usage documents, software tools and development resources related to the iPPC-6731-WES7. All of them are listed below.

### Rescue CF Card



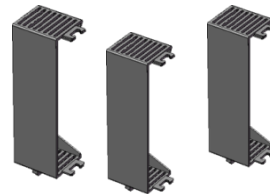
## 2. Getting Started

This chapter provides a guided tour of the iPPC installation and configuration that describes the steps needed to download, install, configure, and run the basic procedures for user working with the iPPC for the first time.

Before starting any task, please check the package contents. If any of the following package contents are missing or damaged, contact your dealer, distributor.



iPPC-6731-WES7



I/O Socket \* 3



Software Utility CD



A 8 GB CF Card



Screw Driver



Panel Clip \* 4



M4 x 30L Screw \* 4

## 2.1. Mounting the Hardware

iPPC-6731-WES7 has 3 expansion I/O slots to expand the functions, allowing it to communicate with external I/O devices, and before choosing the right I/O module, you first need to know the I/O expansion capacities in order to choose the best expansion module for achieving maximal efficiency.

There are more than 30 high profile I/O modules available for interfacing many different measurements, including thermocouple, voltage, RTD, current, resistance, strain, digital, ..., etc., and these modules have their own manuals, so if you are using them you should supplement this manual with the manual specifically designed for the special module.



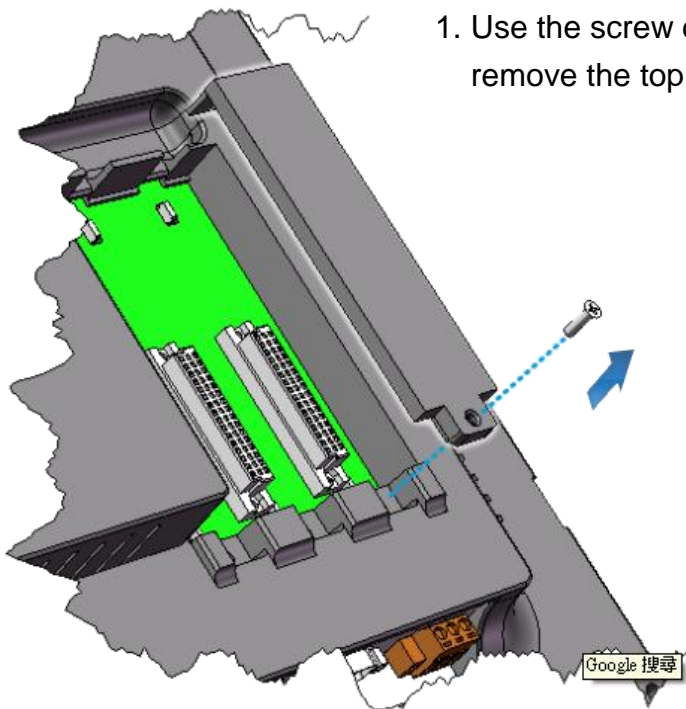
For more information about the I/O expansion modules that are compatible with the iPPC-6731-WES7, please refer to

[http://www.icpdas.com/products/PAC/i-8000/8000\\_IO\\_modules.htm](http://www.icpdas.com/products/PAC/i-8000/8000_IO_modules.htm)

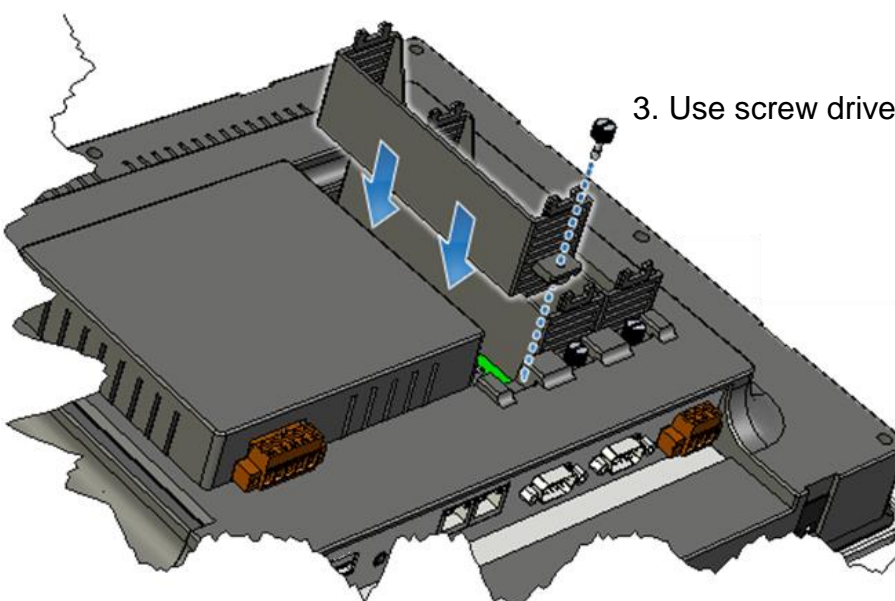
## 2.1.1. Inserting the I/O Modules

There is a top case on each I/O socket. Before inserting the I/O module you first need to remove it.

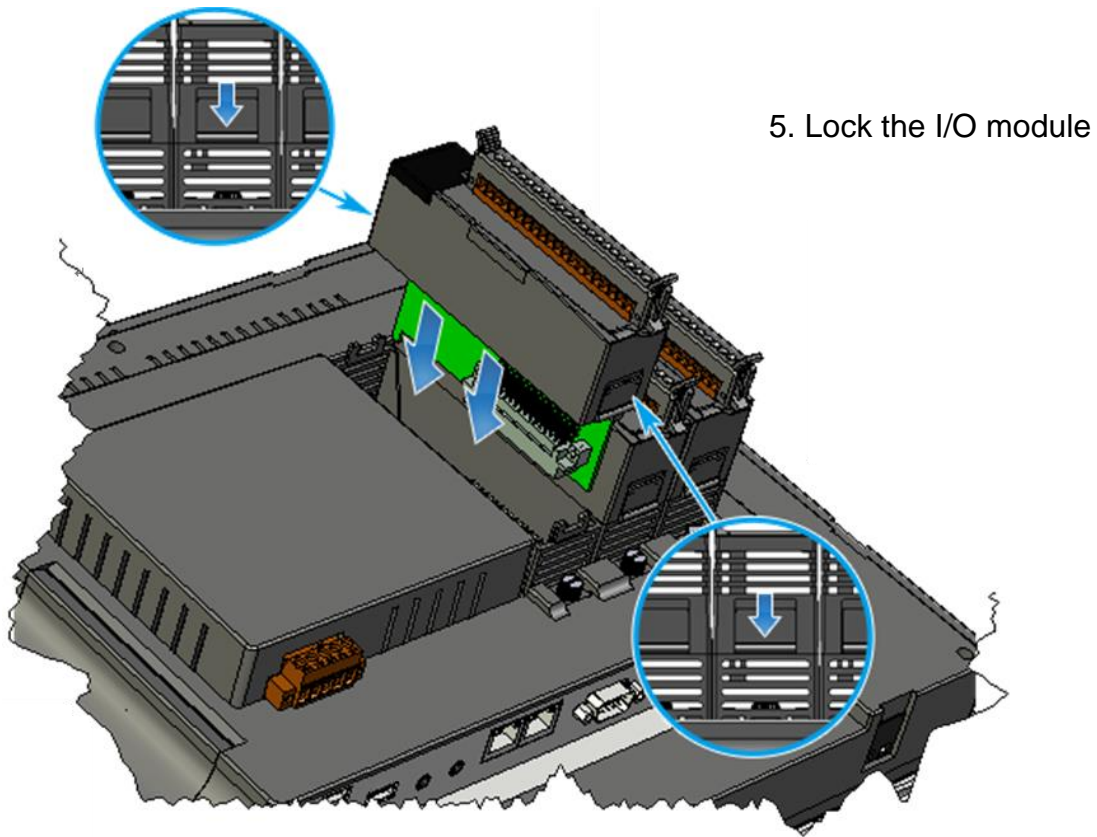
### To insert the I/O module



2. Hold the I/O socket vertically and align the iPPC-6731-WES7 socket and then carefully press the I/O socket onto the socket.



4. Hold the I/O module vertically and align the socket and then carefully press the I/O module onto the socket.



### Tips & Warnings

---

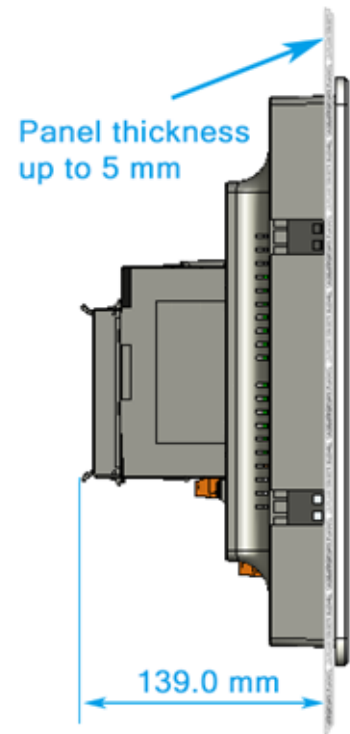


If you do not expand the I/O module full, please keep the top case of the unused slot to protect the backplane from dirt, dust and damage from foreign objects.

---

## 2.1.2. Mounting the iPPC

The iPPC can be mounted on a panel of maximum thickness 5 mm. Adequate access space can be available at the rear of the instrument panel for wiring and servicing purposes.



### Tips & Warnings

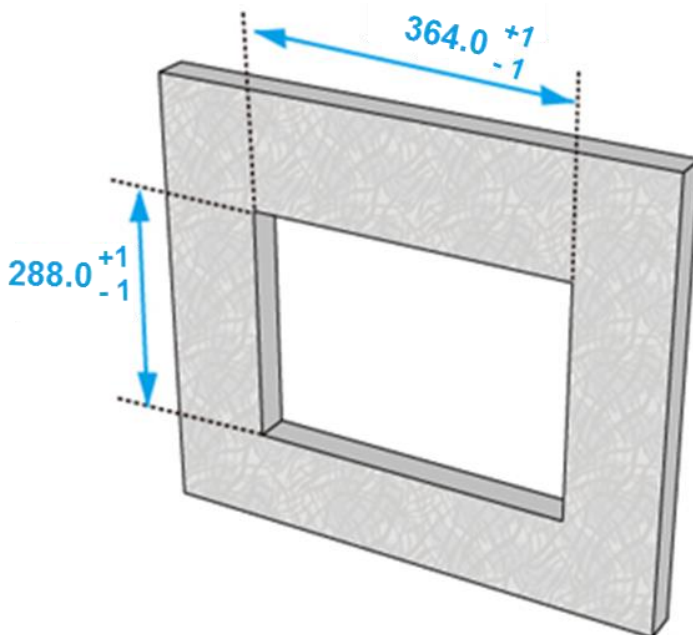


To ensure proper ventilation for the iPPC, leave a minimum of 50 mm space between the top and bottom edges of the iPPC and the enclosure panels.

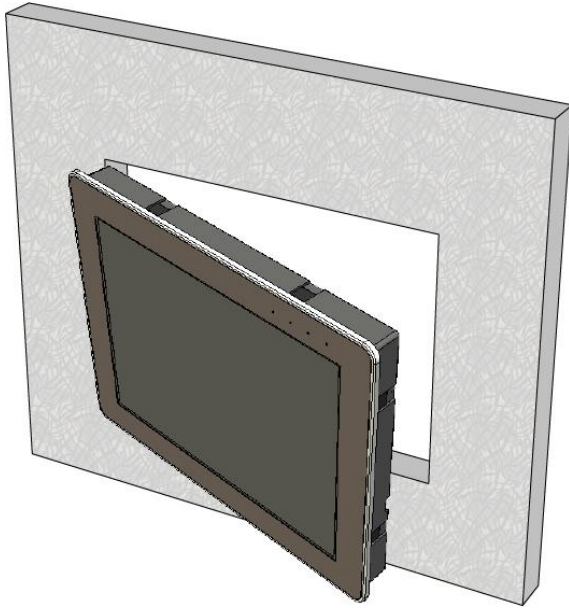
### To mount the iPPC

1. Prepare the panel and cut the hole to the specified size

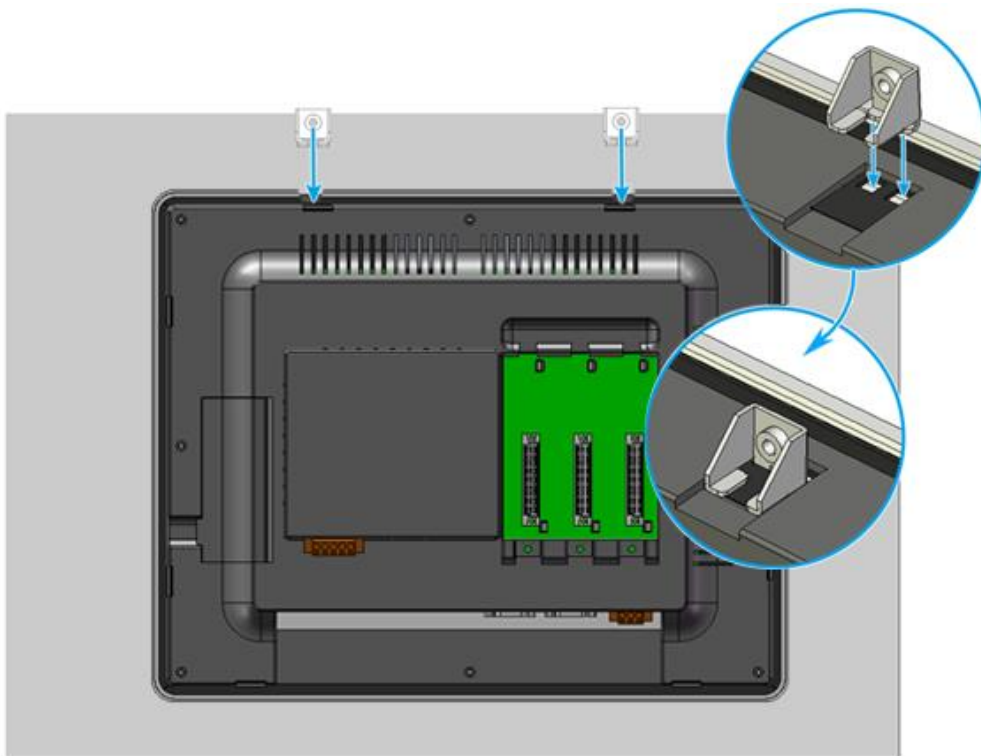
The table below shows the dimension of the panel cut-out of iPPC.



2. Attach the iPPC to the cut-out hole

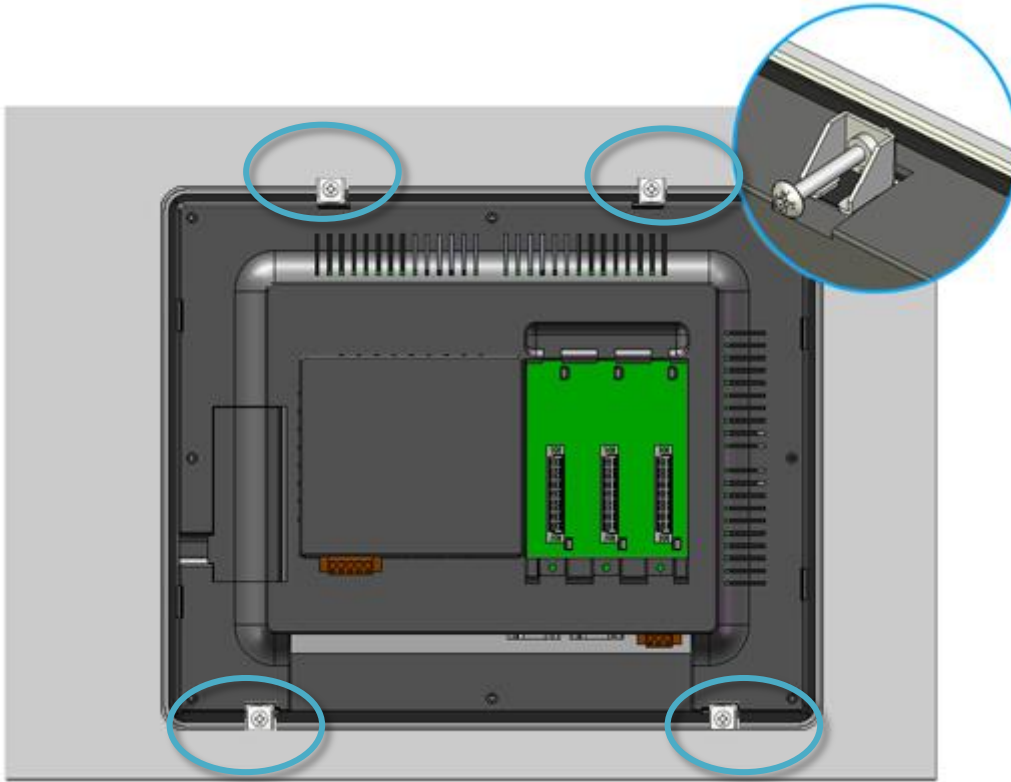


3. Insert the panel mounting clips into the upper and lower ventilation holes.





4. Screw the panel mounting clips to the panel.



### Tips & Warnings

---



Recommended Screw Torque: 3.4 ~ 4.5 kgf-cm.

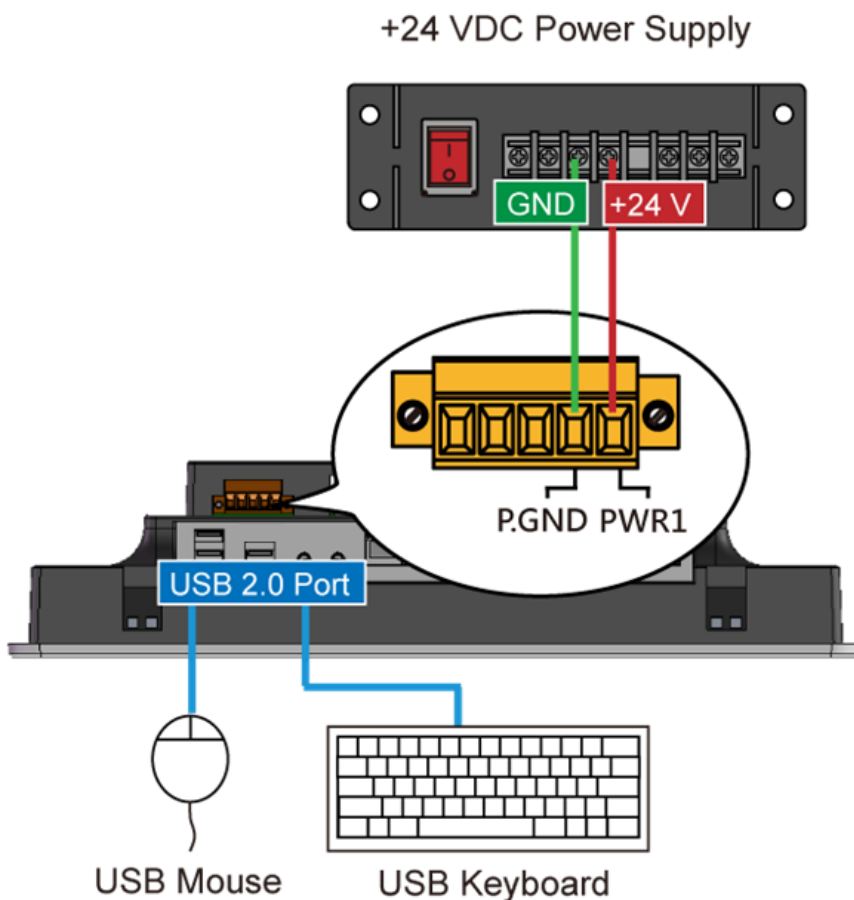
---



### 2.1.3. Deploying a Basic iPPC System

The iPPC provides a variety of interfaces to suit a range of applications. The following illustration shows a simple application.

**To deploy a basic iPPC system**



As the above illustration shows, the interfaces of iPPC can be used to perform the following tasks:

- Connect the positive terminal (+) of the power supply to the terminal PWR and the negative terminal (-) of the power supply to the P.GND.
- Connect to a USB mouse or keyboard

## 2.1.4. Supplying Power to iPPC

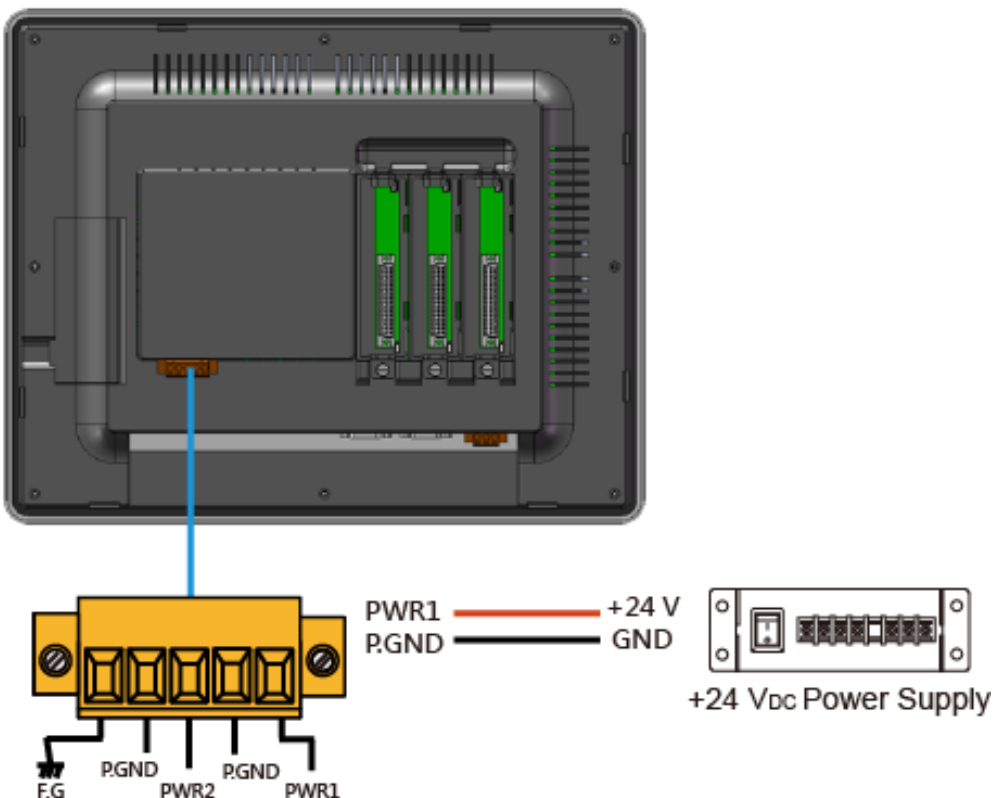
The iPPC has two power inputs that can be connected simultaneously to live DC power sources. If one power source fails, the other live source acts as a backup, and automatically supplies all of iPPC power needs.

For more information about the supported power supply from ICP DAS web site, see [http://www.icpdas.com/products/Accessories/power\\_supply/power\\_list.htm](http://www.icpdas.com/products/Accessories/power_supply/power_list.htm)

### To supply power to iPPC

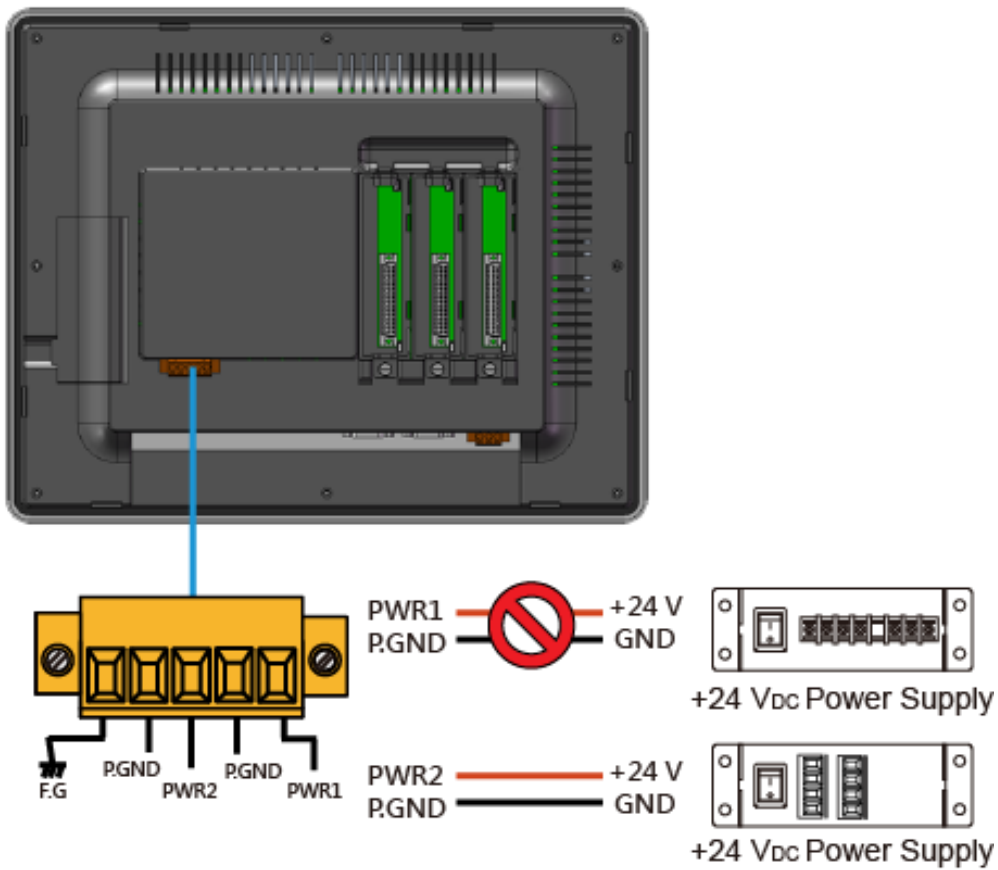
There are two ways to supply power to the iPPC.

#### i. Single power supply



## ii. Redundant power supply

The redundant power can be used single and used two self-governed power to supply to the system, PWR1 and PWR2 input at the same time, when one power fails, the other power acts as a backup, and automatically supplies power needs.



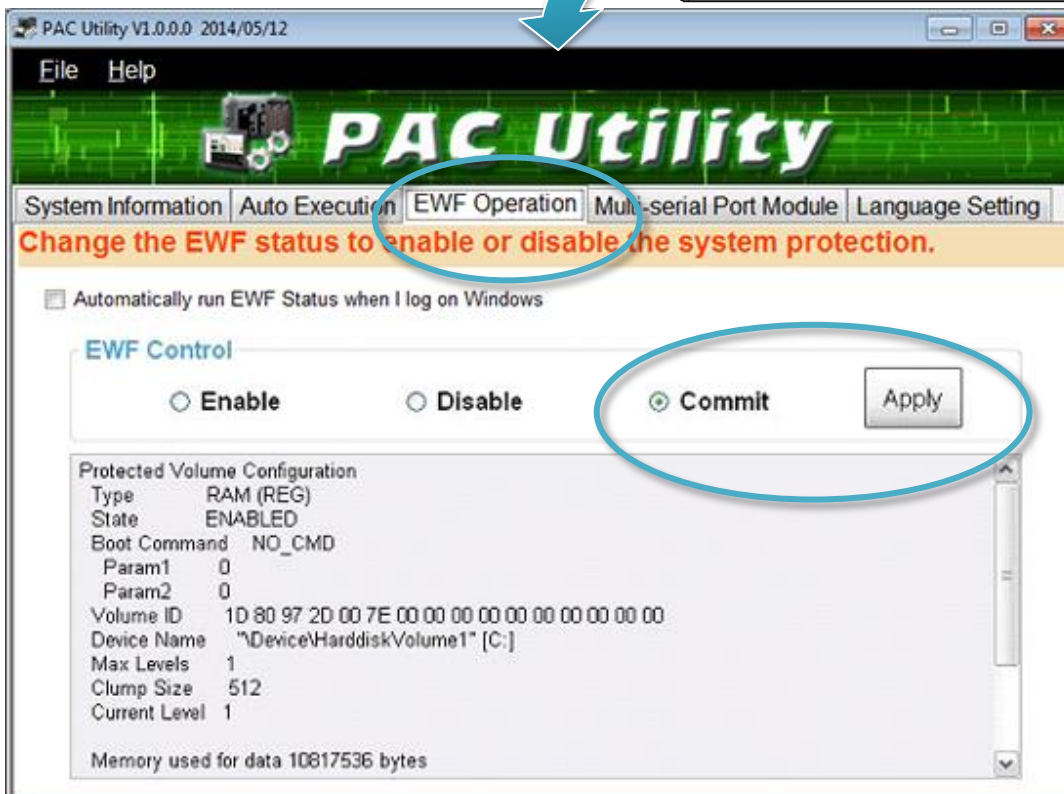
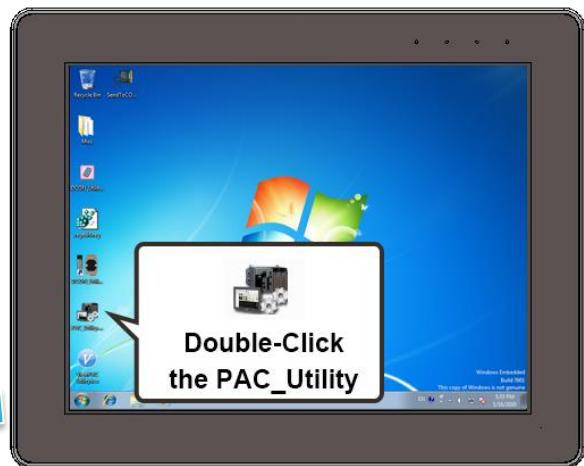
## 2.2. Committing EWF to Allow Settings to Be Saved

The EWF is a safety mechanism that provides the ability to control write protection of the iPPC system built in C: drive. Any changes made to the system are lost when the start restarts while EWF is enabled, unless they are committed to the system.

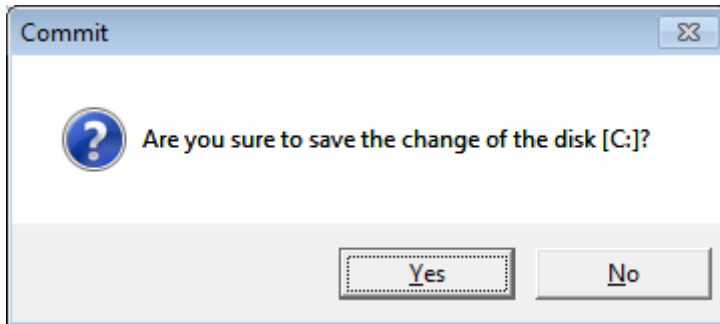
For more details about the EWF, please refer to section 3.4. Configuring the EWF Manager.


### To commit EWF to allow settings to be saved

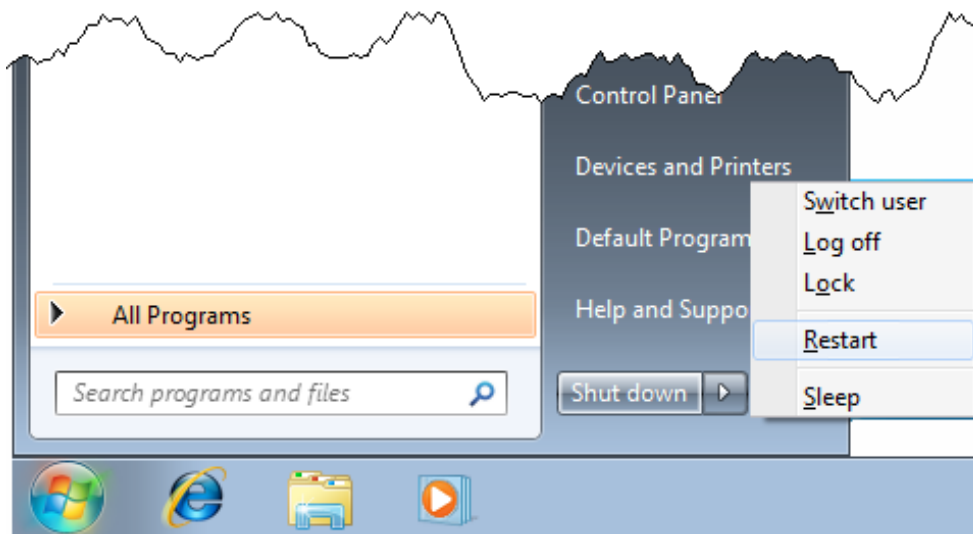
1. Click the **PAC Utility** shortcut on the desktop
2. Click the **EWF Operation** tab, select the **Commit** check box, and then click **Apply** button



3. In the pop-up dialog box, click **Yes** button



4. Click the **Start** button , click the arrow next to the **Shut Down** button, and then click **Restart**

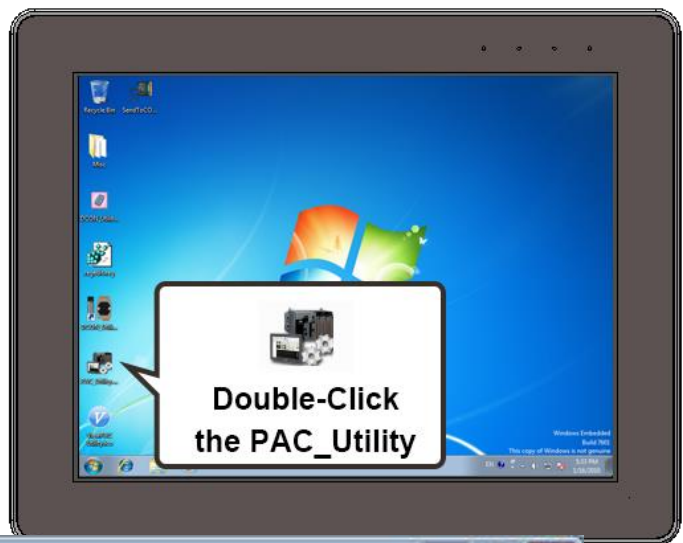


## 2.3. Changing the User Interface Language

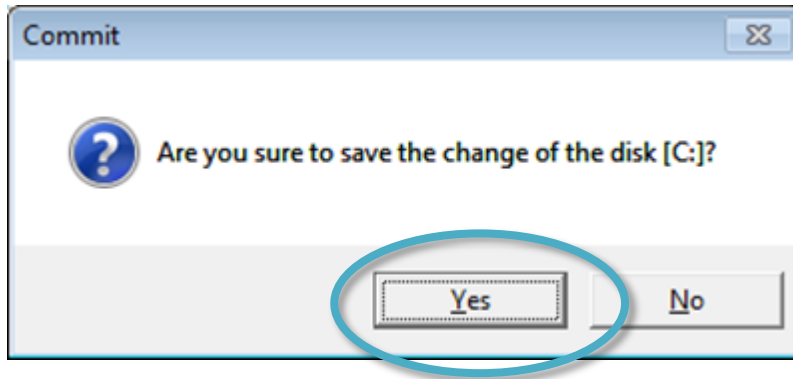
The Region and Languages Settings is a WES7 functionality that allows users to easily change the iPPC user interface with your native language by PAC Utility.

To change the user interface language

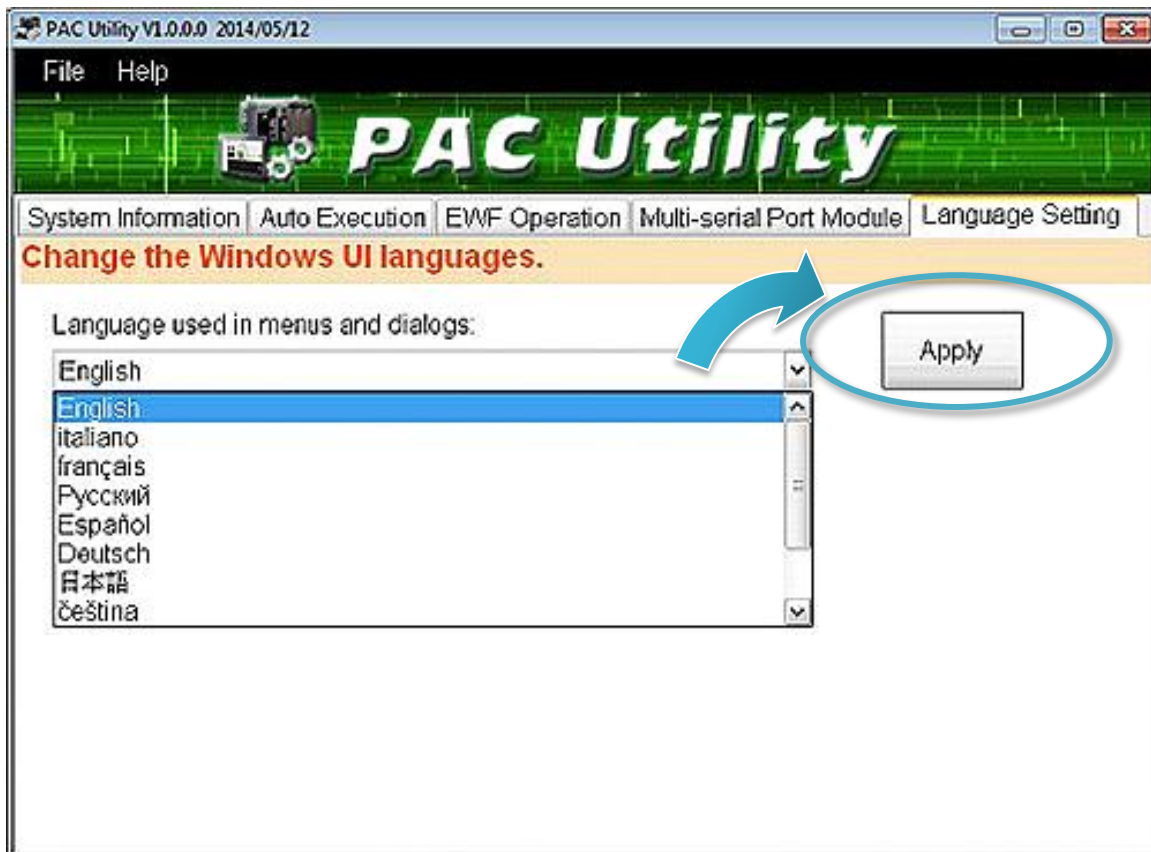
1. Click the **PAC Utility** shortcut on the desktop
2. Click the **Language Setting** tab



3. In the pop-up dialog box, click **Yes** button to save changes to the disk.

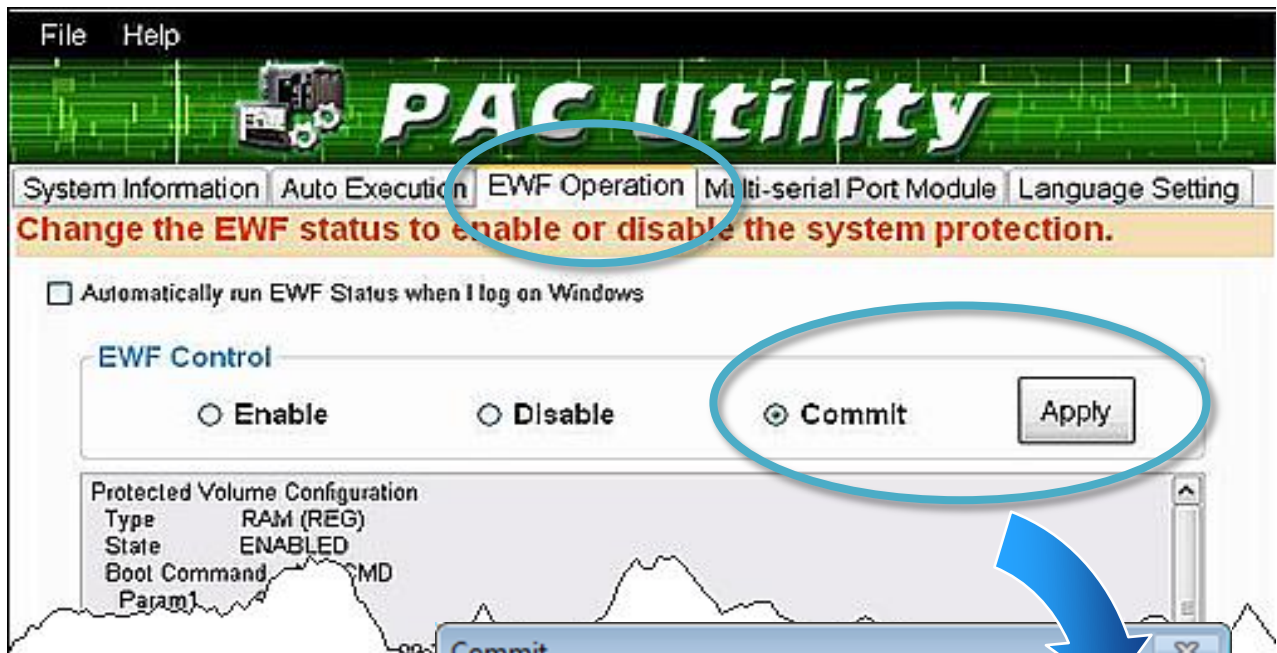


4. Click on the drop down arrow and choose your preferred language that you want to use for iPPC menus and dialogs, and then click **Apply** button

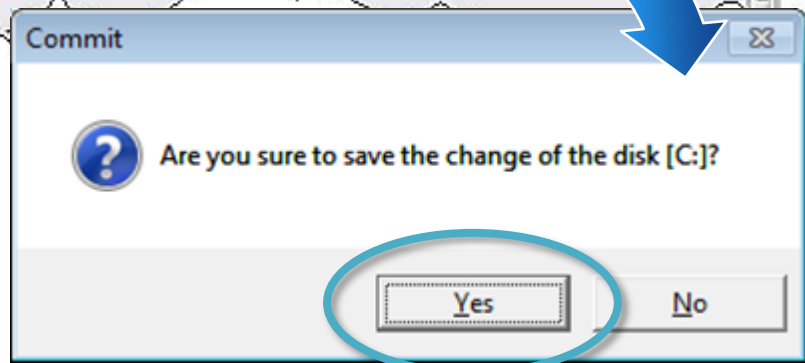





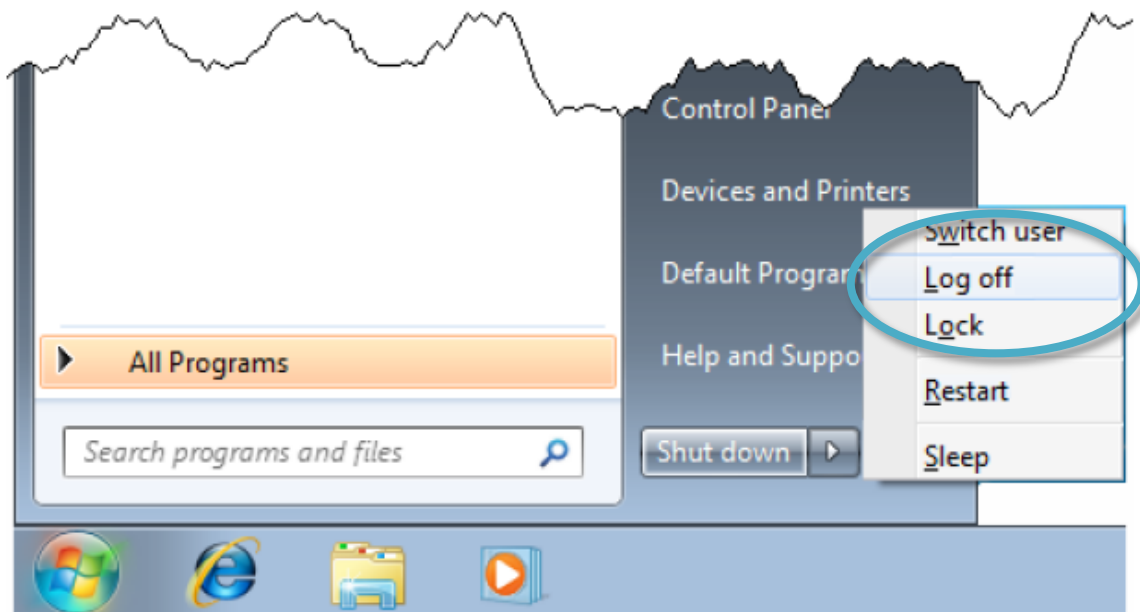
5. Click the **EWF Operation** tab, select the **Commit** option, and then click **Apply** button.



6. In the pop-up dialog box, click **Yes** button to save changes to the disk.



7. Click the **Start** button , click the arrow next to the **Shut Down** button, and click the Log off button to log out immediately, and then logon again for the change to be effective.




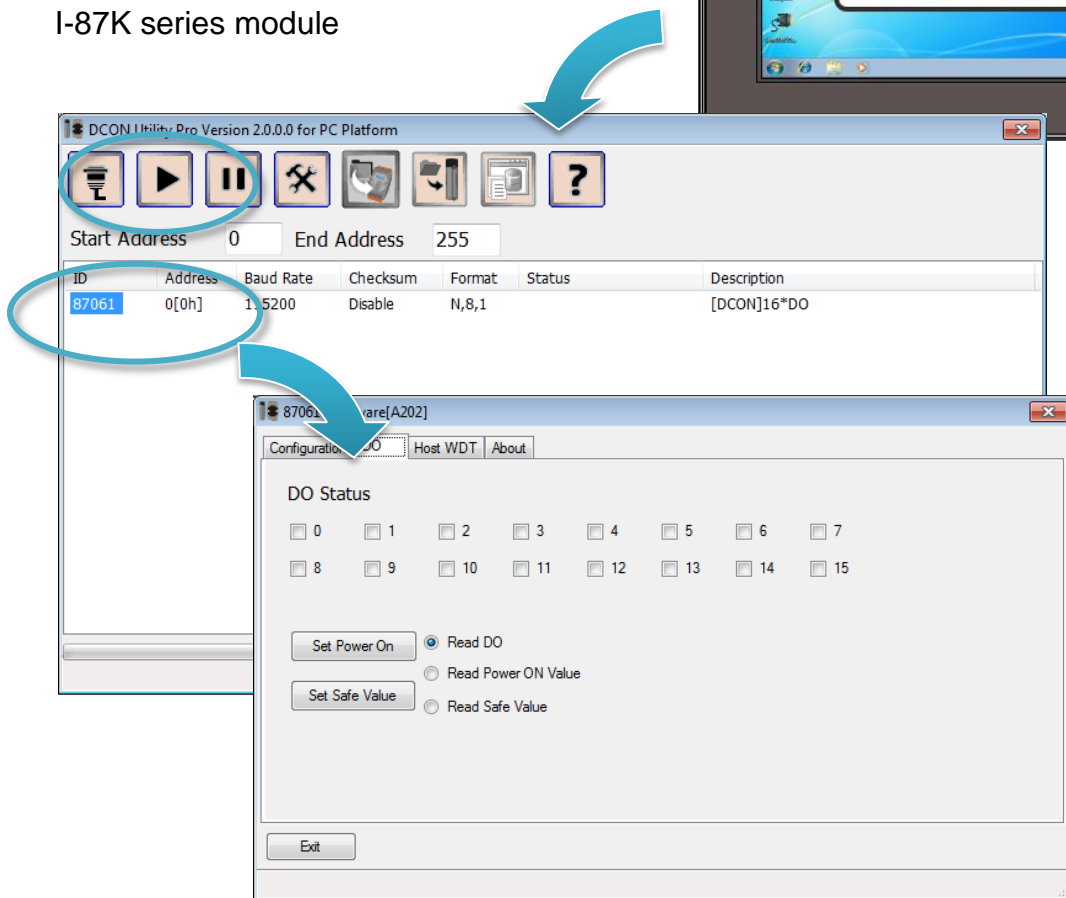
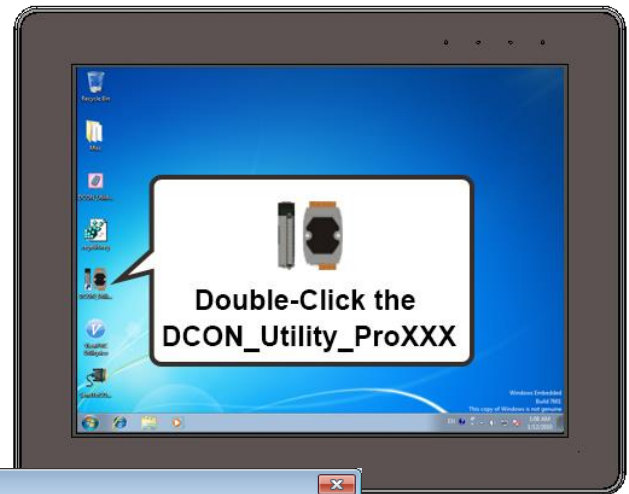


## 2.4. Using DCON Utility Pro to Configure I/O Modules

DCON Utility Pro is a tool kit designed to quickly control and manage I-87K series expansion I/O modules.

### To use DCON Utility Pro to configure I/O module

1. Click the **DCON Utility Pro** shortcut on the desktop
2. Click the **Search**  button
3. Click the module name to configure the I-87K series module



### 3. Security and Risk

This chapter provides information of technological security risks and solutions associated with the iPPC services.

Security is important for iPPC. Based on WES7, iPPC can avoid many security vulnerabilities. The following provides some security policy that you should consider before you develop your iPPC.

- Windows Firewall
- IIS (Internet Information Service)
- EWF (Enhanced Write Filter)

The following table provides the default settings of the iPPC security policy.

Security Item	Default Settings	User Name	Password
Firewall	Enable	N/A	N/A
IIS	Enable	anonymous	Blank
EWF	Enable	N/A	N/A


## 3.1. Creating and Managing User Accounts

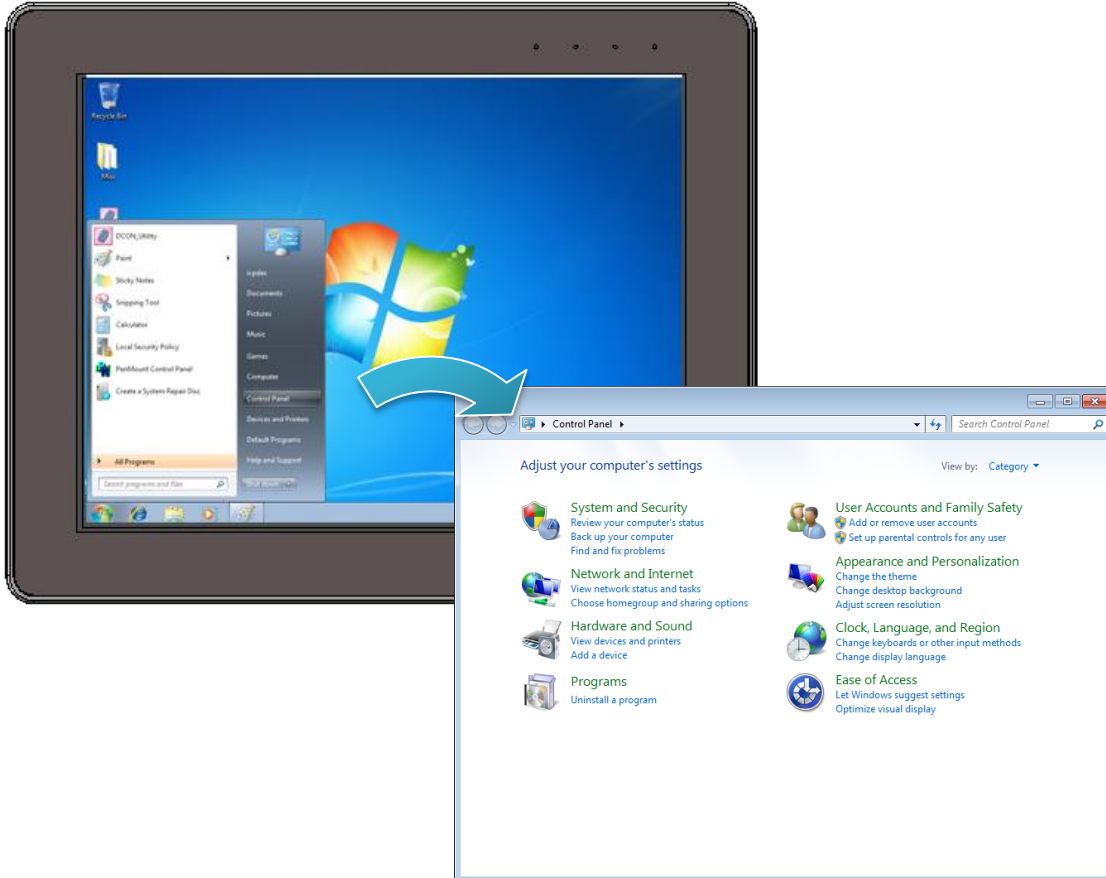
Based on WES7, iPPC includes several components for managing user account names, groups, and passwords.

- The Administrator Account component allows you to specify the password for the local Administrator account. You can only include one Administrator Account component in your configuration.
- The User Account component allows you to specify the user name, group, and password for a local user account. You must add a separate User Account component for each user in your configuration.
- Additional components are required if you want to provide end-user access to account settings, passwords, and display names in User Accounts in Control Panel.

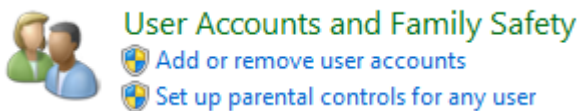
In a new OS installation, you can enter in your user name to create the initial account. This user will have full administrator privileges and automatically be logged in after setup is completed.

## To open the user accounts tool

1. Click the **Start** button  , and then click **Control Panel**



2. Click the **User Accounts and Family Safety**

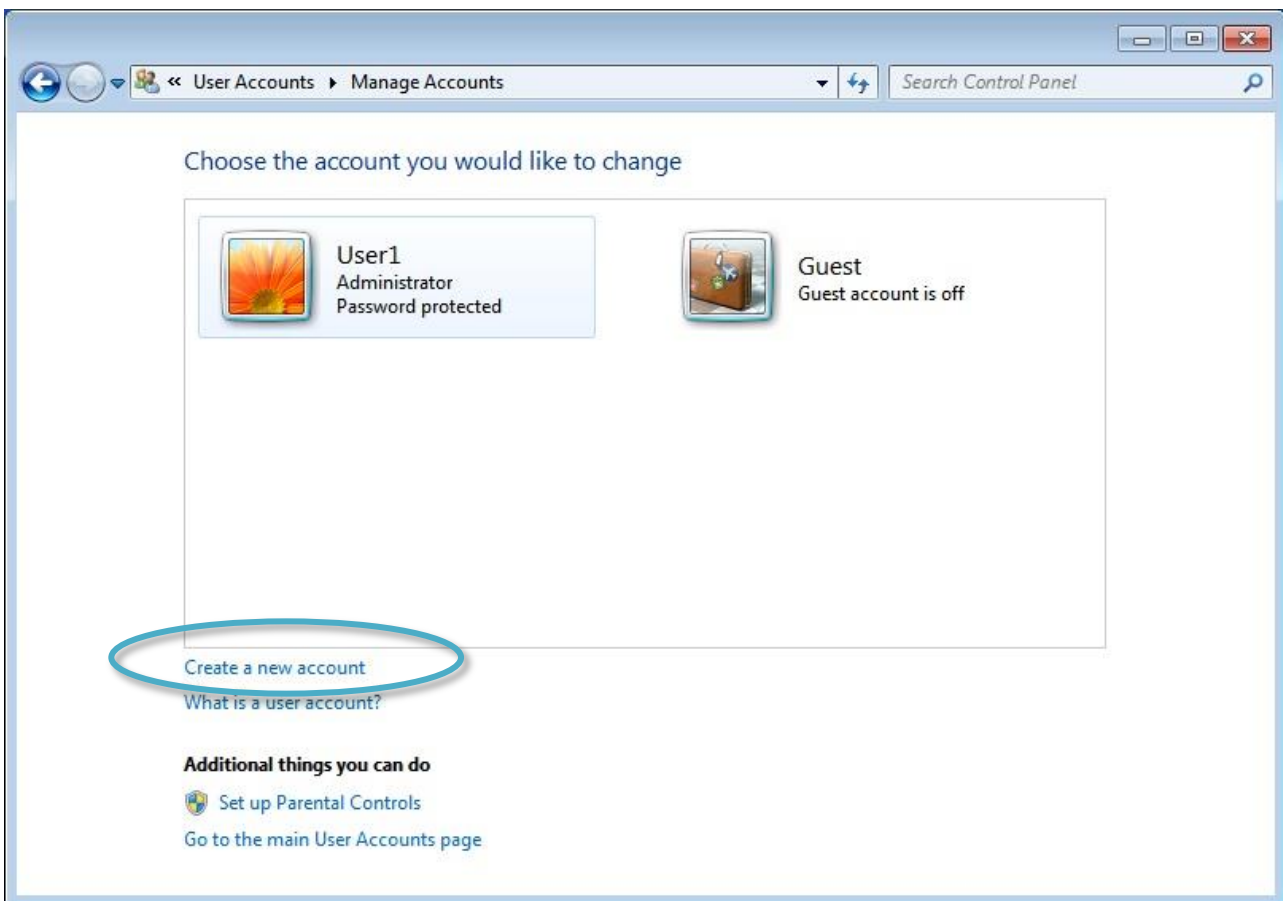


## To create a new use account

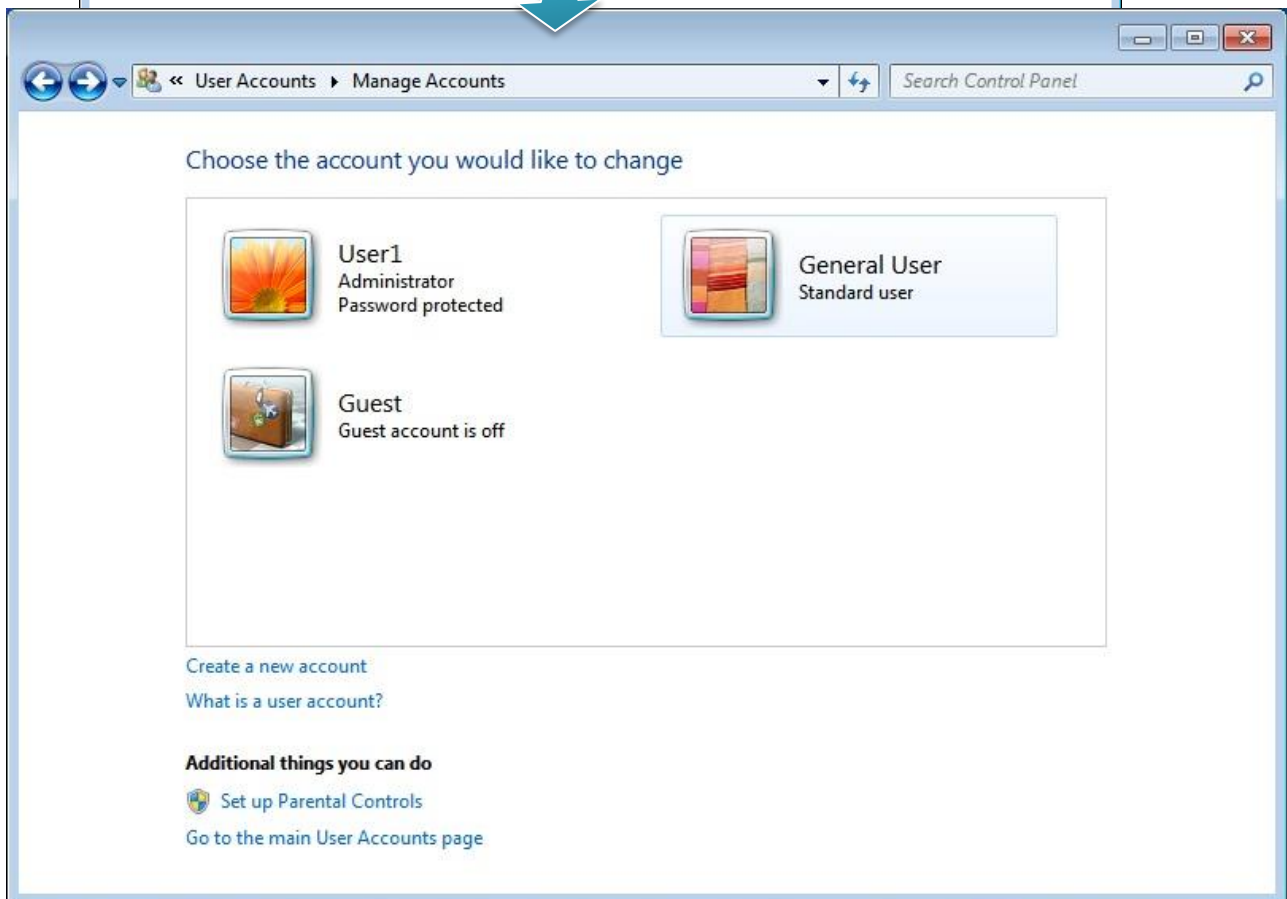
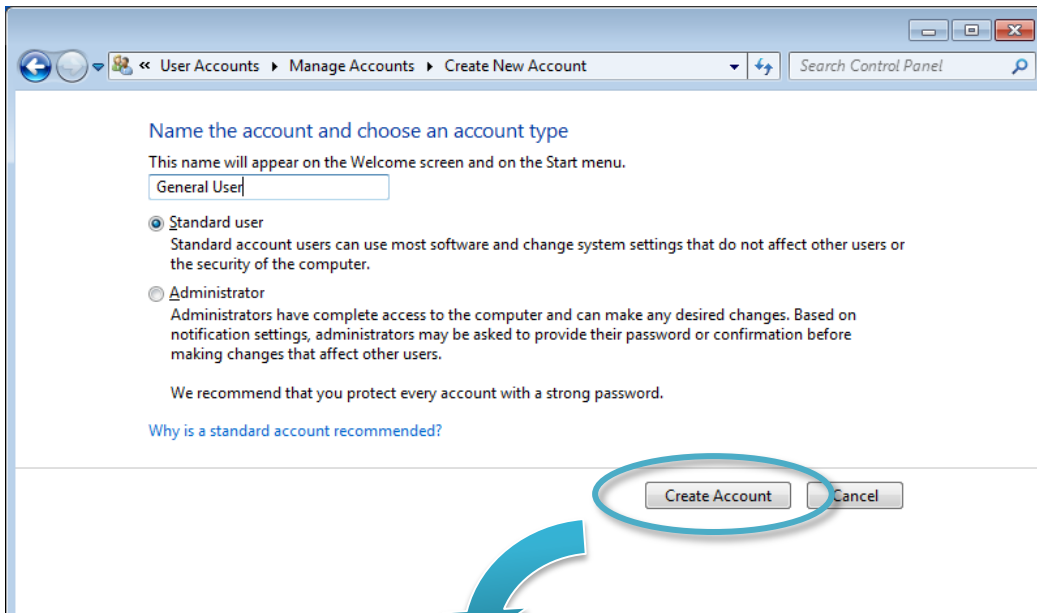
1. Click the **Add or remove user accounts**



2. Click the **Create a new account**

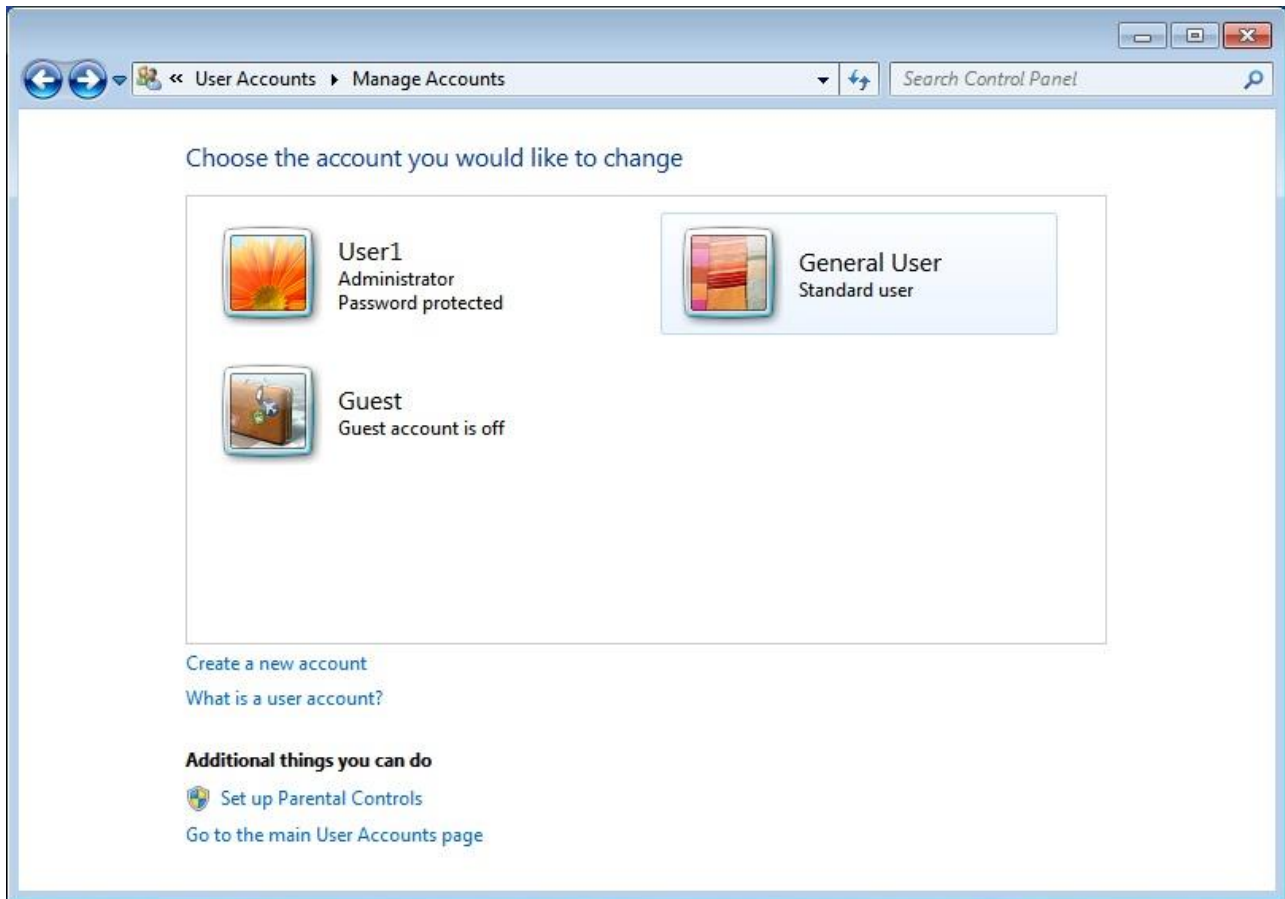


3. Type the name that you want to use for the account, select the desired account type, and then click **Create Account**

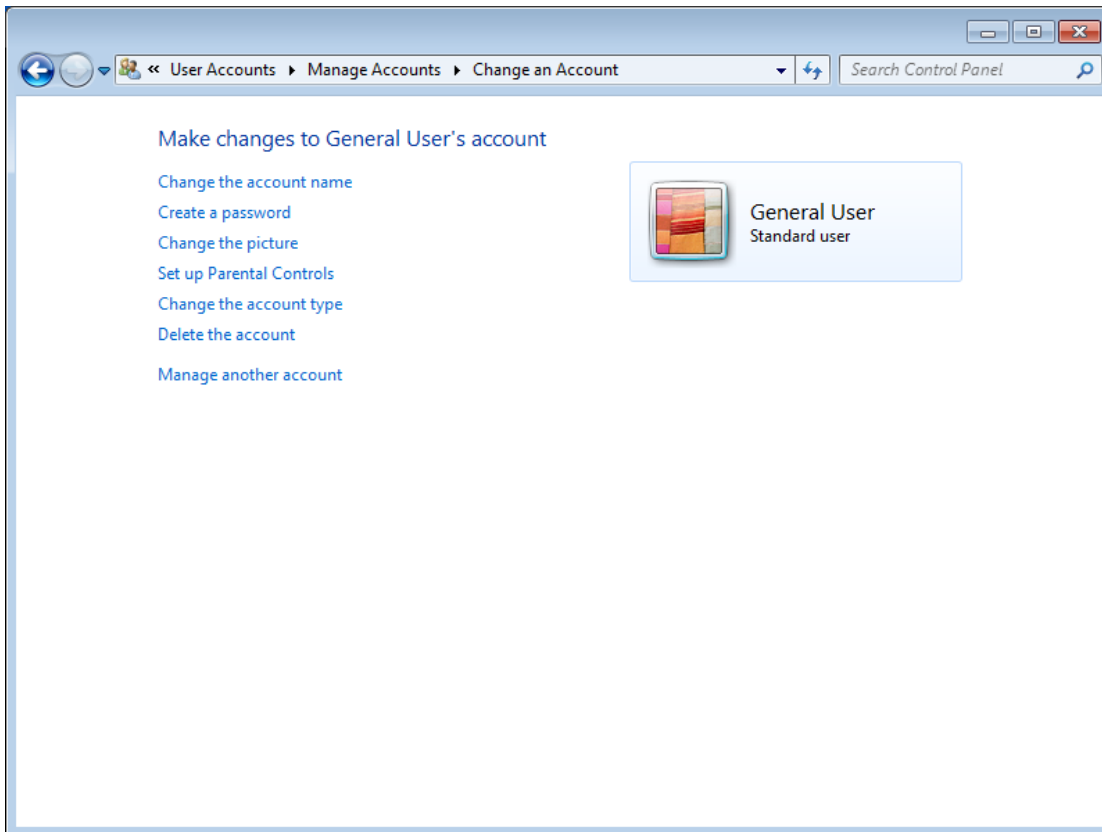


## To Make Changes to an Account

1. Click the account that you want to change.



## 2. Select the item that you would like to change:




- Click the **Change the account name** to change the name that appears on the Welcome screen for the account.
- Click the **Create/change a password** to create or change the password for the user and create or change the password hint.
- Click the **Change the picture** to change the picture that is used to represent the user account. You can use any image file on the computer for the user's picture.
- Click the **Set up Parental Controls** to set up the controls on each child's account.
- Click the **Change the account type** to change the account type to increase or decrease the user's rights on the computer.
- Click the **Delete the account** to delete the user account from the computer. When you delete the account, you are given the option to save the user's files on the computer.
- Click the **Manage another account** to manage another account.

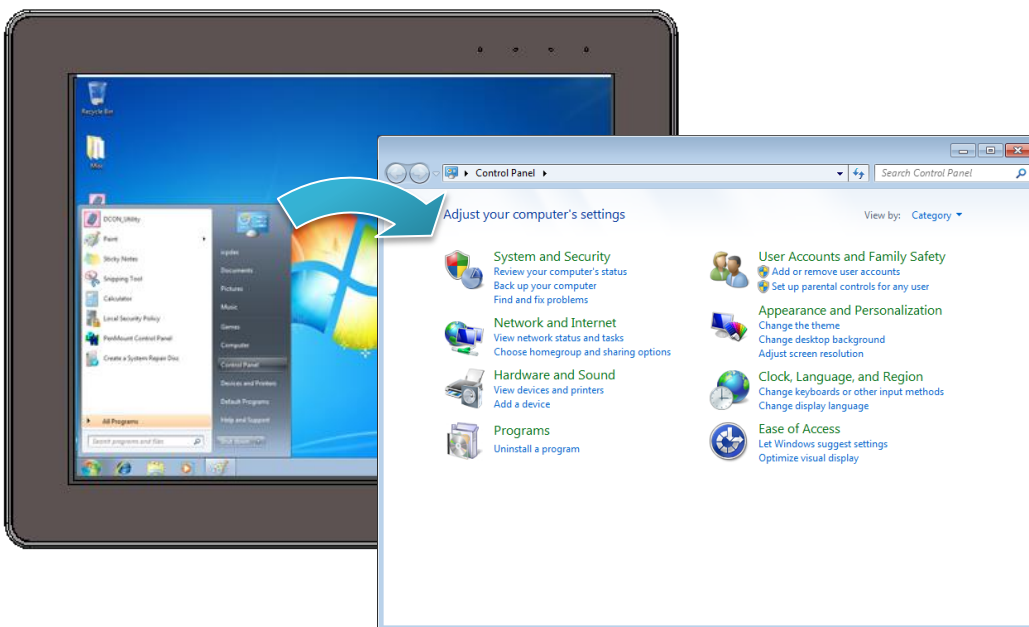


## 3.2. Turning Firewall On or Off

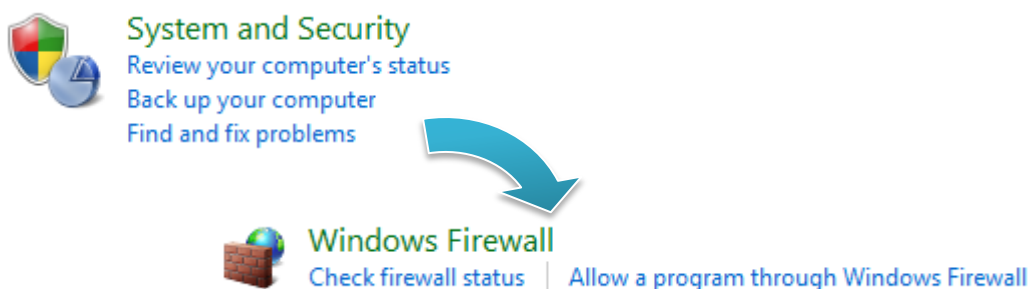
Based on WES7, iPPC Firewall with Advanced Security and the related firewall technologies documented here enable user to share Internet connections, protect connections using a firewall, and provide Network Address Translation (NAT).

### To open the Windows Firewall tool

1. Click the **Start** button  , and then click **Control Panel**

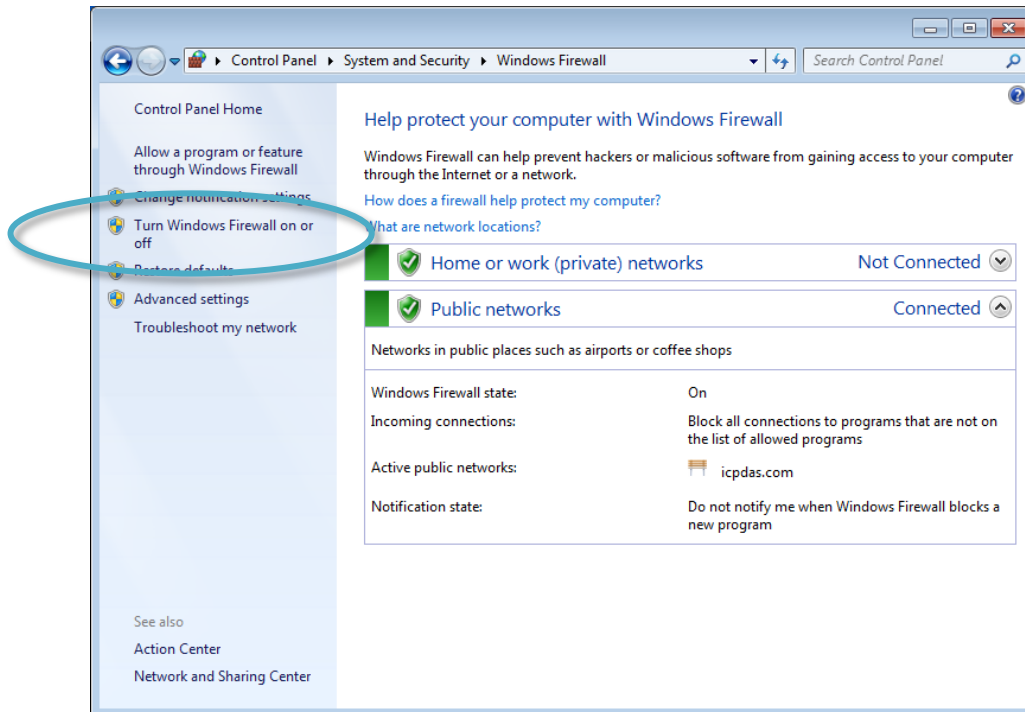


2. Click the **System and Security**, and then click **Windows Firewall**

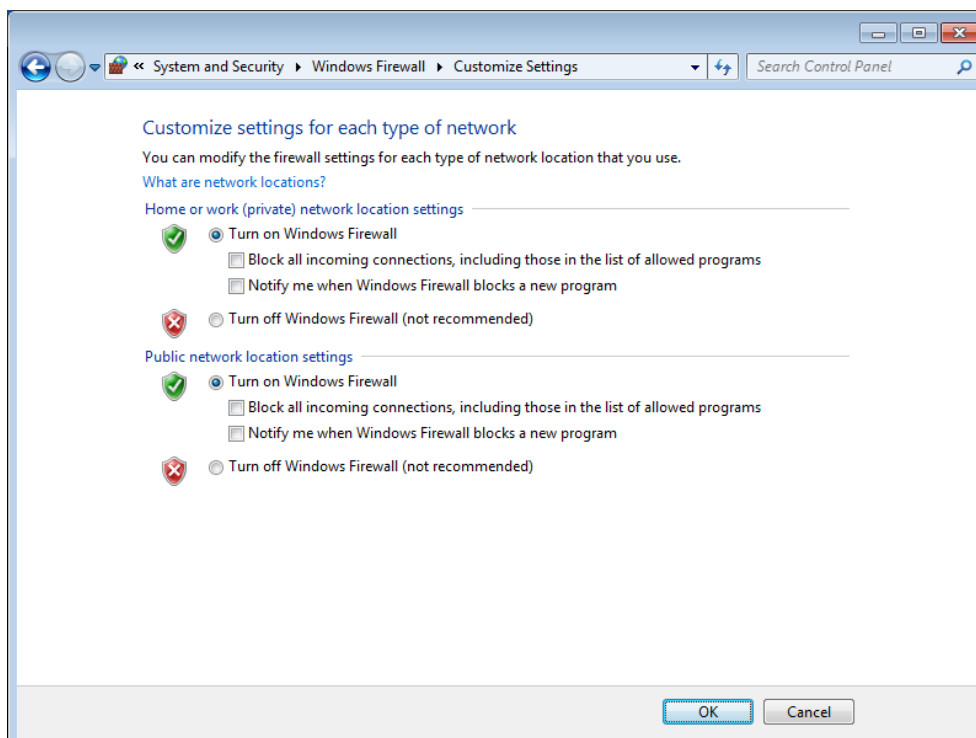


## To turn on/off Windows Firewall

1. Click the **Turn Windows Firewall on or off** in the left panel



2. Select the settings for your home/work (private) or public network, and then **OK**



## 3.3. Configuring the FTP Server


Microsoft Internet Information Service (IIS) is installed in iPPC and enabled as default. The IIS includes FTP Server and Web Server. It is convenient to use anonymous FTP access as default. However, it may cause some security problems too.

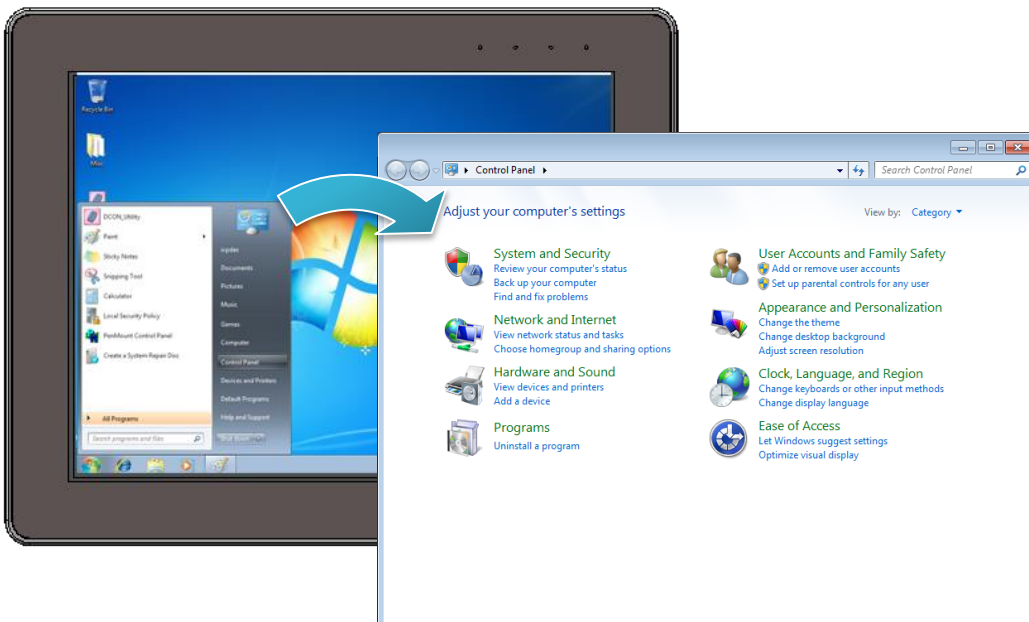
### Tips & Warnings



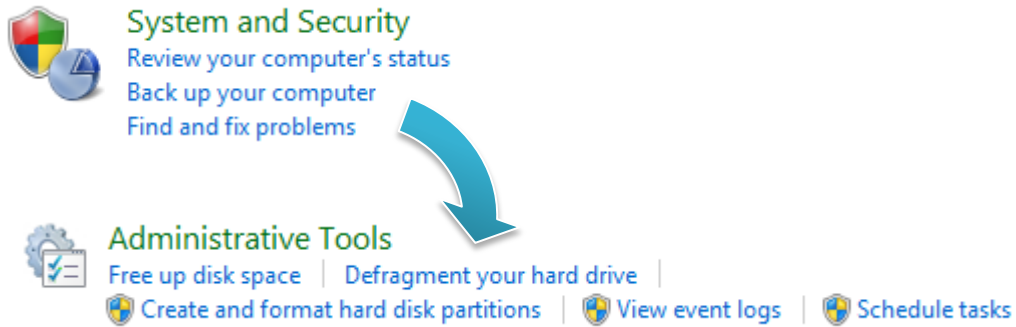
If you can't set up the FTP server, this is probably caused by the firewall setting, turn the firewall off.

### To configure the FTP Server

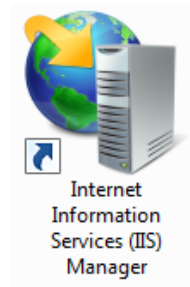
1. Click the **Start** button  , and then click **Control Panel**



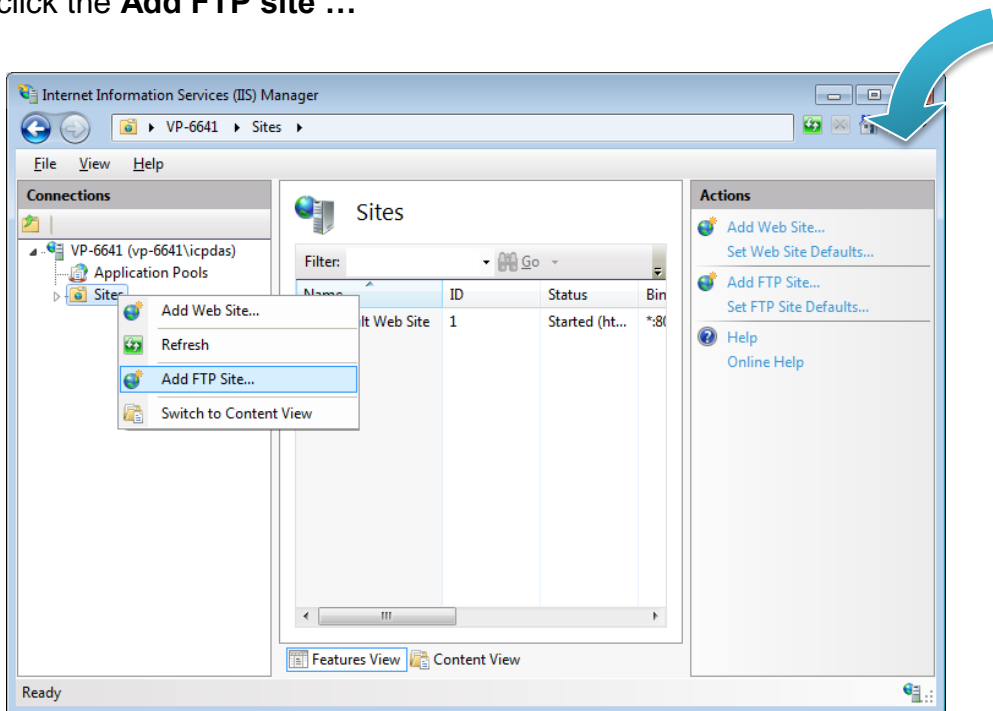
2. Click the **System and Security**, and then click **Administrative Tools**



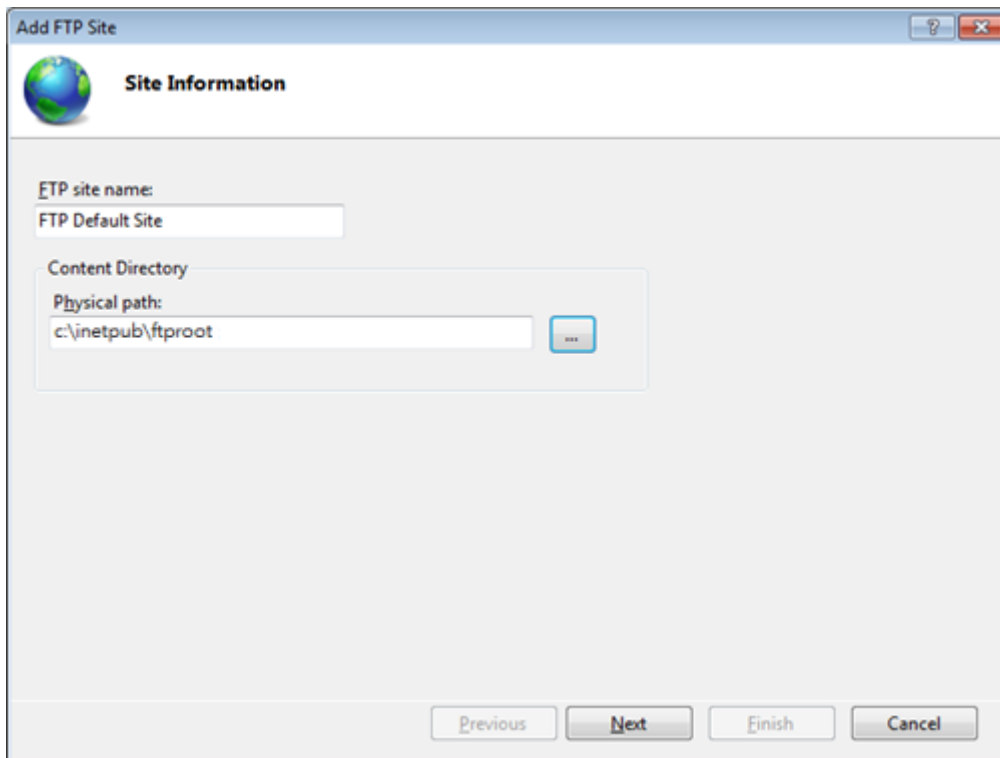
3. Click the **Internet Information Services (IIS) Manager**



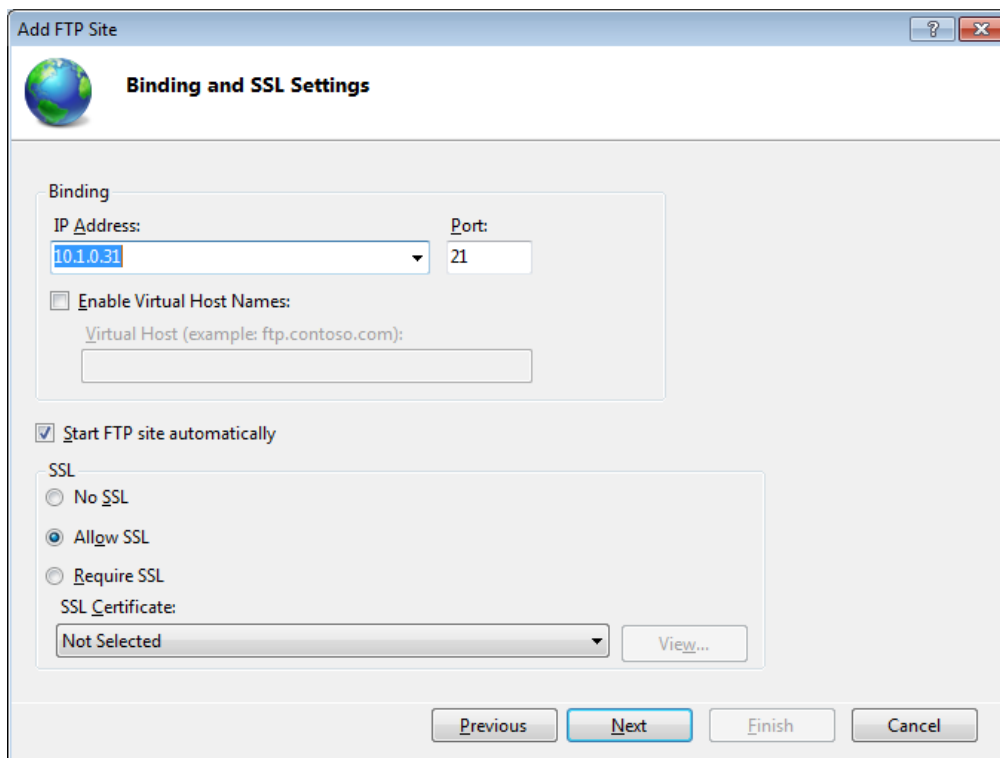
4. In the left pane, expand the local machine, and right-click **Sites**, and then click the **Add FTP site ...**



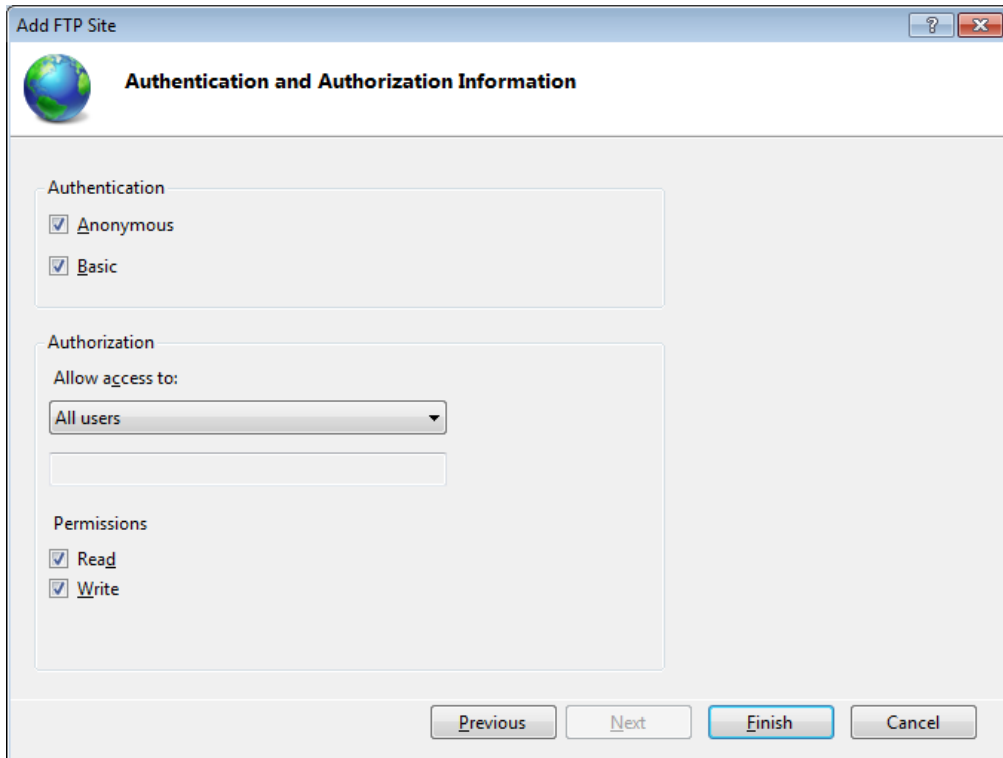
5. Type the name, and then select the Physical path (The default path is c:\inetpub\ftproot)



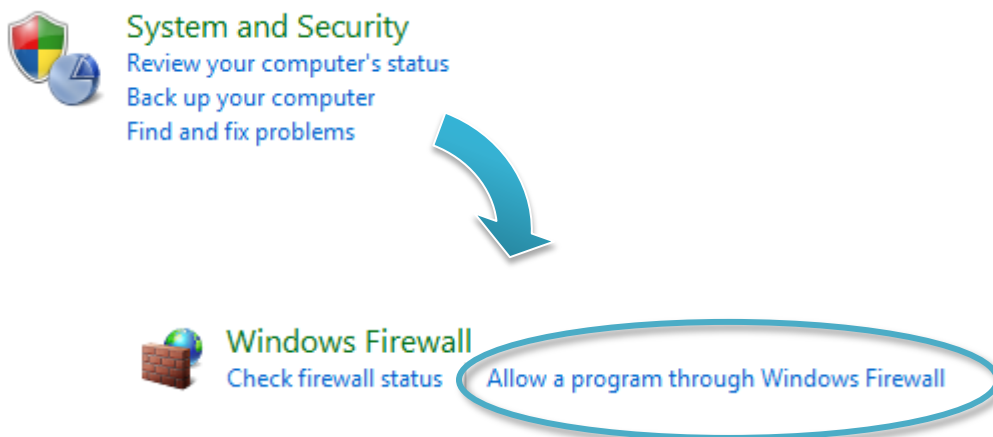
6. Click on the drop down arrow and choose the IP address of your iPPC, and then click the **Next** button



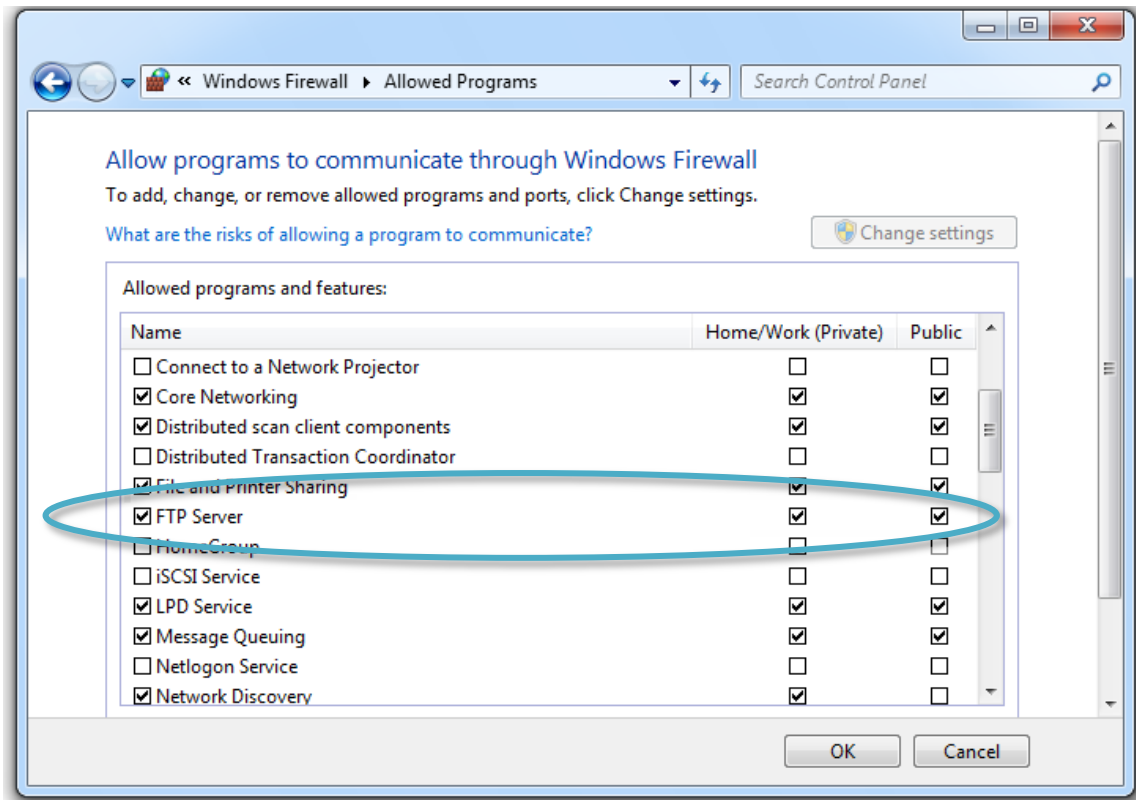
7. Click on the drop down arrow and choose the **All users**, and then select the **Read** and **Write** check box, and then click the **Finish** button



8. Go to the **Control Panel**, click **System and Security**, and then click the **Allow a program through Windows Firewall**



9. Select the **FTP Server** check box, and then click the **OK** button



## 3.4. Configuring the EWF Manager

EWF provides a means for protecting a volume from writes. All writes to an EWF -protected volume are redirected to an overlay. These writes are stored in the overlay and made available as part of the volume. In this way, it feels like that the volume is writeable. The overlay may exist either on disk or in RAM. If desired, the data stored in the overlay may be committed to the protected volume. The following figure is an overview of EWF.

For more detailed information about Enhanced Write Filter (EWF), please refer to [http://msdn.microsoft.com/en-us/library/ms912906\(WinEmbedded.5\).aspx](http://msdn.microsoft.com/en-us/library/ms912906(WinEmbedded.5).aspx)

On iPPC, only the C drive that OS resides can be protected.

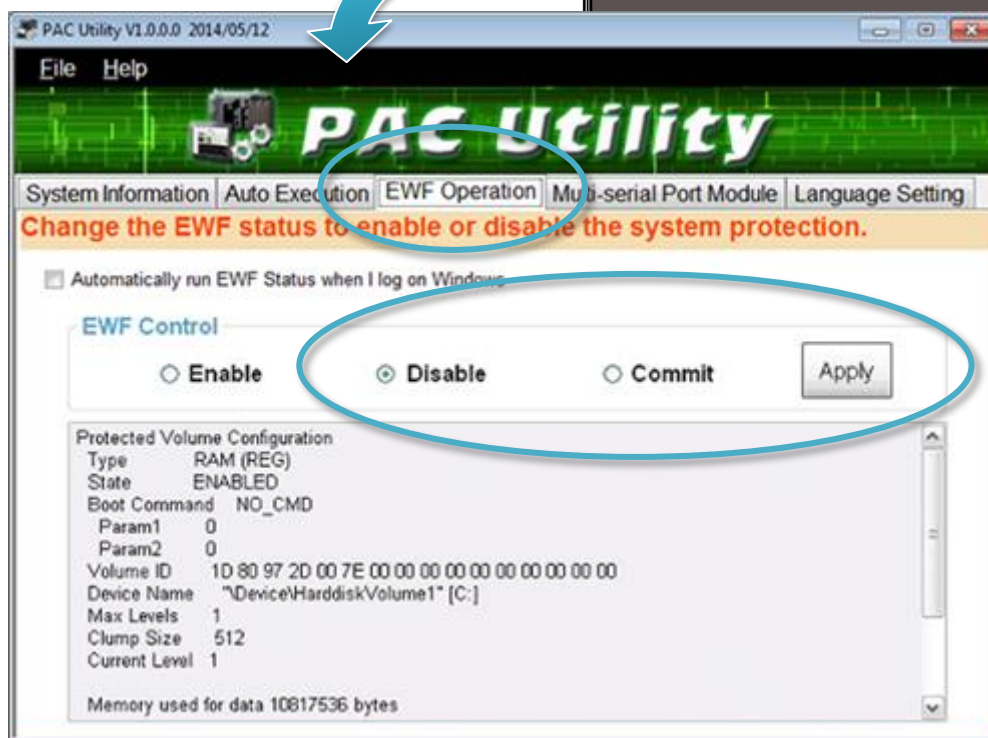
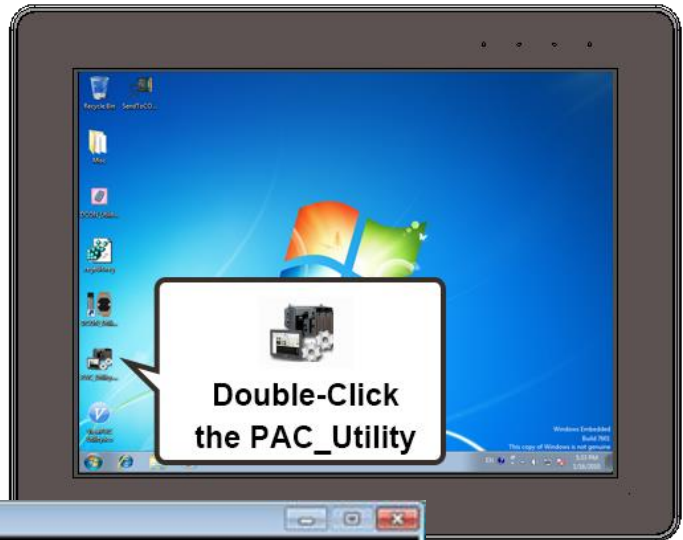
In cases of maintenance, the disk must be updated to your desired changes.

There are two way to use: one is using the commit function (We strongly recommend you to use this way), and the other contains three steps: (1) disabling EWF, (2) updating, and (3) re-enabling EWF.



## To disable the EWF

1. Click the **PAC Utility** shortcut on the desktop
2. Click the **EWF Operation** tab, select the **Disable** check box, and then click **Apply** button

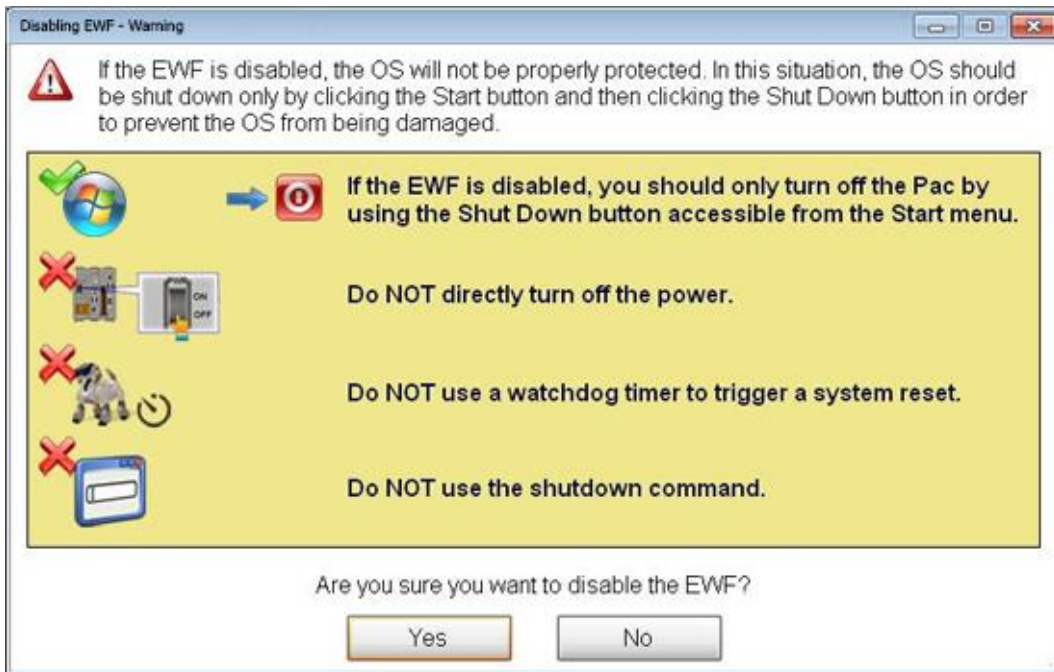



## Tips & Warnings

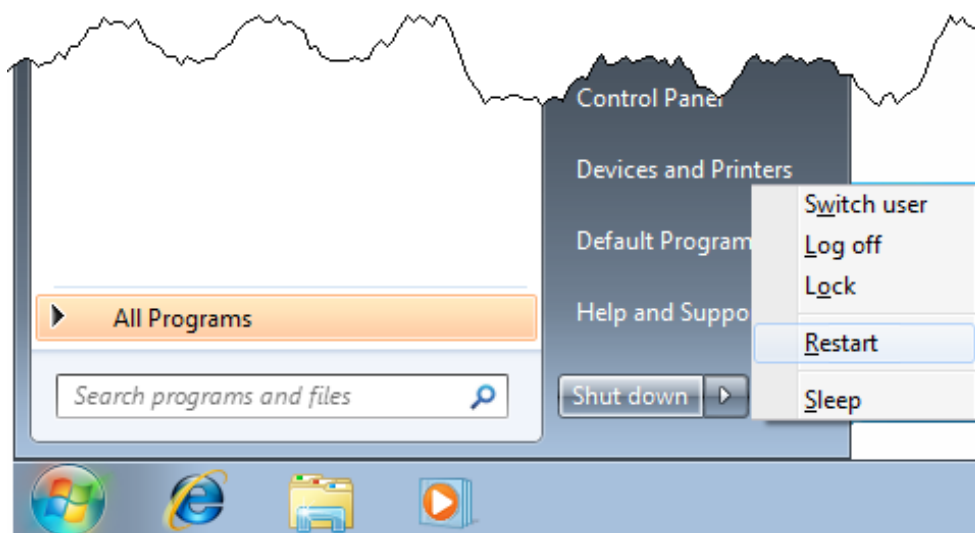


If EWF is disabled and iPPC suffers sudden power off, the operating system of iPPC may be damaged or incomplete.

3. In the pop-up dialog box, click **Yes** button

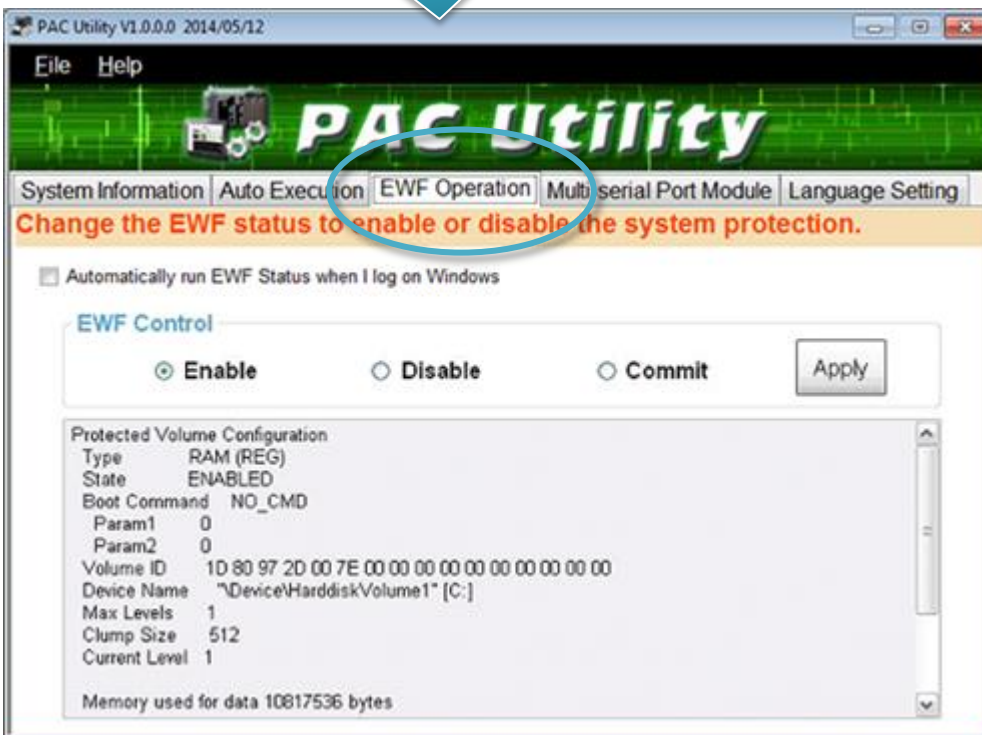
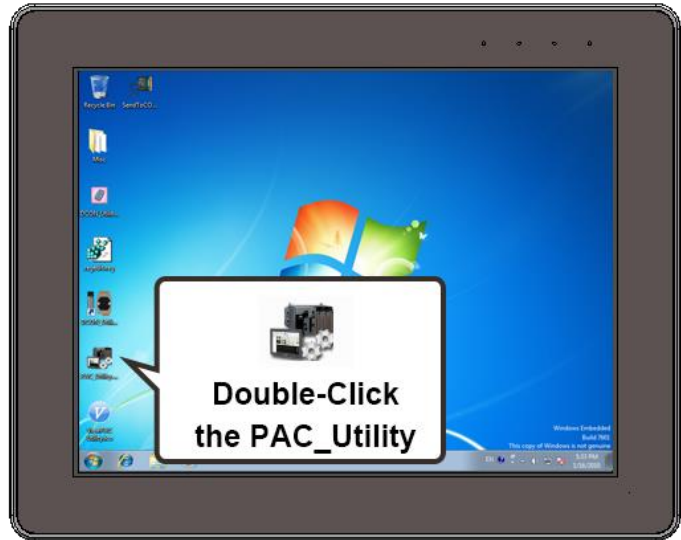


4. Click the **Start** button , click the arrow next to the **Shut Down** button, and then click **Restart**



## To enable the EWF

1. Click the **PAC Utility** shortcut on the desktop
2. Click the **EWF Operation** tab, select the **Enable** check box, and then click **Apply** button

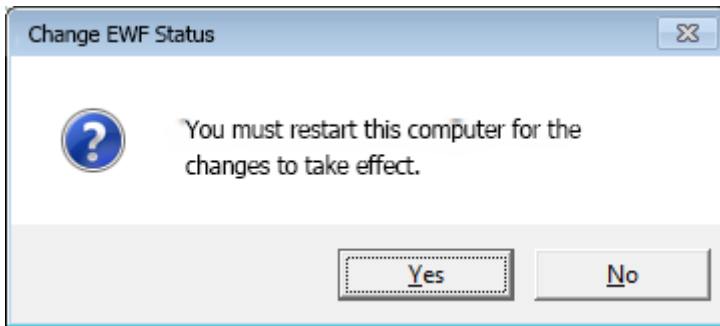



## Tips & Warnings

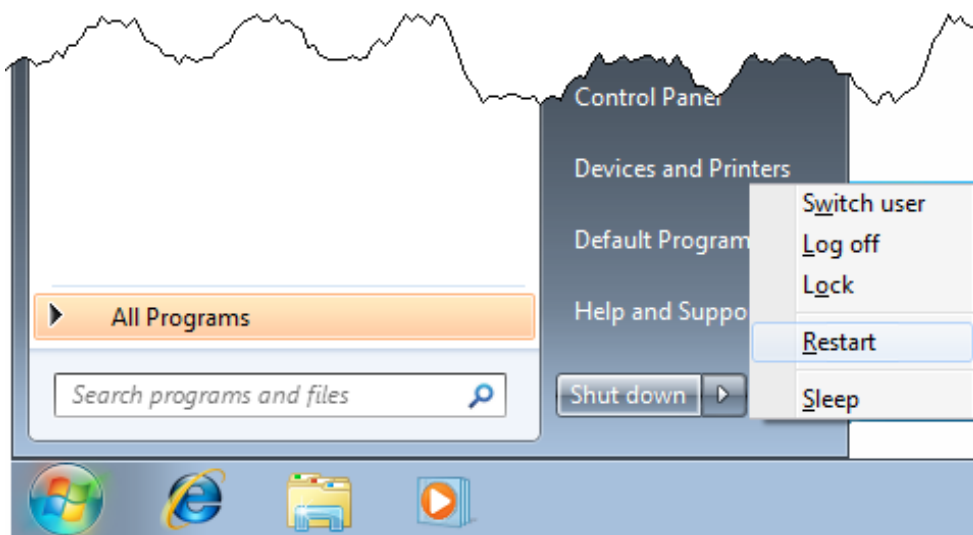


EWF only takes effect on hard drive C (where the operating system resides), it is recommended to download your programs to Compact Flash or USB-HDD. It'll prevent operating system from damages of illegal writing or sudden power off.

3. In the pop-up dialog box, click **Yes** button



4. Click the **Start** button , click the arrow next to the **Shut Down** button, and then click **Restart**



## How to use the EWF console application command-line tool

To control the status of EWF, use the EWF Manager Command “EWFMgr”. Windows Embedded Standard 7 includes the Enhanced Write Filter (EWF) console application command-line tool, Ewfmgr.exe.

- Check the EWF status:  
ewfmgr c:
- Enable the EWF:  
ewfmgr c: -enable (it is effective after rebooting.)
- Disable EWF:  
ewfmgr c: -commitanddisable –live

For more information about using EWF Manager Commands, please refer to **Manager Commands**

[http://msdn.microsoft.com/en-us/library/ms940853\(v=WinEmbedded.5\).aspx](http://msdn.microsoft.com/en-us/library/ms940853(v=WinEmbedded.5).aspx)

### Tips & Warnings

---



Only the disk drive (usually, c:\) that OS resides can use the feature of EWF

---

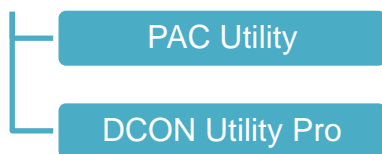
## 4. Tools and Tasks

This chapter provides a brief introduction of the iPPC service tools and its benefits.

There are several tools and utilities built-in and designed for use with iPPC. Some of these are pre-installed on iPPC and can work directly on iPPC, and some of these are supporting tools and can help you to manage the iPPC remotely on a PC.

Tools for working with PC can be found separately on the CD that was provided with the package or by downloading the latest version from ICP DAS web site.

CD:\ippc-wes7\tools\



## 4.1. PAC Utility

PAC Utility is a collection of software applications that enable management and configuration of iPPC system and features.

### 4.1.1. System Information

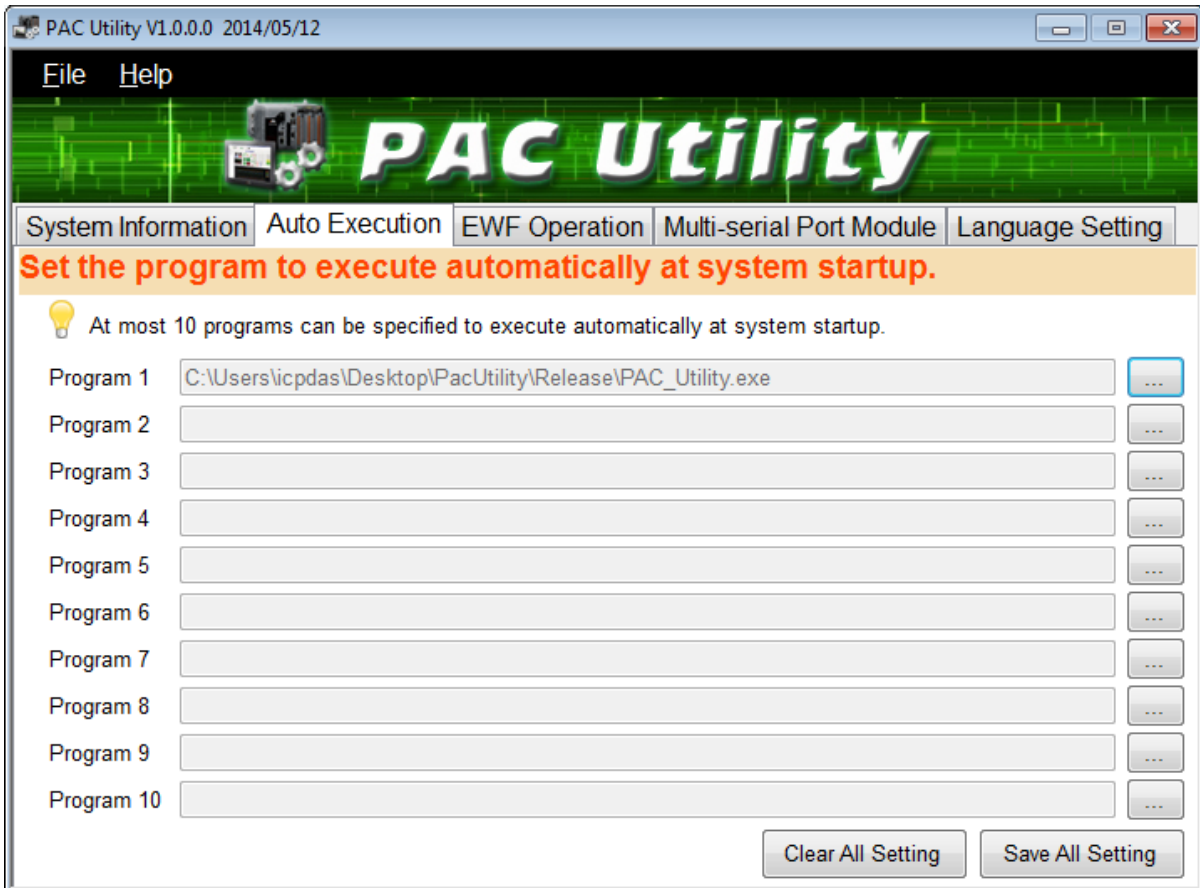
The System Information tab provides functions to monitor necessary device information of iPPC. The system information is the most important note of version control for upgrading system.





## 4.1.2. Auto Execution

The Auto Execution tab provides functions to configure programs running at iPPC startup, it allows users to configure ten execute files at most.



### Tips & Warnings



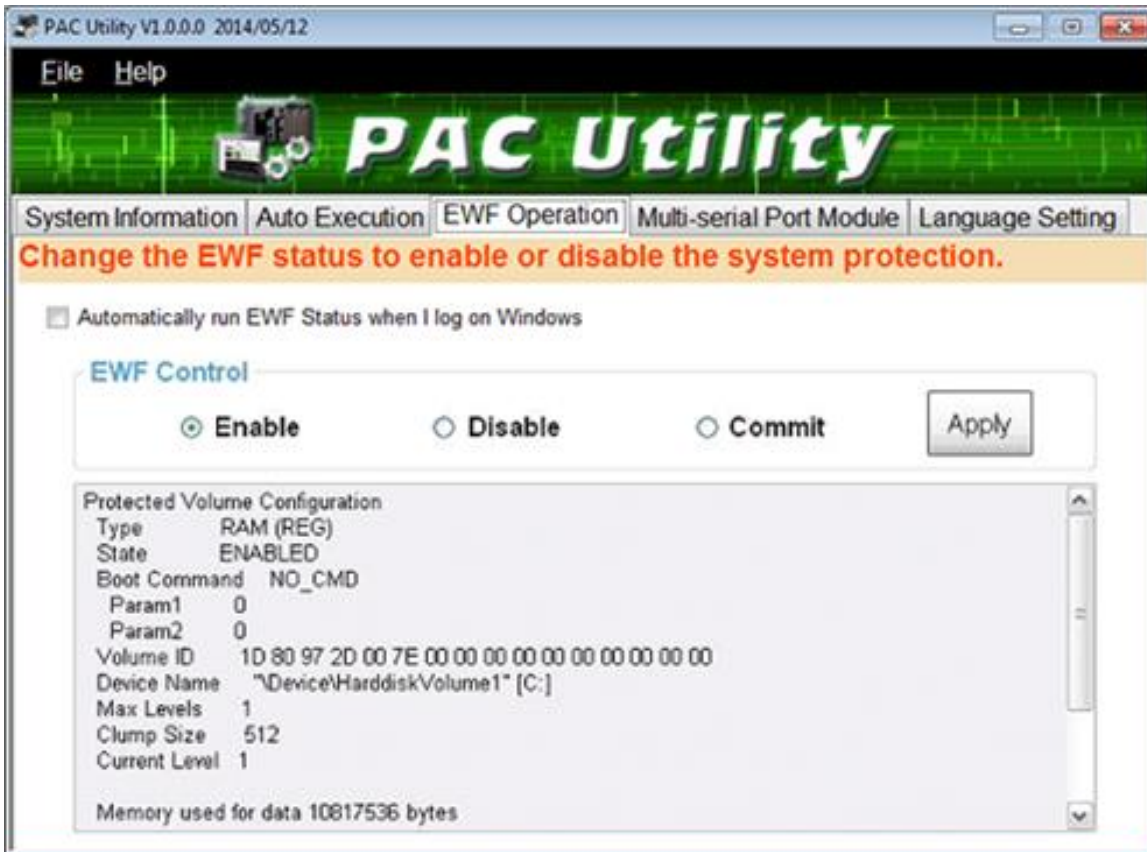
The allowed file types are .exe and .bat, and they are executed in order of program 1, program 2, etc.

The tab use to	How to use
Configure programs running at startup	Click on the Browse button and select the execute file which you want, and then click the <b>Save All Setting</b> button.



### 4.1.3. EWF Operation

The EWF Operation tab provides functions to configure EWF.



The tab use to	How to use
Enable/disable the EWF function	<p><b>Enable the EWF function:</b> Select the <b>Enable</b> option, and then click the <b>Apply</b> button.</p> <p><b>Disable the EWF function:</b> Select the <b>Disable</b> option, and then click the <b>Apply</b> button.</p>
Commit changes	Select the <b>Commit</b> option, and then click the <b>Apply</b> button.

## 4.1.4. Multi-serial Port Module



The Multi-serial port provides functions for installation of the RS-232/RS-422/RS-485 communication module driver.

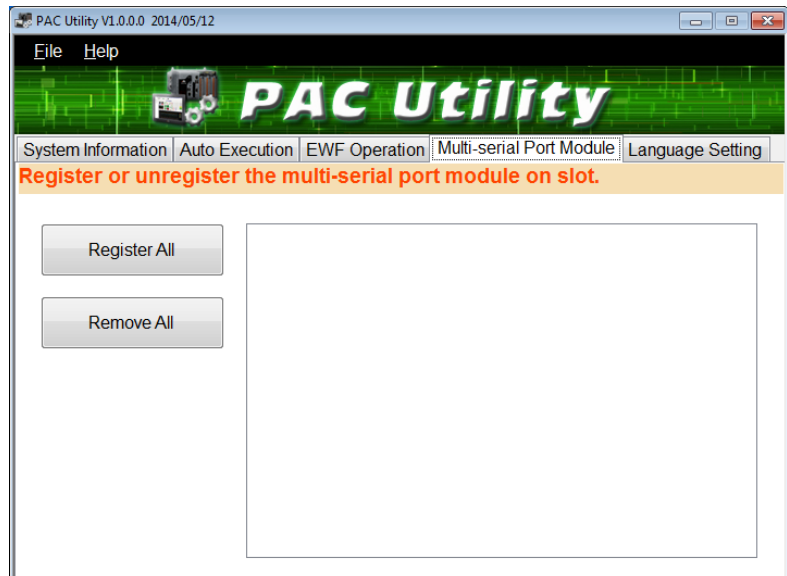


The table below shows the expansion RS-232/RS-422/RS-485 communication modules that are compatible with the iPPC.

Item	RS-232	RS-422/RS-485	Isolation	Connector
I-8112iW	2	-	2500 Vrms	DB-9 x 2
I-8114W	4	-	-	DB-37 x 1
I-8114iW	4	-	2500 Vrms	DB-37 x 1
I-8142iW	-	2	2500 Vrms	Terminator block x 1
I-8144iW	-	4	2500 Vrms	Terminator block x 1

The iPPC can be expanded to support up to 16 I/O modules. For more detailed information about these support modules, please refer to

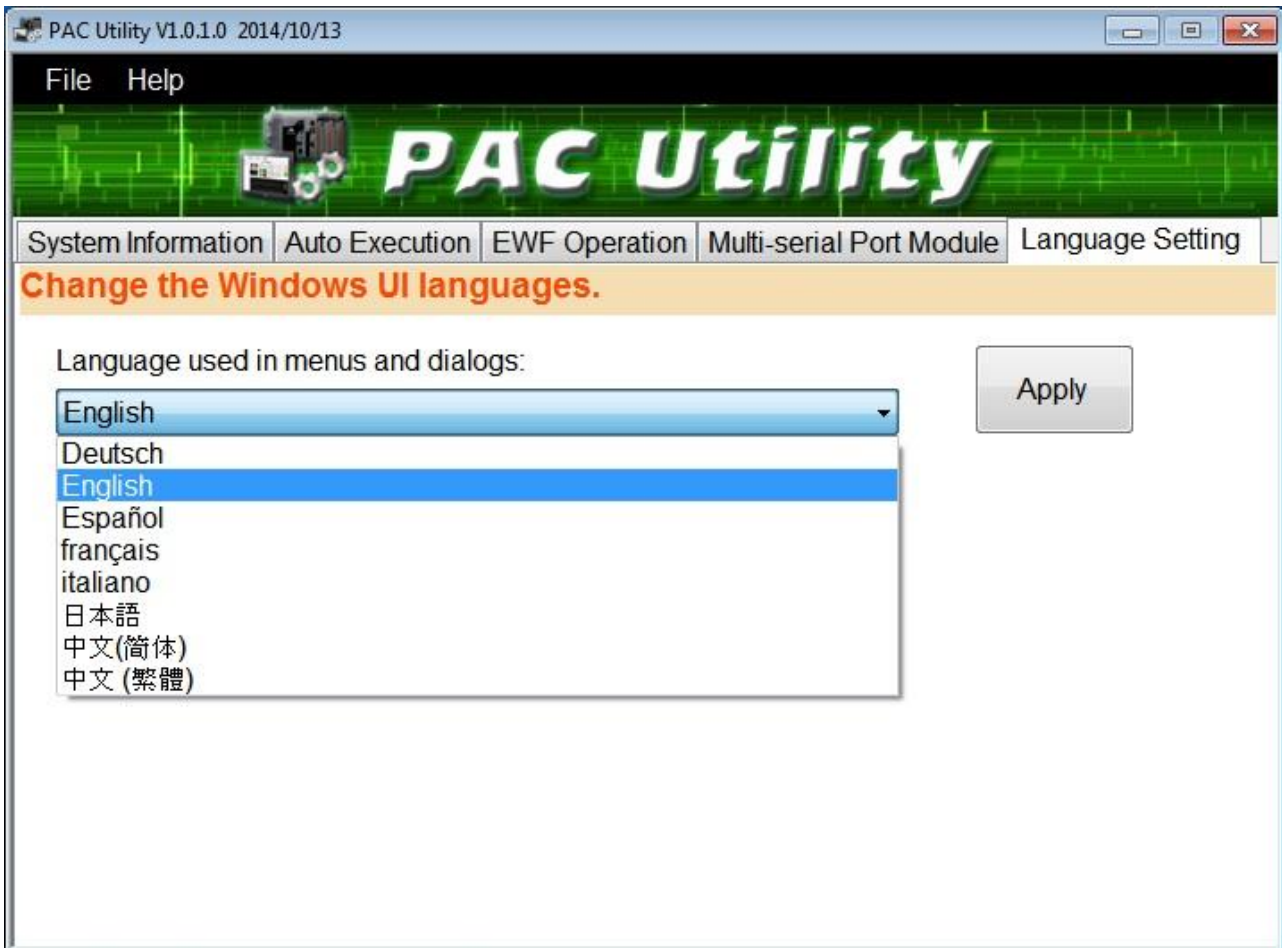
[http://www.icpdas.com/products/Remote\\_IO/i-8ke/selection\\_rs232\\_i8k.htm](http://www.icpdas.com/products/Remote_IO/i-8ke/selection_rs232_i8k.htm)



The tab use to	How to use
Register/remove the multi-serial port	<p><b>Register the multi-serial port:</b> Click the <b>Register All</b> button.</p> <p><b>Remove the multi-serial port:</b> Click the <b>Remove All</b> button.</p>

## 4.1.5. Language Setting

The Language Setting tab provides functions to change the user interface language.



The tab use to	How to use
Change the user interface language	Click on the drop down arrow and select your preferred language, and then click the <b>Apply</b> button.

## 4.2. DCON Utility Pro

The DCON Utility Pro is a toolkit that help user to search the network, easily to configure and test the I/O modules via the serial port (RS-232/485) or Ethernet port (using virtual com port).

For more information on how to use DCON Utility Pro to configure I/O modules, please refer to section 2.4. Using DCON Utility Pro to Configure I/O Modules.

# 5. Your First iPPC Program

This chapter provides a guided tour that describes the steps needed to set-up a development environment, download, install, configure for user programming with the iPPC.

Before writing your first program, ensure that you have the necessary development tool and the corresponding iPPC SDKs are installed on your system.

## Development Tools

---

iPPC is a WES7 based unit. WES7 is a mature embedded operating system which supports rapid development. Three standard development tools are list as follows which are highly integrated, with comprehensive support for developing applications of WES7 based iPPC.

- Visual Basic.net
- Visual C#
- Visual C++

## iPPC SDKs

---

The PAC SDK is a Software Development Kit (SDK) that contains C header files, C libraries and documents.

The iPPC SDK are classified by development tools that can be obtained from the CD that was provided with the package or by downloading the latest version from ICP DAS web site.

CD:\ippc-wes7\sdk\pacsdk

<ftp://ftp.icpdas.com/pub/cd/ippc-wes7/sdk/pacsdk/>

## 5.1. Your First iPPC Program in VB.NET

The best way to learn programming with iPPC is to actually create a iPPC program.

The example below will guide you through creating this simple program in VB.net and running them on iPPC.

To create a demo program with VB.NET that includes the following main steps:

1. Create a new project
2. Specify the path of the PAC reference
3. Add the control to the form
4. Add the event handling for the control
5. Upload the application to iPPC
6. Execute the application on iPPC

All main steps will be described in the following subsection.

## 5.1.1. Create a New Project

The Visual VB.net project template is a composite control that you use in this example creates a new project with this user control.

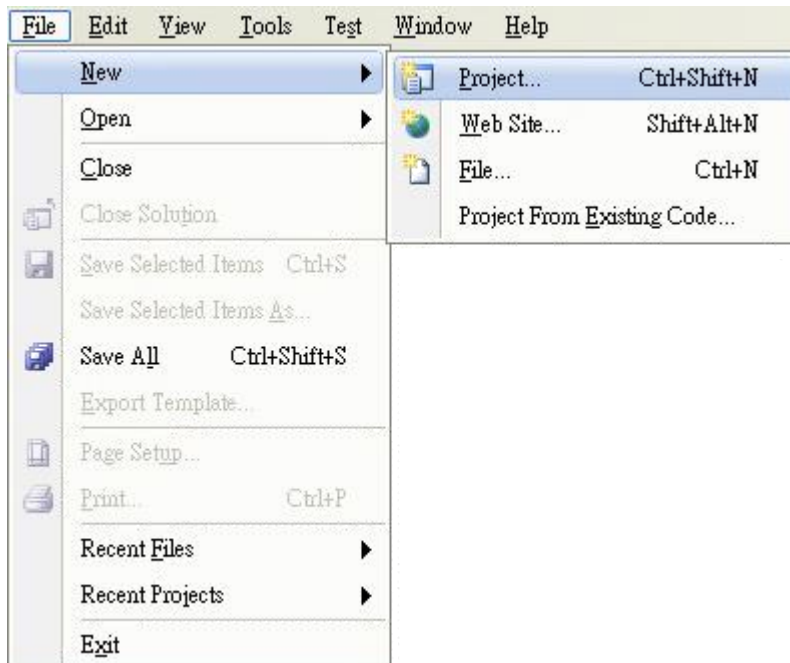
### To create a new project

#### 1. Run the **Visual Studio 2008**

Visual Studio 2008



#### 2. On the **File** menu, point to **New**, and then click **Project**

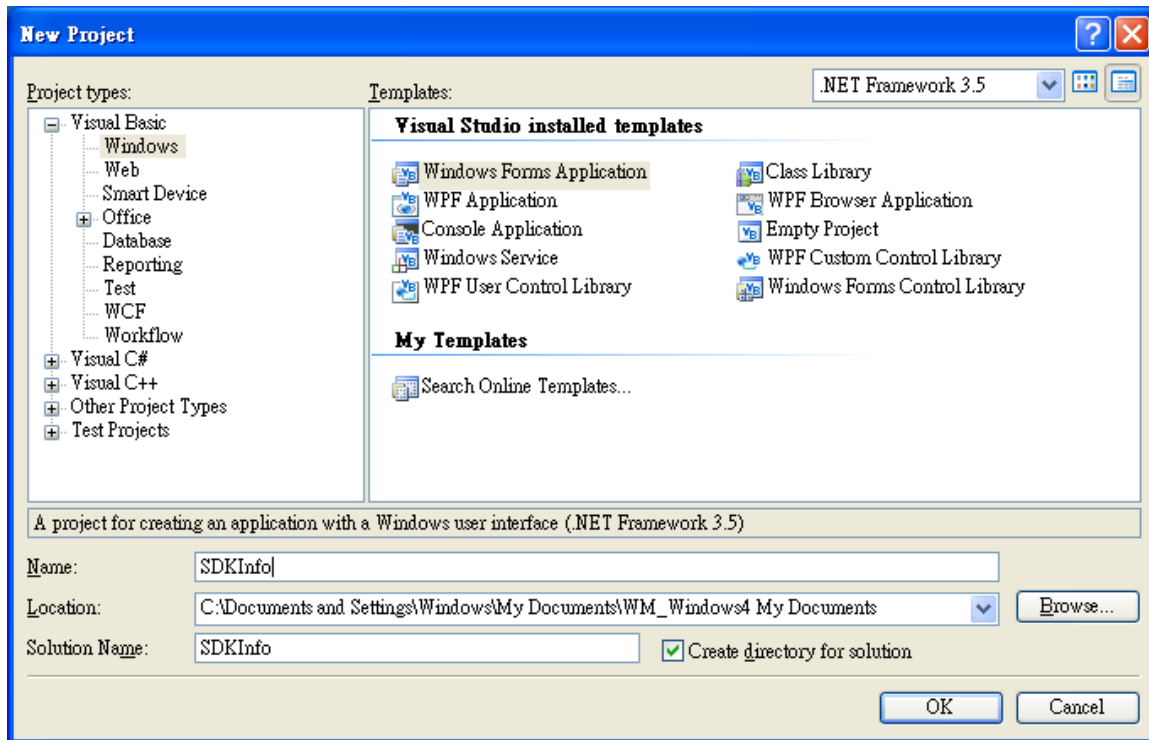


3. In the **Project types** pane, expand **Visual Basic**, and then click **Windows**

4. In the **Templates** pane, click **Windows Forms Application**

5. Type a name in the **Name** field, and then click **OK** button

Here we will enter the name “SDKInfo” and a different location for the project if you wish





## 5.1.2. Specify the Path of PAC Reference

The PAC SDK provides a complete solution to integrate with iPPC and it's 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.

### To specify the path of PAC reference

1. Get the **PACNET.dll** and copy it to the project folder

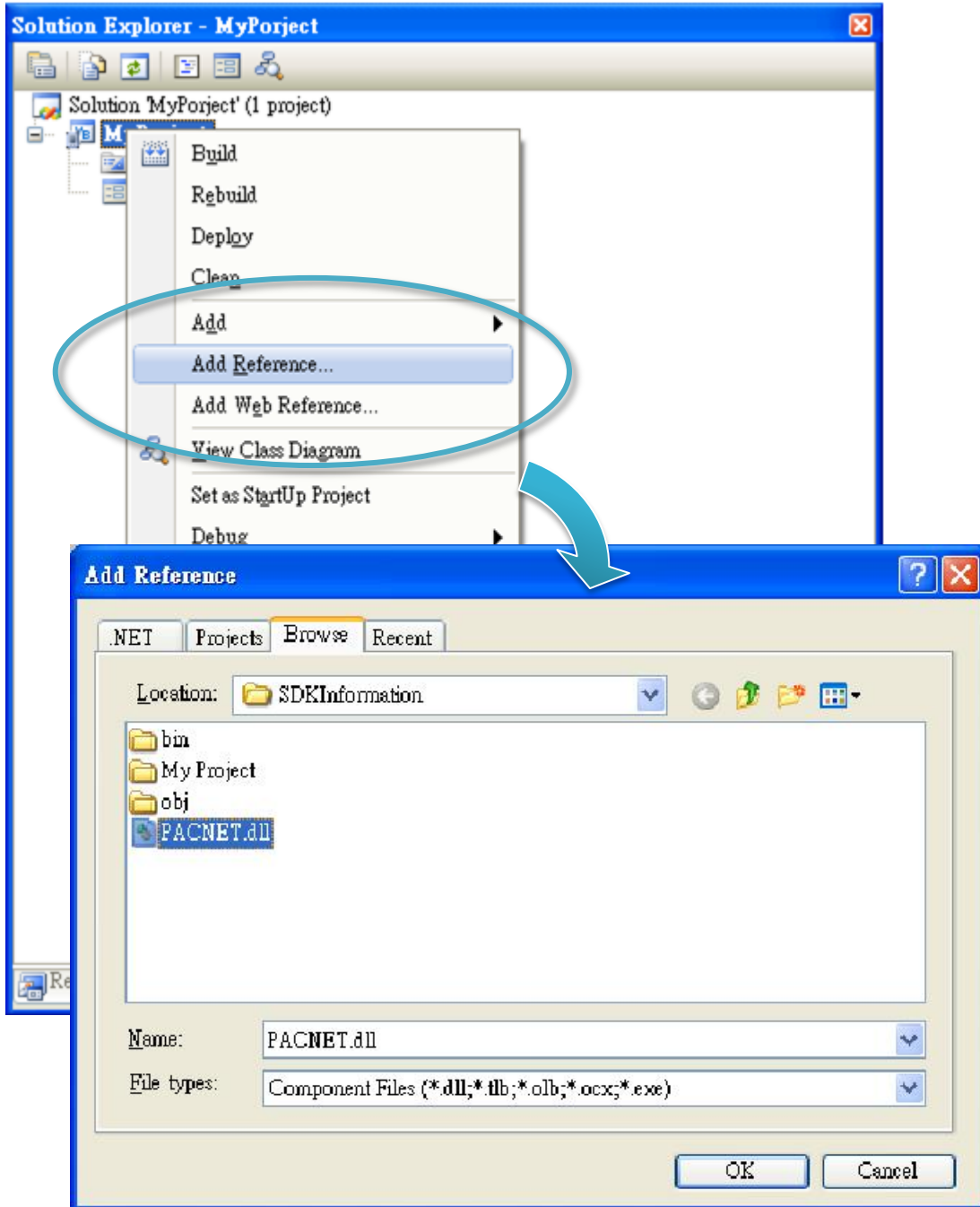
The PACNET.dll can be obtained separately on the CD that was provided with the package or by downloading the latest version from ICP DAS web site.

CD:\ippc-wes7\sdk\pacsdk\pacnet\

<ftp://ftp.icpdas.com/pub/cd/ippc-wes7/sdk/pacsdk/pacnet/>



2. In the **Solution Explorer**, right-click the References node, and then click **Add Reference...**
3. Click the **Browse** tab, and then select the **PACNET.dll**



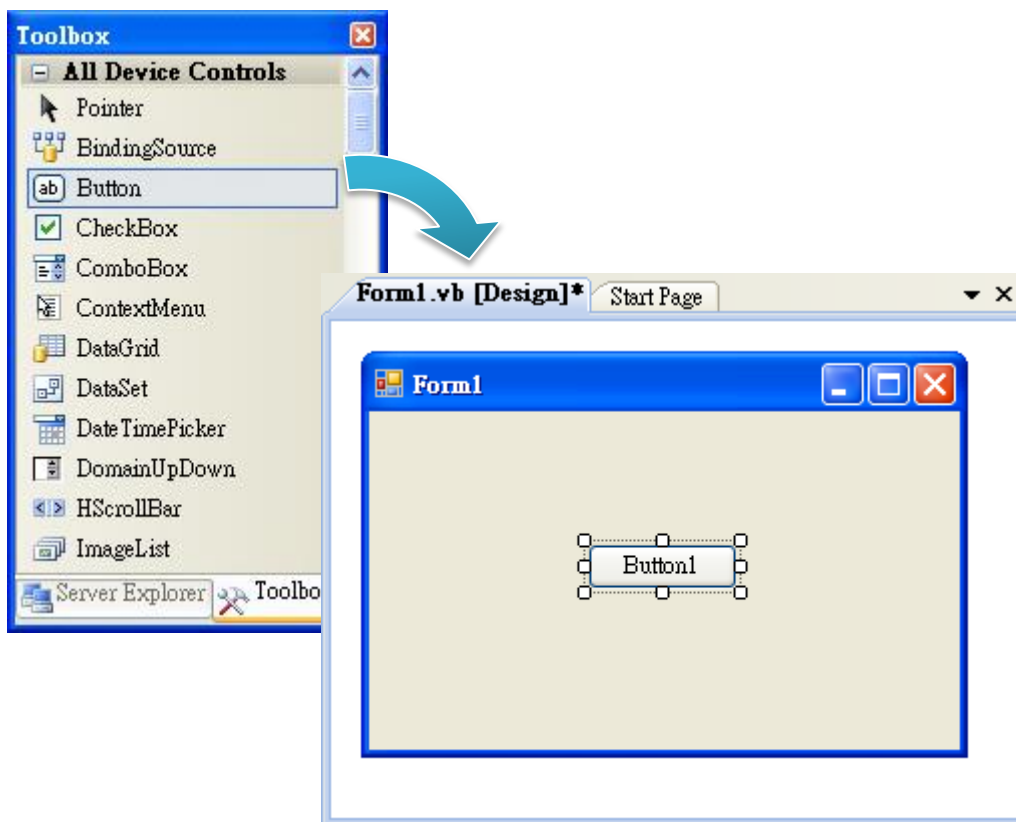
### 5.1.3. Add the Control to the Form

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.

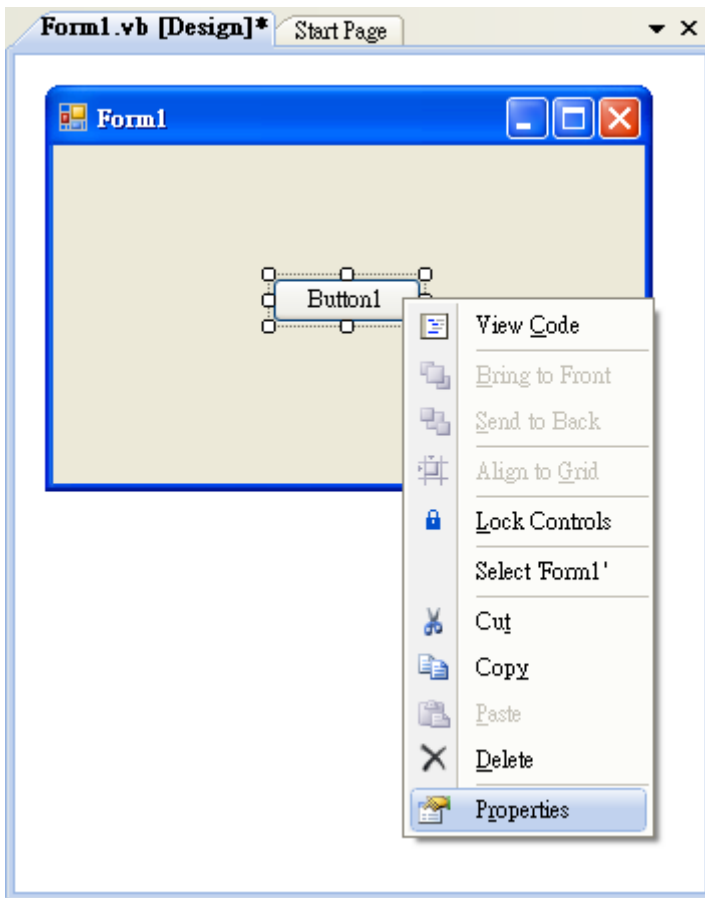
After you add a control to your form, you can use the Properties window to set its properties, such as background color and default text. The values that you specify in the Properties window are the initial values that will be assigned to that property when the control is created at run time.

#### To add the control to the form

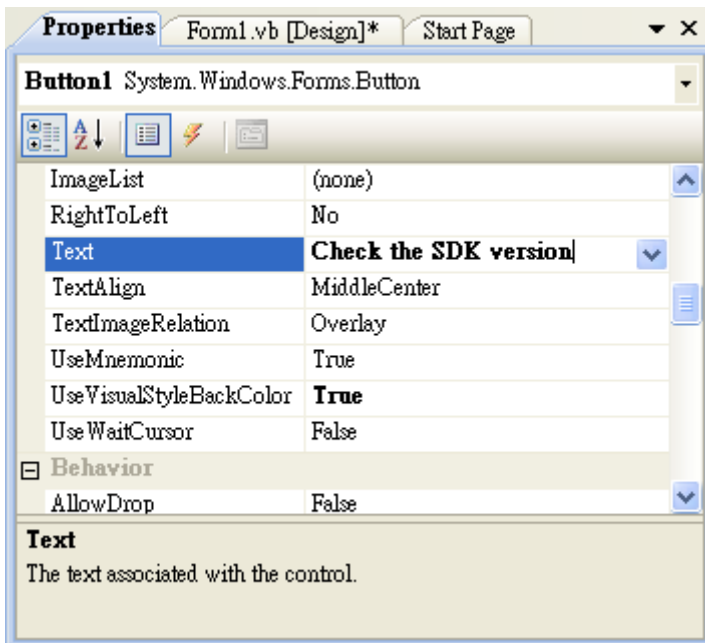
1. From the **Toolbox**, drag a **Button** control onto the form



2. Right-click the **Button** control, and then click **Properties**



3. In the **Properties** window, type **Check the SDK version** in the **Text** item, and press **ENTER**

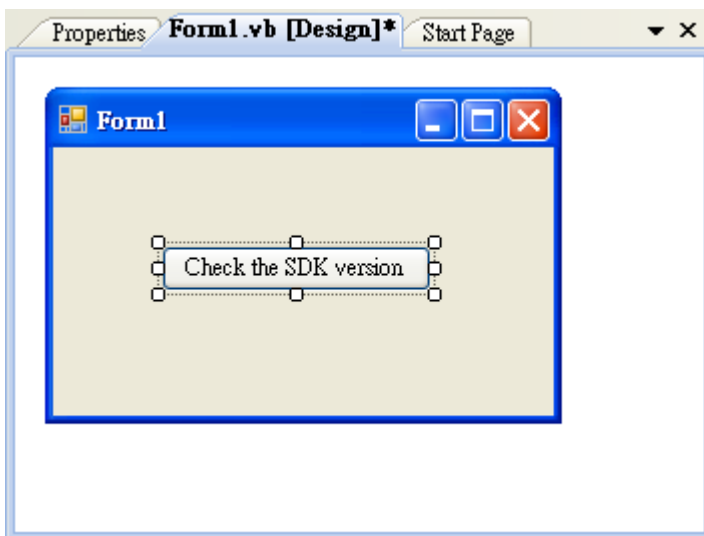


## 5.1.4. Add the Event Handling for the Control

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.

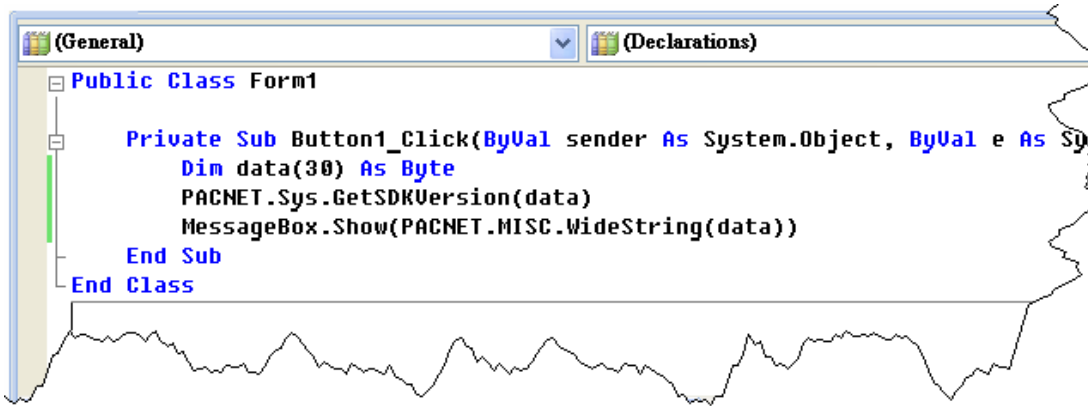
### To add the event handling for the control

1. Double-click the **button** on the form



## 2. Inserting the following code

```
Dim data(30) As Byte
PACNET.Sys.GetSDKVersion(data)
MessageBox.Show(PACNET.MISC.WideString(data))
```



The screenshot shows a Visual Studio code editor window with two tabs: "(General)" and "(Declarations)". The code is written in a class named "Form1". The code is as follows:

```
Public Class Form1
    Private Sub Button1_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles Button1.Click
        Dim data(30) As Byte
        PACNET.Sys.GetSDKVersion(data)
        MessageBox.Show(PACNET.MISC.WideString(data))
    End Sub
End Class
```

## Tips & Warnings

---



The "PACNET" of "using PACNET" is case-sensitive.

---

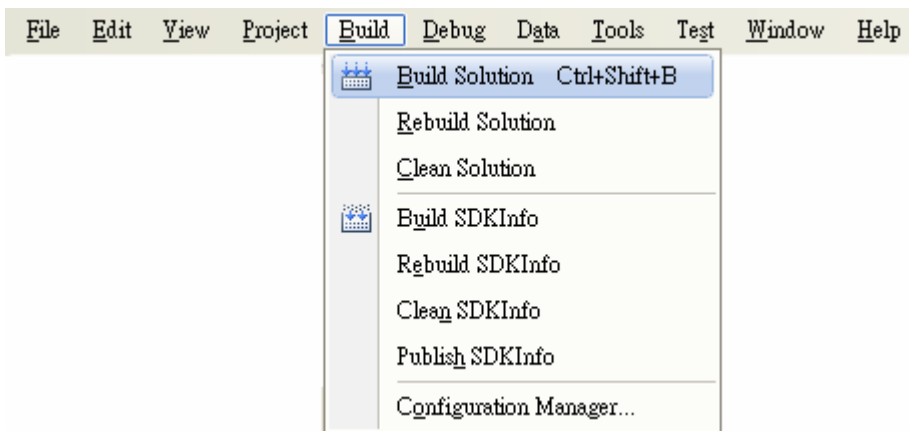
## 5.1.5. Upload the Application to iPPC

iPPC supports FTP server service. You can upload files to iPPC or download files from a public FTP server.



### To upload the application to iPPC

1. On the **Build** menu, click **Build Solution**



2. Open the **browser** and type the **IP address** of iPPC

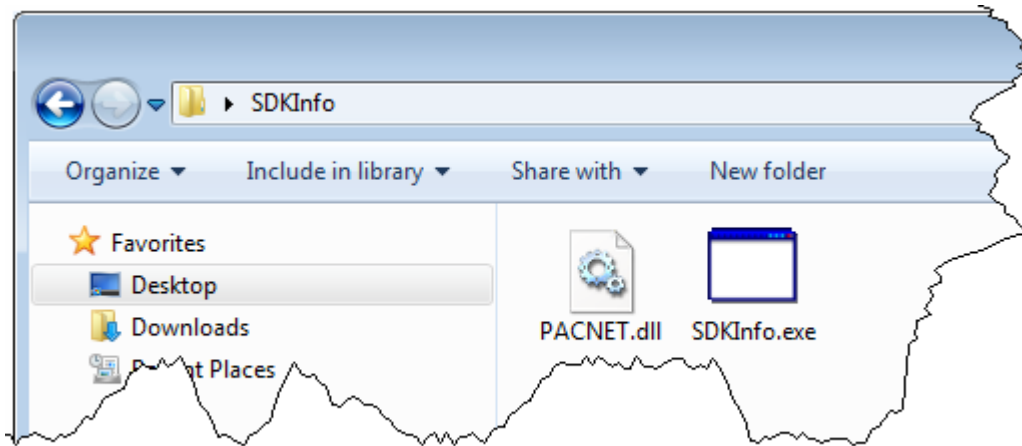
3. Upload the SDKInfo.exe application and the corresponding PACNET.dll files to iPPC

### Tips & Warnings

---



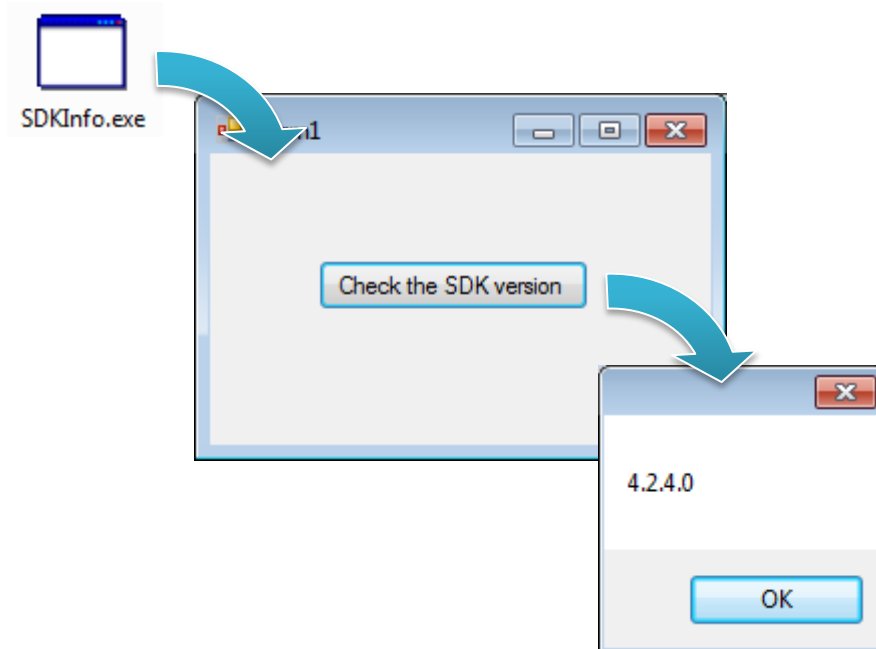
For applications programming in C# and VB.net with .net compact framework, when executing these application on iPPC, the corresponding PACNET.dll must be in the same directory as the .exe file.





## 5.1.6. Execute the Application on iPPC

After uploading the application to iPPC, you can just double-click it to execute it.



## 5.2. Your First iPPC Program in C#

The best way to learn programming with iPPC is to actually create a iPPC program.

The example below will guide you through creating this simple program in C# and running them on iPPC.

To create a demo program with C# that includes the following main steps:

1. Create a new project
2. Specify the path of the PAC reference
3. Add the control to the form
4. Add the event handling for the control
5. Upload the application to iPPC
6. Execute the application on iPPC

All main steps will be described in the following subsection.

## 5.2.1. Create a New Project

The C# project template is a composite control that you use in this example creates a new project with this user control.

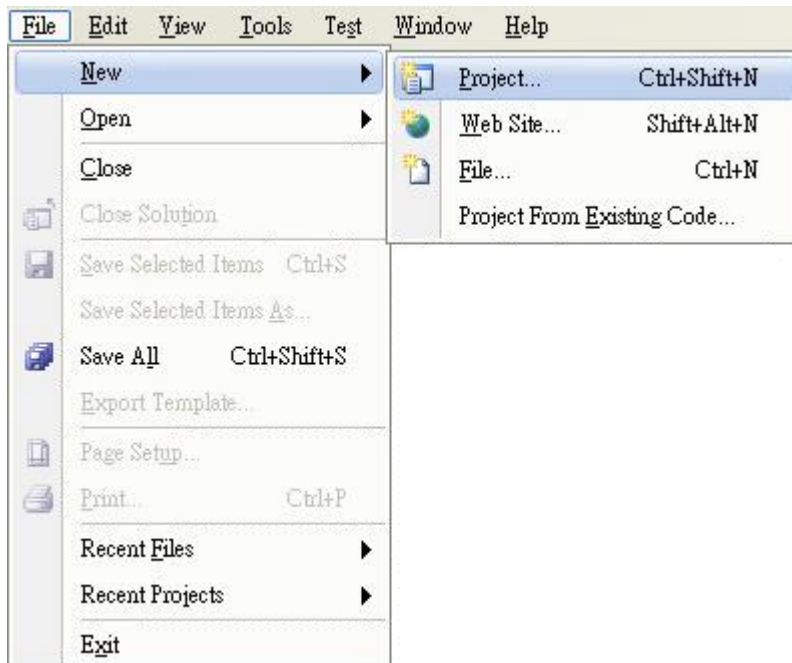
### To create a new project

#### 1. Run the **Visual Studio 2008**

Visual Studio 2008



#### 2. On the **File** menu, point to **New**, and then click **Project**

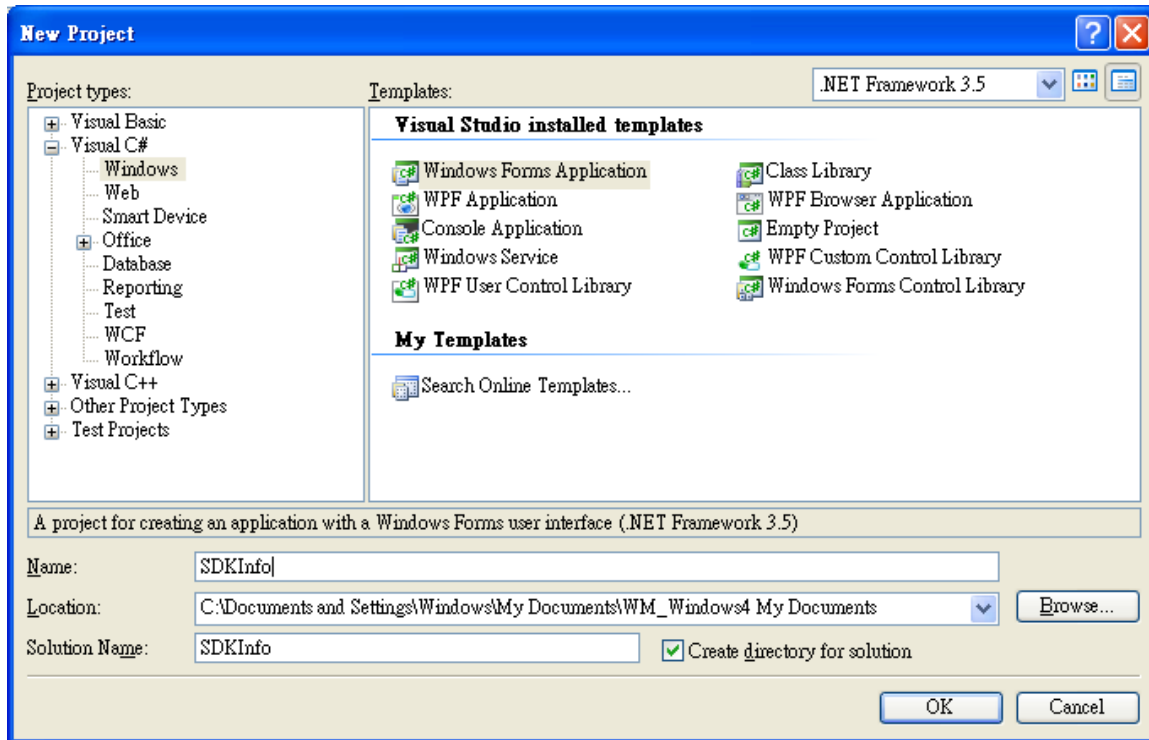


3. In the **Project types** pane, expand **Visual C#**, and then click **Windows**

4. In the **Templates** pane, click **Windows Forms Application**

5. Type a name in the **Name** field, and then click **OK** button

Here we will enter the name “SDKInfo” and a different location for the project if you wish



## 5.2.2. Specify the Path of PAC Reference

The PAC SDK provides a complete solution to integrate with iPPC and it's 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.

### To specify the path of PAC reference

1. Get the **PACNET.dll** and copy it to the project folder

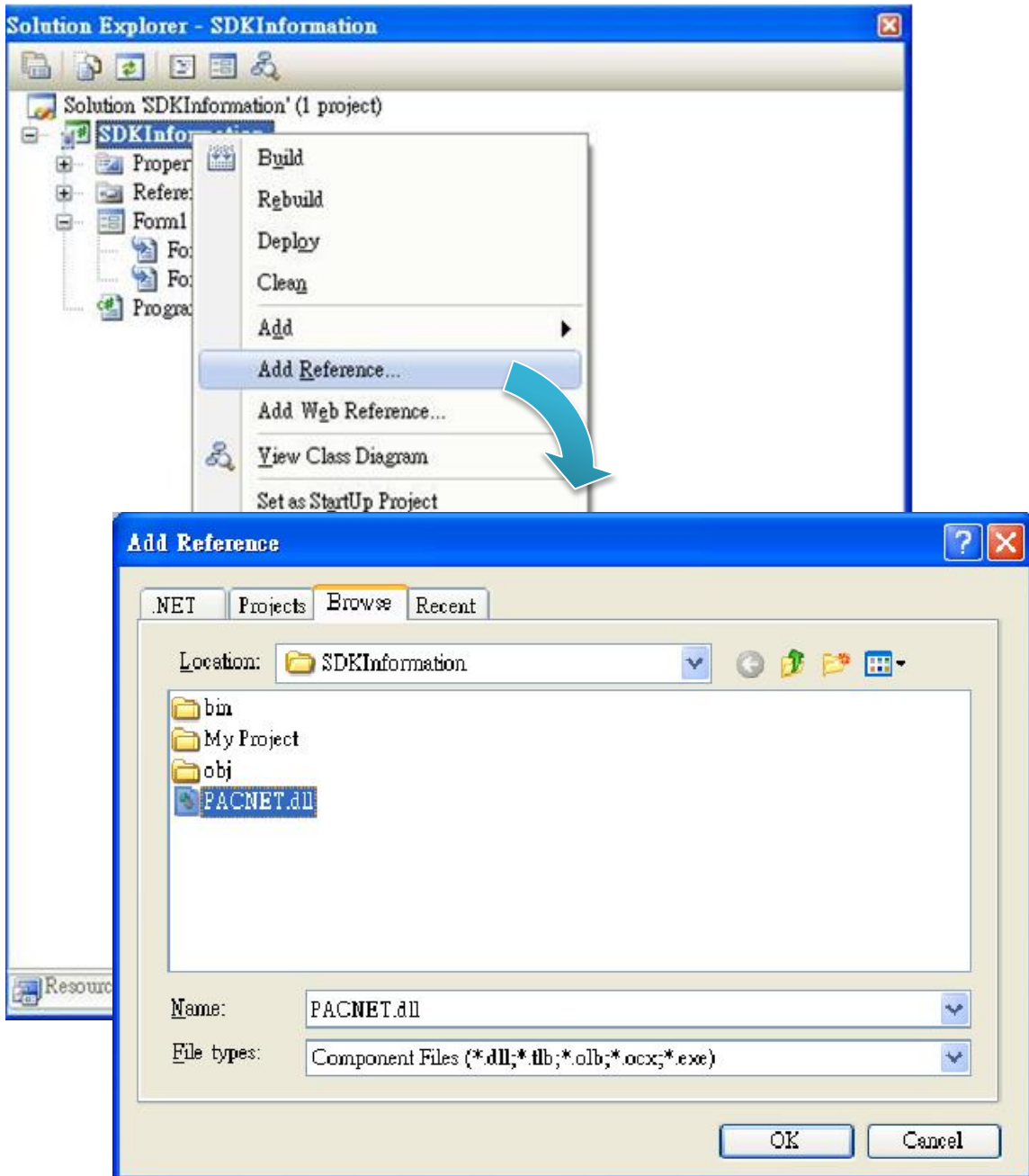
The PACNET.dll can be obtained separately on the CD that was provided with the package or by downloading the latest version from ICP DAS web site.

CD:\ippc-wes7\sdk\pacsdk\pacnet\

<ftp://ftp.icpdas.com/pub/cd/ippc-wes7/sdk/pacsdk/pacnet/>



2. In Solution Explorer, right-click the References node, and then click Add Reference...
3. Select Browse tab and add the PACNET.dll



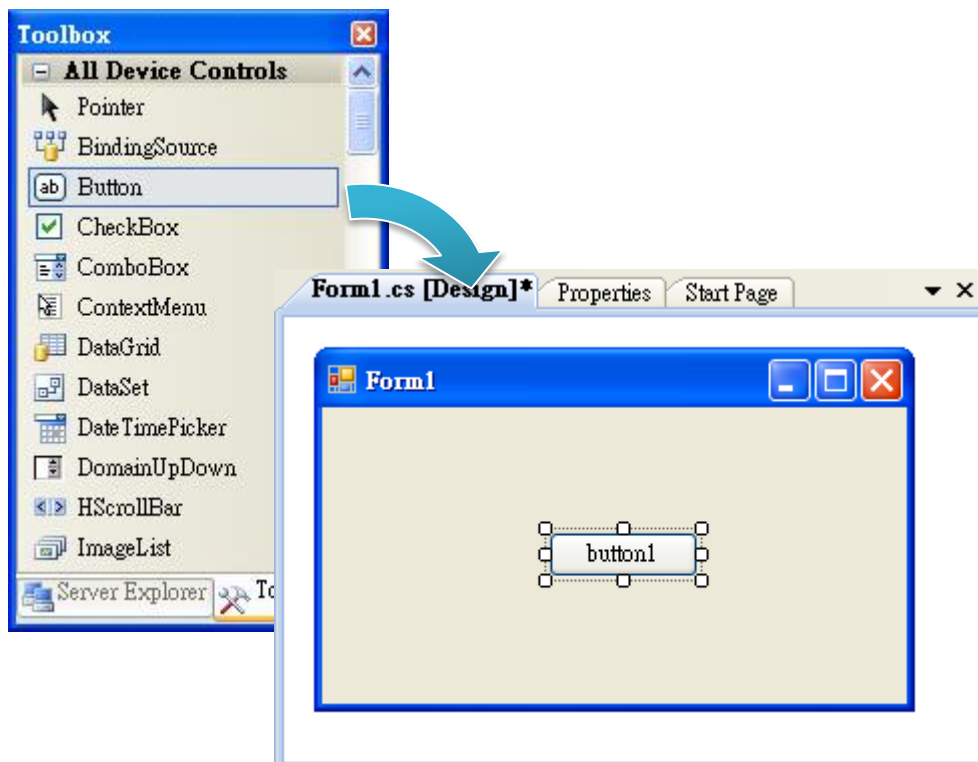
### 5.2.3. Add the Control to the Form

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.

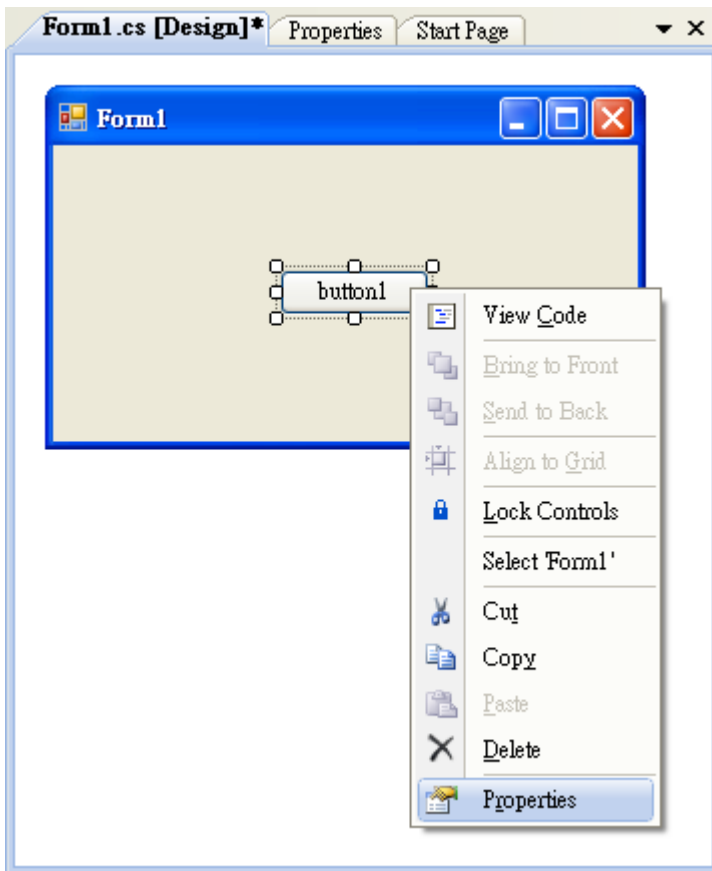
After you add a control to your form, you can use the Properties window to set its properties, such as background color and default text. The values that you specify in the Properties window are the initial values that will be assigned to that property when the control is created at run time.

#### To add the control to the form

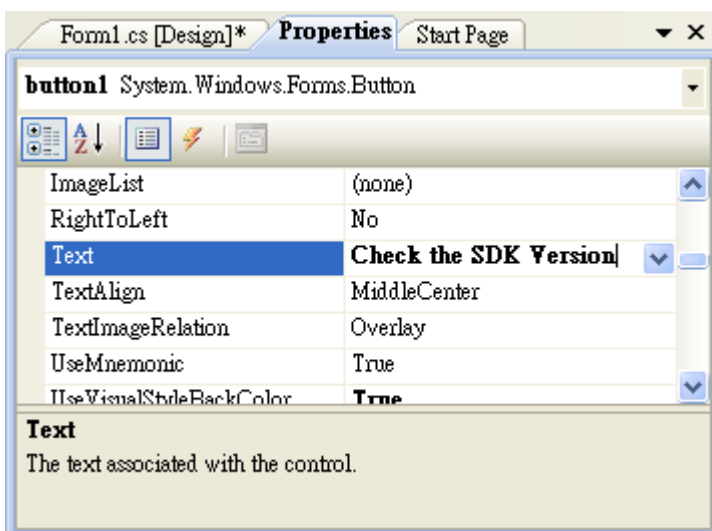
1. From the **Toolbox**, drag a **Button** control onto the form



2. Right-click the **Button** control, and then click **Properties**



3. In the **Properties** window, type **Check the SDK version** in the **Text** item, and press **ENTER**



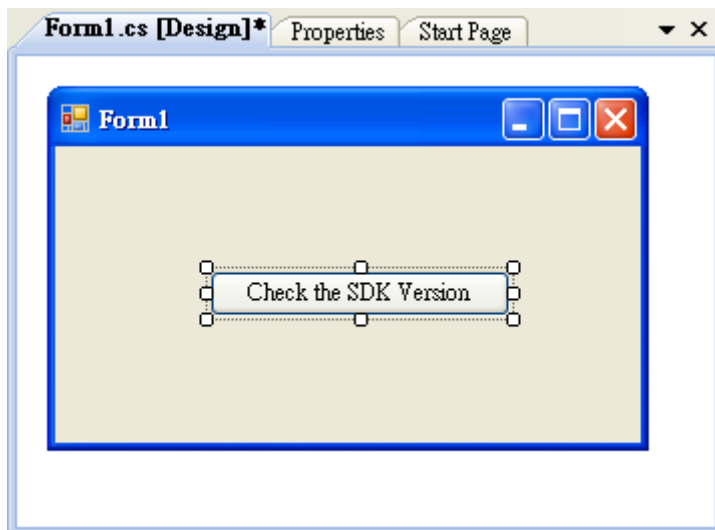


## 5.2.4. Add the Event Handling for the Control

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.

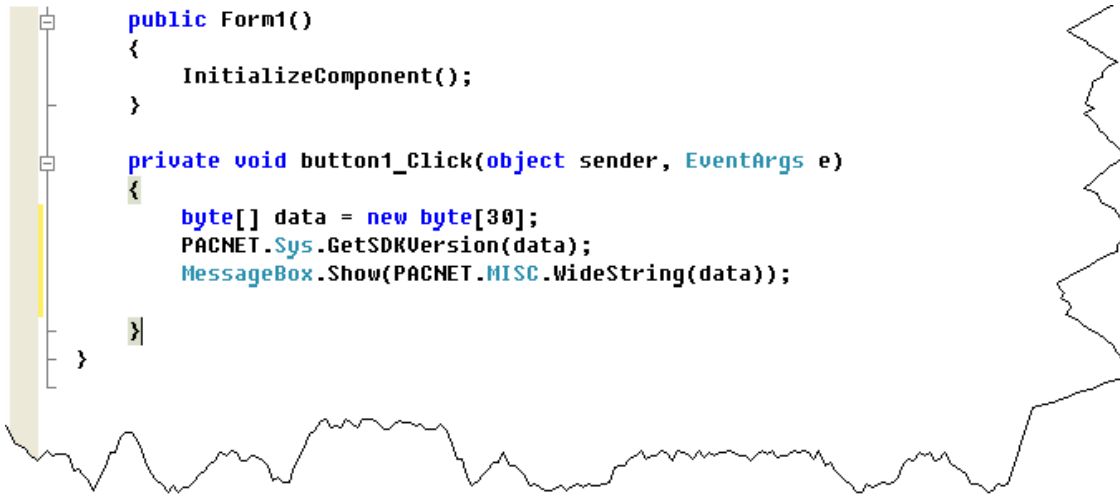
### To add the event handling for the control

1. Double-click the **button** on the form



## 2. Inserting the following code

```
byte []data = new byte[30];  
PACNET.Sys.GetSDKVersion(data);  
MessageBox.Show(PACNET.MISC.WideString(data));
```



```
public Form1()  
{  
    InitializeComponent();  
}  
  
private void button1_Click(object sender, EventArgs e)  
{  
    byte[] data = new byte[30];  
    PACNET.Sys.GetSDKVersion(data);  
    MessageBox.Show(PACNET.MISC.WideString(data));  
}
```

## Tips & Warnings

---



The “PACNET” of “using PACNET” is case- sensitive.

---

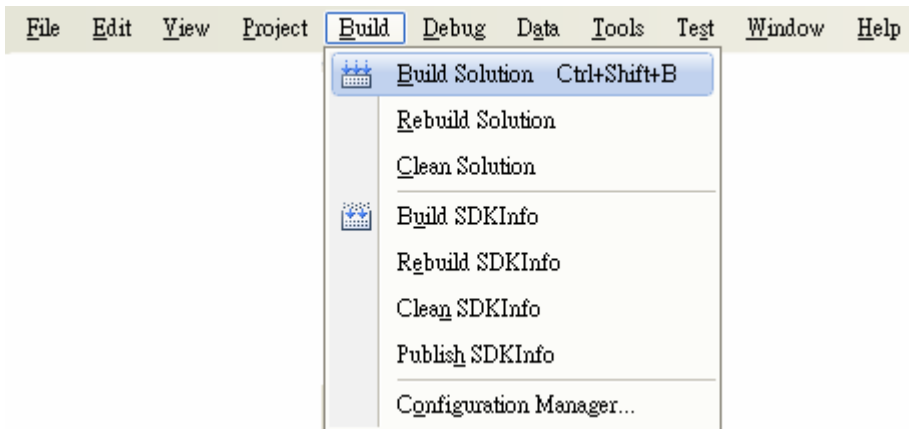
## 5.2.5. Upload the Application to iPPC

iPPC supports FTP server service. You can upload files to iPPC or download files from a public FTP server.



### To upload the application to iPPC

1. On the **Build** menu, click **Build Solution**



2. Open the **browser** and type the **IP address** of iPPC

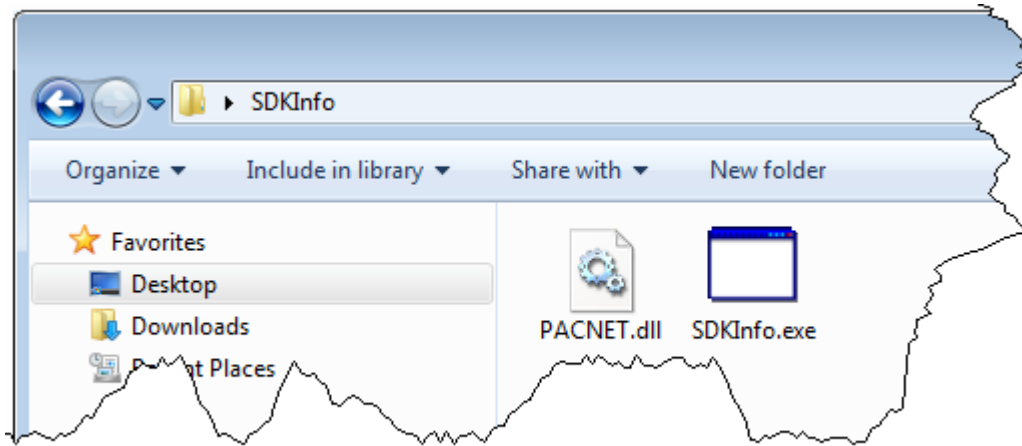
3. Upload the SDKInfo.exe application and the corresponding PACNET.dll files to iPPC

### Tips & Warnings

---

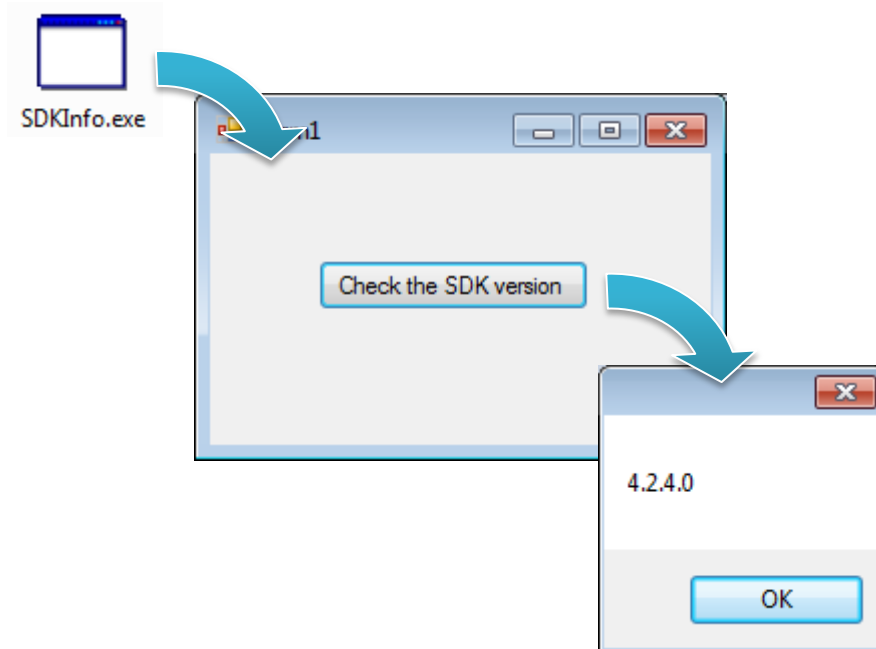


For applications programming in C# and VB.net with .net compact framework, when executing these application on iPPC, the corresponding PACNET.dll must be in the same directory as the .exe file.



## 5.2.6. Execute the Application on iPPC

After uploading the application to iPPC, you can just double-click it to execute it.



## 5.3. Your First iPPC Program in Visual C++

The best way to learn programming with iPPC is to actually create a iPPC program.

The example below will guide you through creating this simple program in Visual C++ and running them on iPPC.

To create a demo program with Visual C++ that includes the following main steps:

1. Create a new project
2. Specify the path of the iPPC reference
3. Add the control to the form
4. Add the event handling for the control
5. Upload the application to iPPC
6. Execute the application on iPPC

All main steps will be described in the following subsection.

## 5.3.1. Create a New Project

The Visual C++ project template is a composite control that you use in this example creates a new project with this user control.

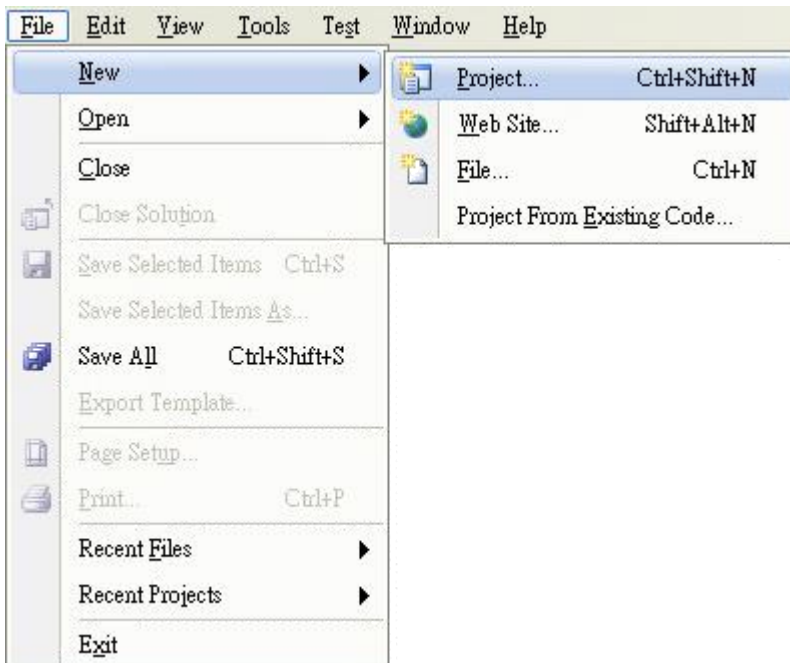
### To create a new project

#### 1. Run the **Visual Studio 2008**

Visual Studio 2008

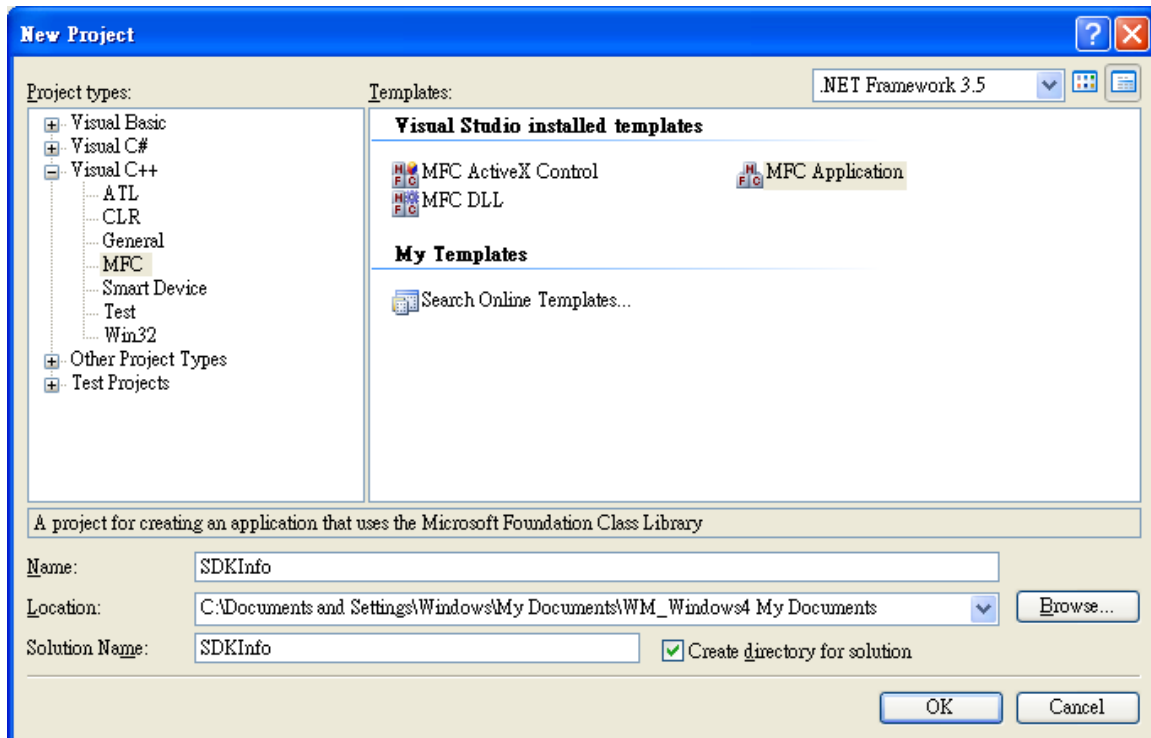


#### 2. On the **File** menu, point to **New**, and then click **Project**



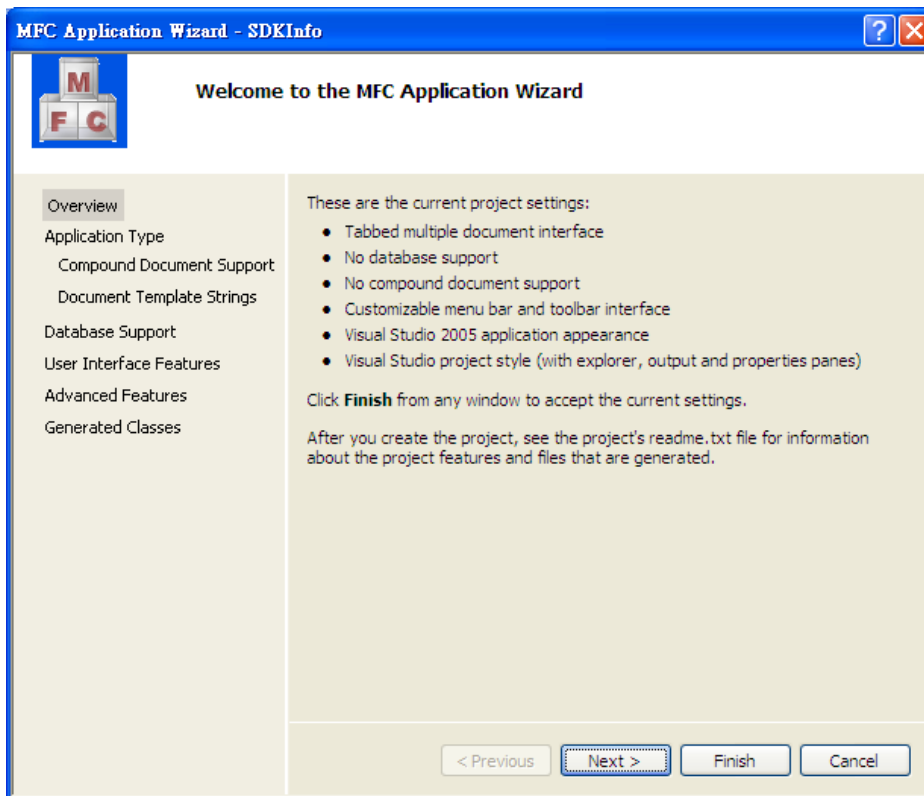
3. In the **Project types** pane, expand **Visual C++**, and then click **MFC**
4. In the **Templates** pane, click **MFC Application**
5. Type a name in the Name field, and then click **OK**

Here we will enter the name “SDKInfo” and a different location for the project if you wish

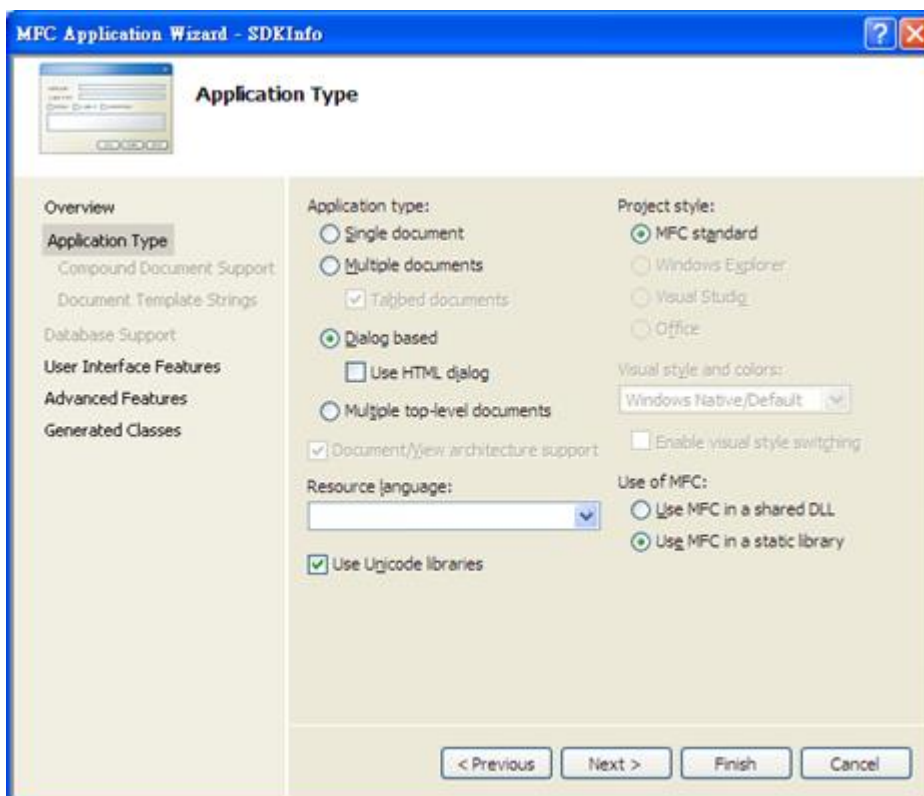




6. On the first page of the wizard, click **Next >**



7. On the next page of the wizard, select **Dialog based**, select **Use MFC in a static library**, and then click **Finish**



## 5.3.2. Specify the Path of the PAC Reference

The PAC SDK provides a complete solution to integrate with iPPC and it's 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.

### To specify the path of PAC reference

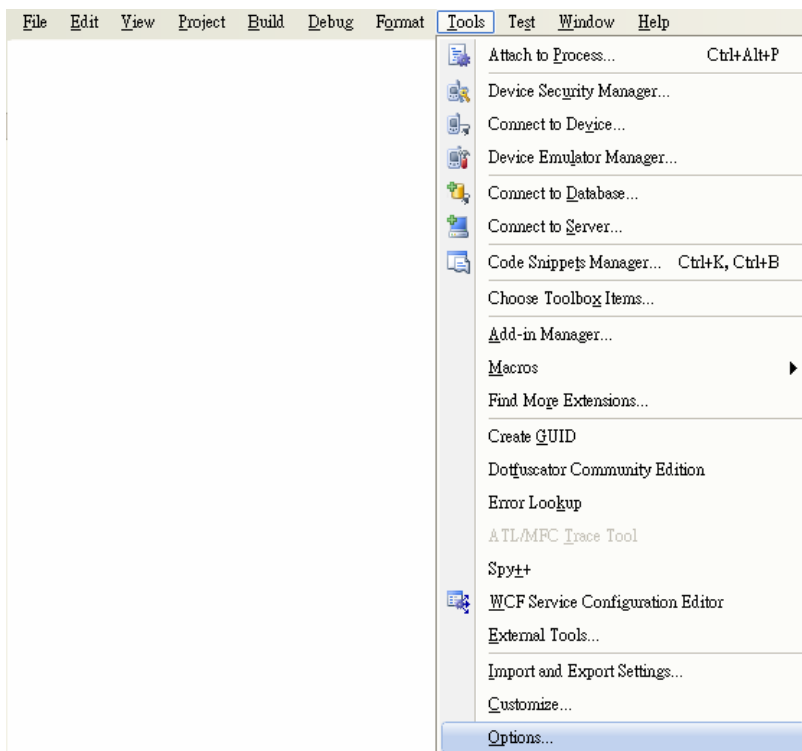
1. Get the **PACSDK.H** and **PACSDK.lib**, and copy them to the project folder

The PACSDK.H and PACSDK.lib can be obtained separately on the CD that was provided with the package or by downloading the latest version from ICP DAS web site.

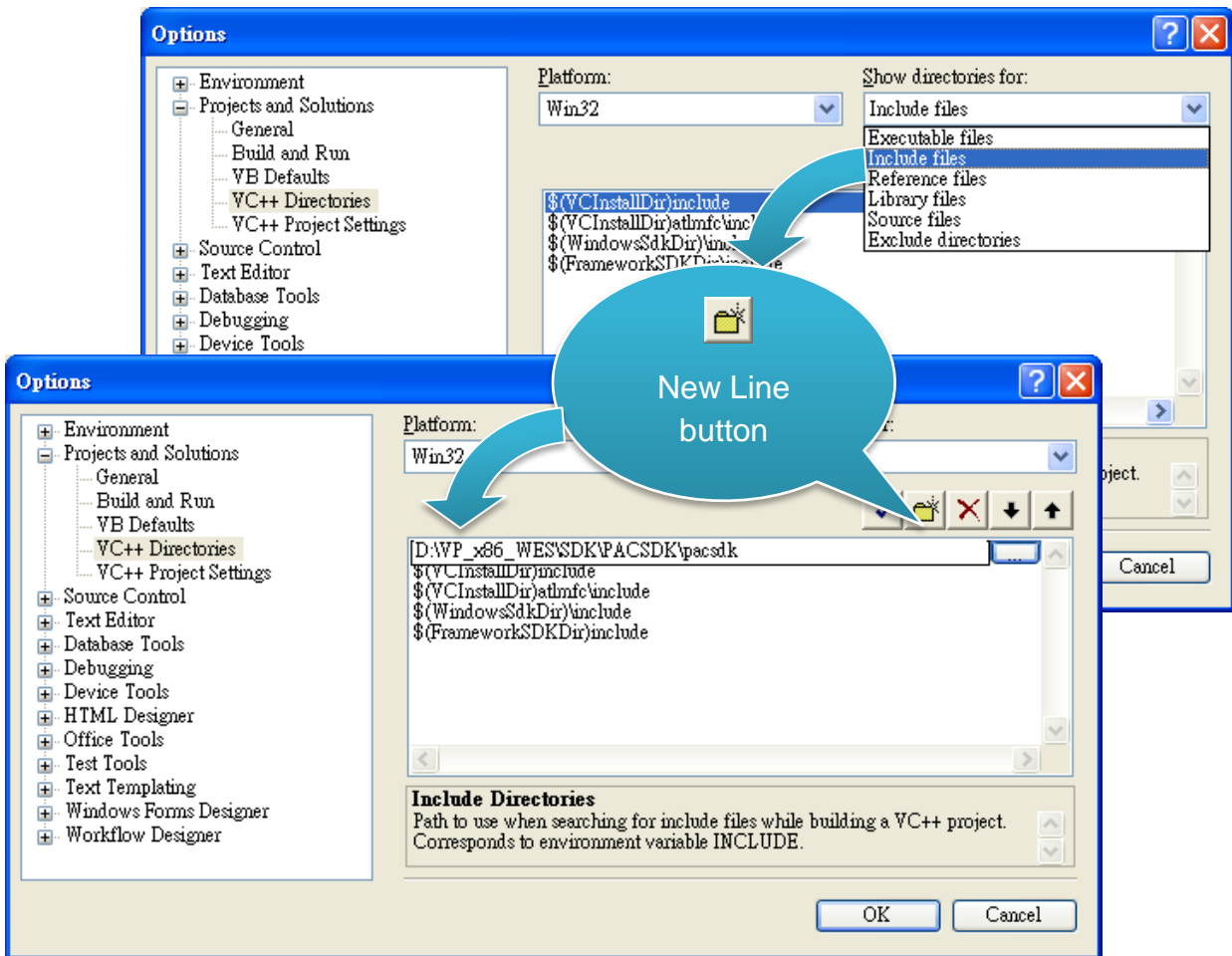
CD:\ippc-wes7\sdk\pacsdk\pacsdk\

<ftp://ftp.icpdas.com/pub/cd/ippc-wes7/sdk/pacsdk/pacsdk/>

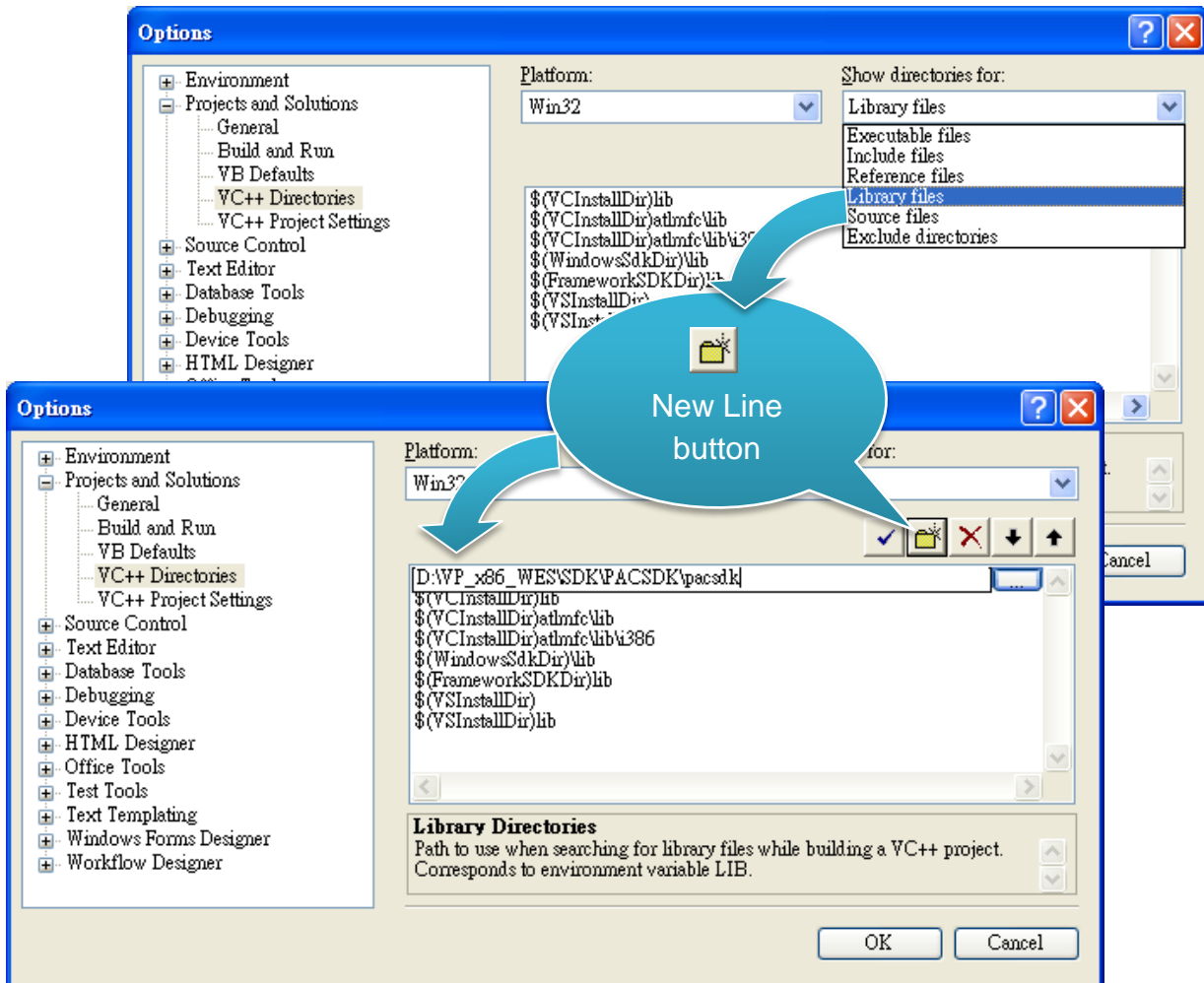
2. On the **Tools** menu, and then click **Options**



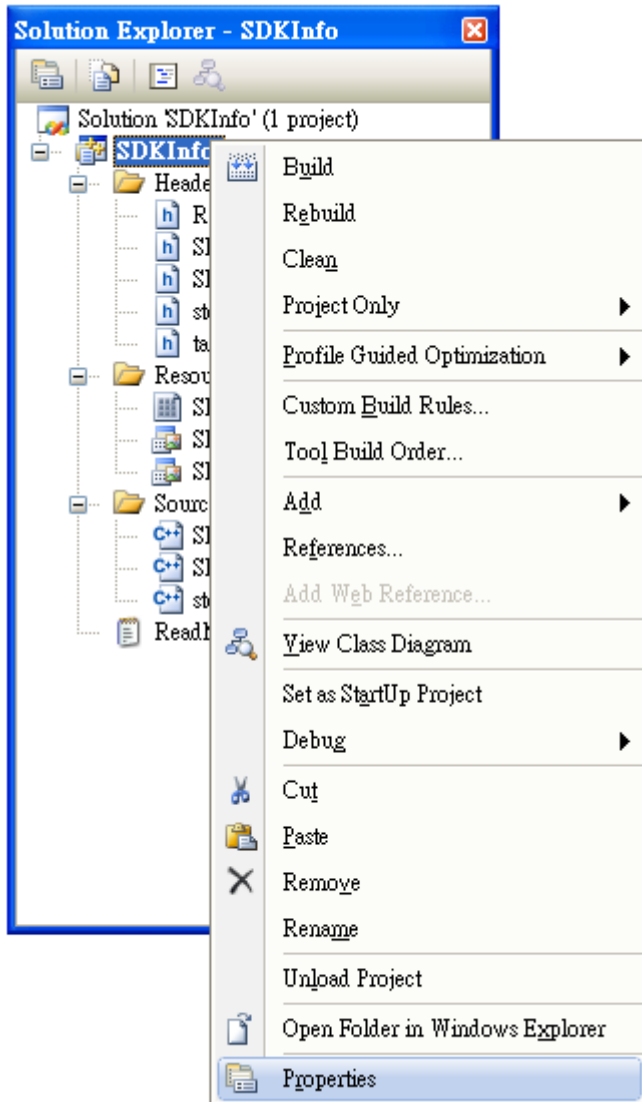
3. In the left pane, expand **Projects and Solutions**, and then click the **VC++ Directories**
4. Select **Include files** in the **Show directories for** drop down box, and then click the **New Line** button
5. Add a new line to the list of directories. Browse to the directory that contains the **PACSDK.H** file.



6. Select **Library files** in the **Show directories for** drop down box, and then click the **New Line** button
7. Add a new line to the list of directories. Browse to the directory that contains the **PACSDK.lib** file, and then click **OK** button

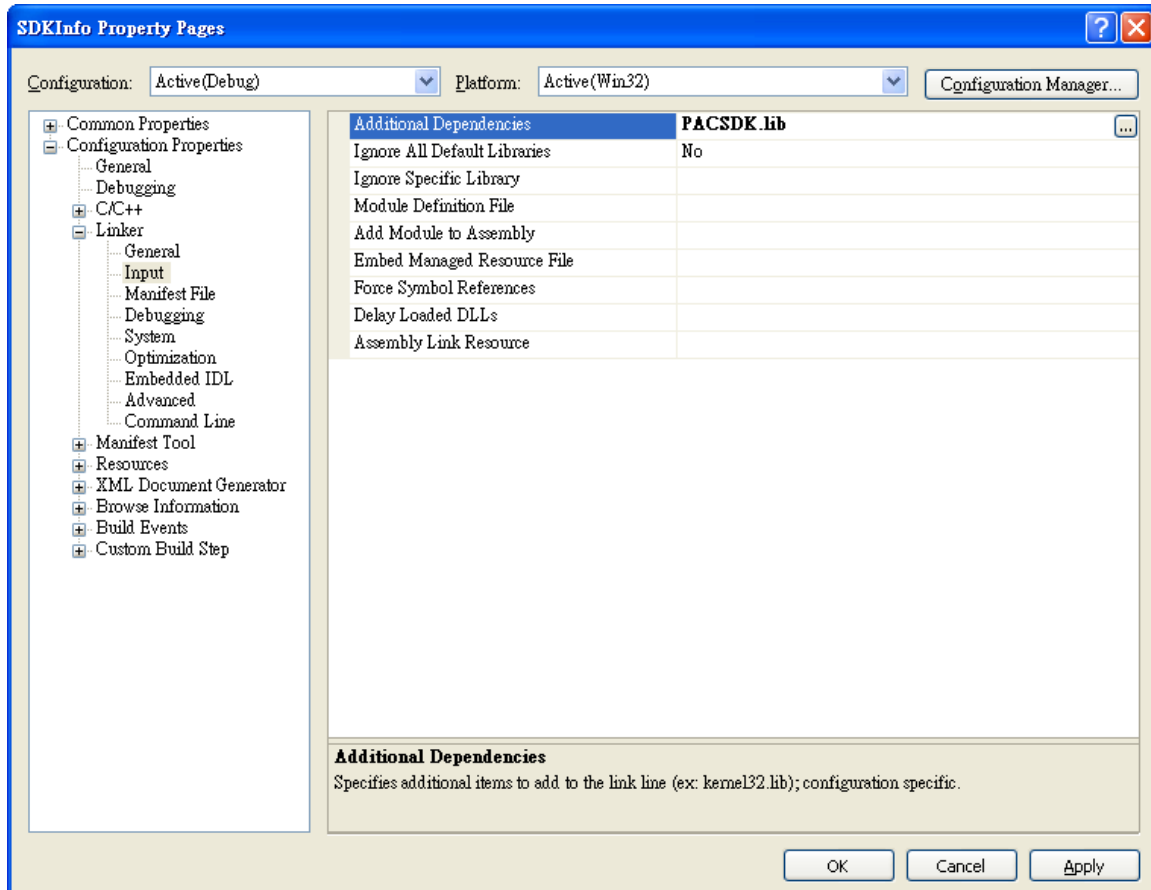


8. In the **Solution Explorer** windows, right-click the **project** name, and then click **Properties**



9. In the left pane, expand **Configuration Properties**, and then click the **Link**

10. In the right pane, type the **PACSDK.lib** in the **Additional Dependencies** item, click **Apply** button, and then click the **OK** button



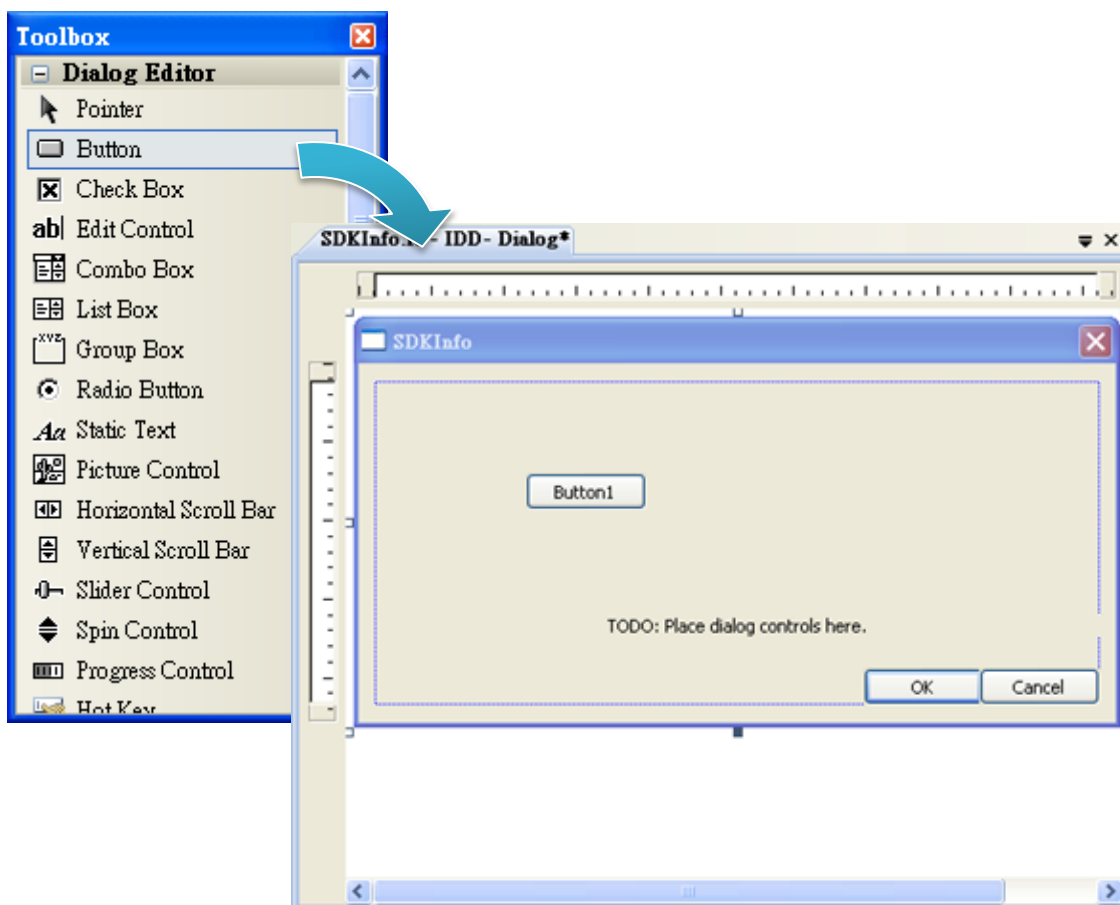
### 5.3.3. Add the Control to the Form

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.

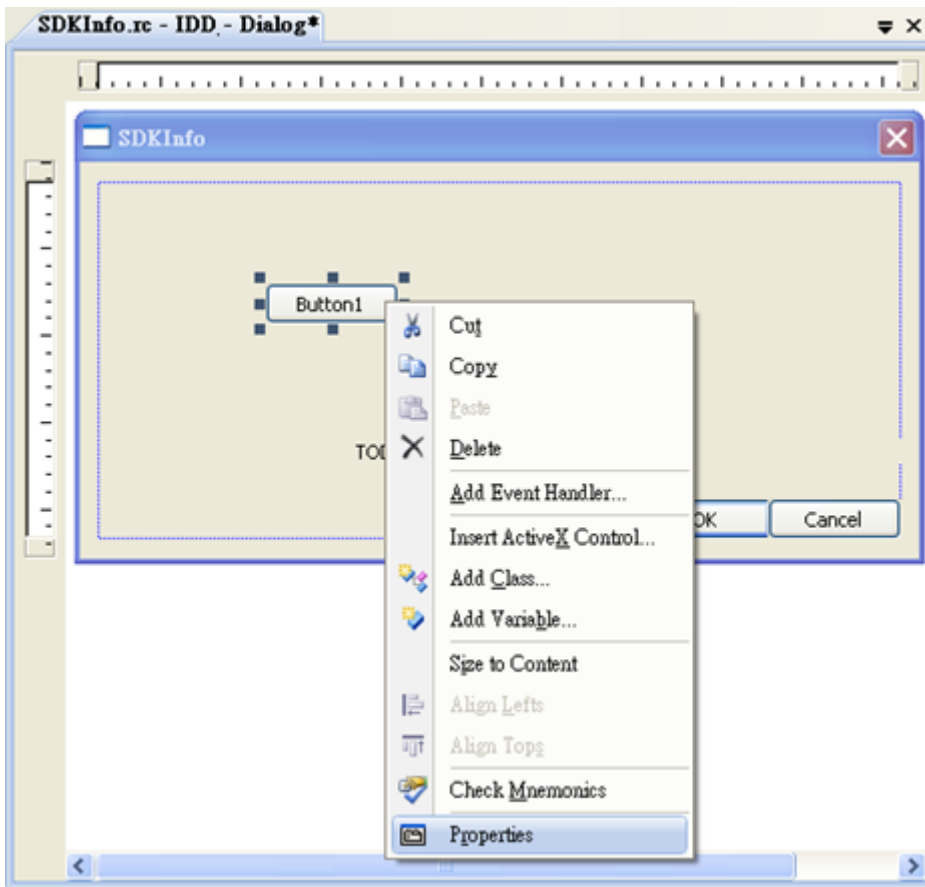
After you add a control to your form, you can use the Properties window to set its properties, such as background color and default text. The values that you specify in the Properties window are the initial values that will be assigned to that property when the control is created at run time.

#### To add the control to the form

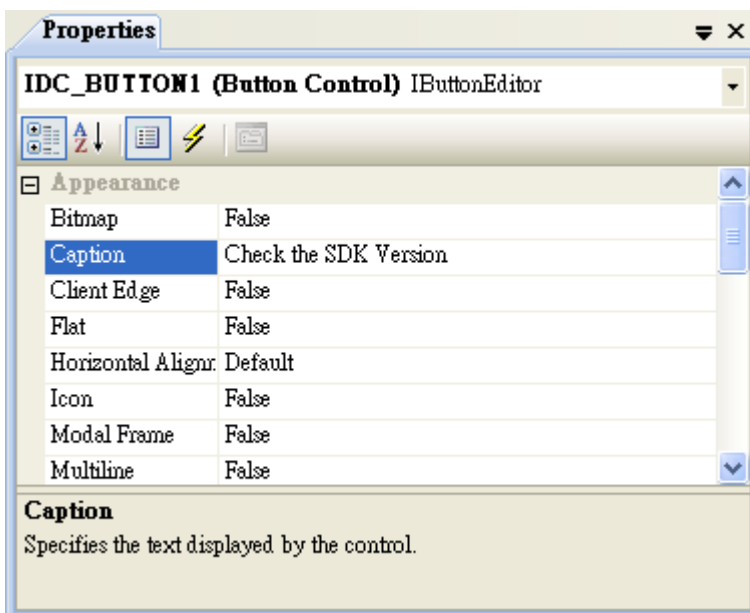
1. From the **Toolbox**, drag a **Button** control onto the form



2. Right-click the **Button** control, and then click **Properties**



3. In the **Properties** window, type **Check the SDK version** in the **Caption** item, and press **ENTER**



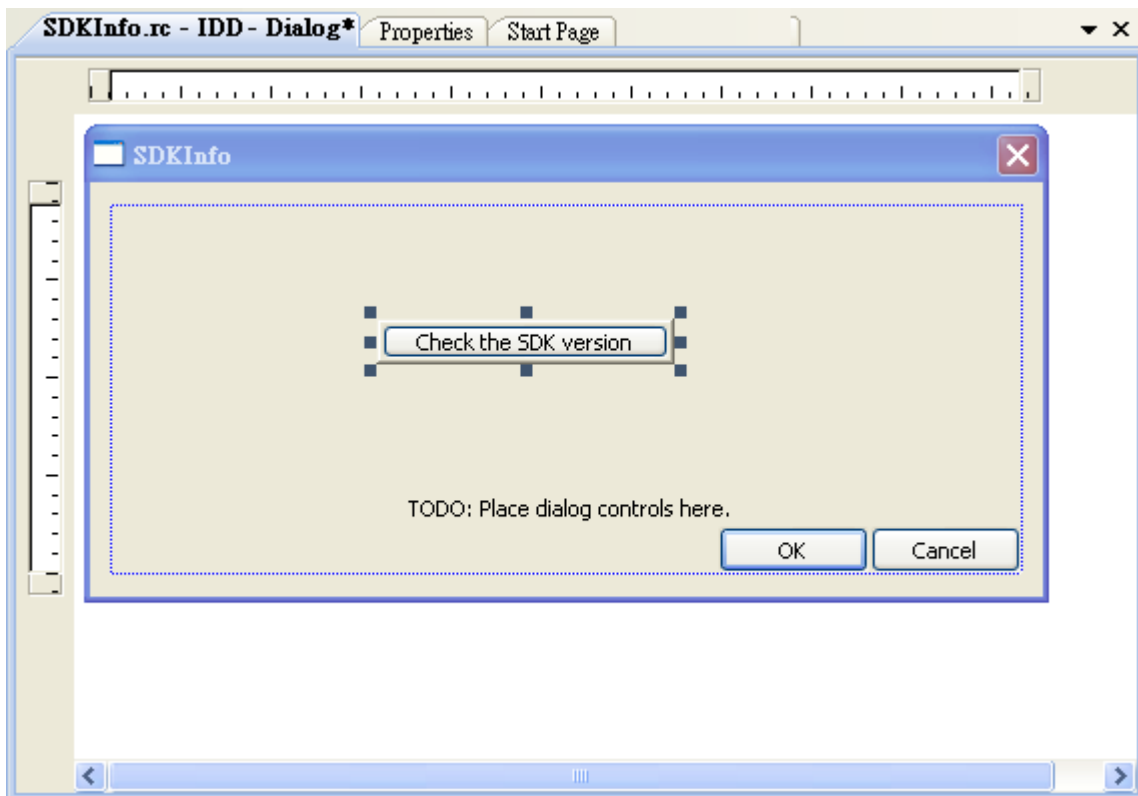


### 5.3.4. Add the Event Handling for the Control

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.

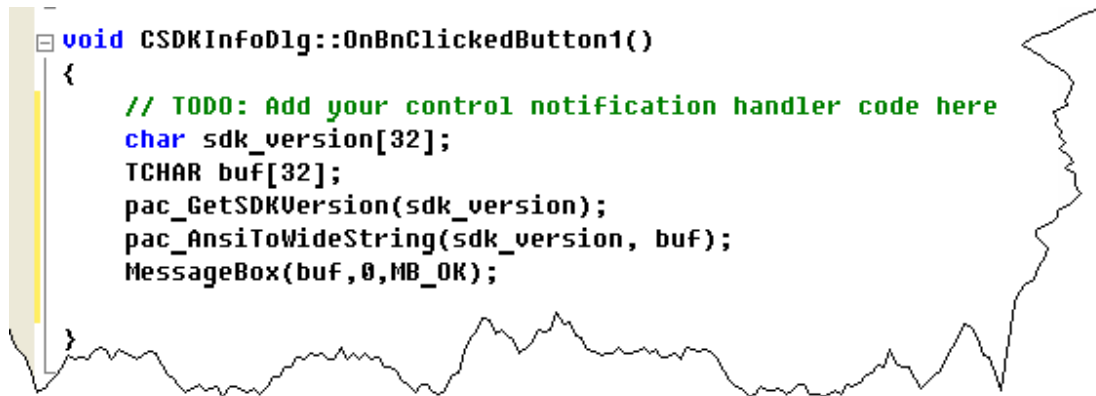
#### To add the event handling for the control

1. Double-click the button on the form



2. Inserting the following code

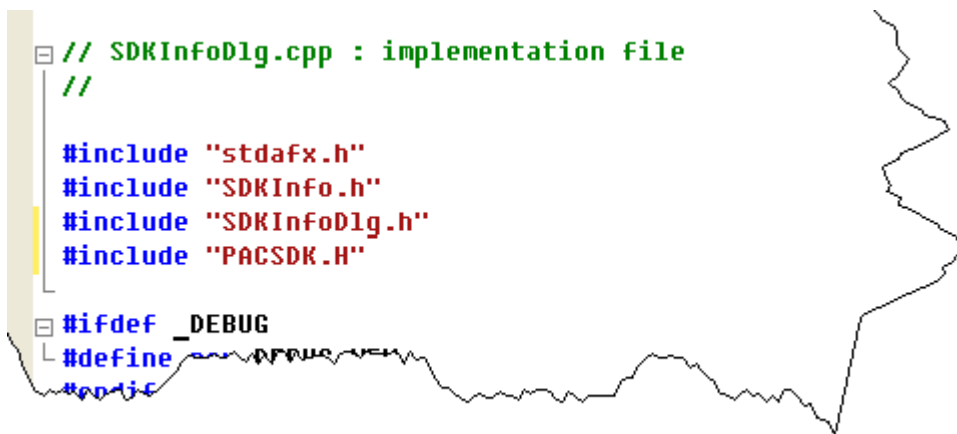
```
char sdk_version[32];
TCHAR buf[32];
pac_GetSDKVersion(sdk_version);
pac_AnsiToWideString(sdk_version, buf);
MessageBox(buf,0,MB_OK);
```



```
void CSDKInfoDlg::OnBnClickedButton1()
{
    // TODO: Add your control notification handler code here
    char sdk_version[32];
    TCHAR buf[32];
    pac_GetSDKVersion(sdk_version);
    pac_AnsiToWideString(sdk_version, buf);
    MessageBox(buf,0,MB_OK);
}
```

3. Inserting the following code into the header area

```
#include "PACSDK.H"
```



```
// SDKInfoDlg.cpp : implementation file
//
#include "stdafx.h"
#include "SDKInfo.h"
#include "SDKInfoDlg.h"
#include "PACSDK.H"

#ifdef _DEBUG
#define _AFX_NO_VALIDATION_SUPPORT
#endif
```

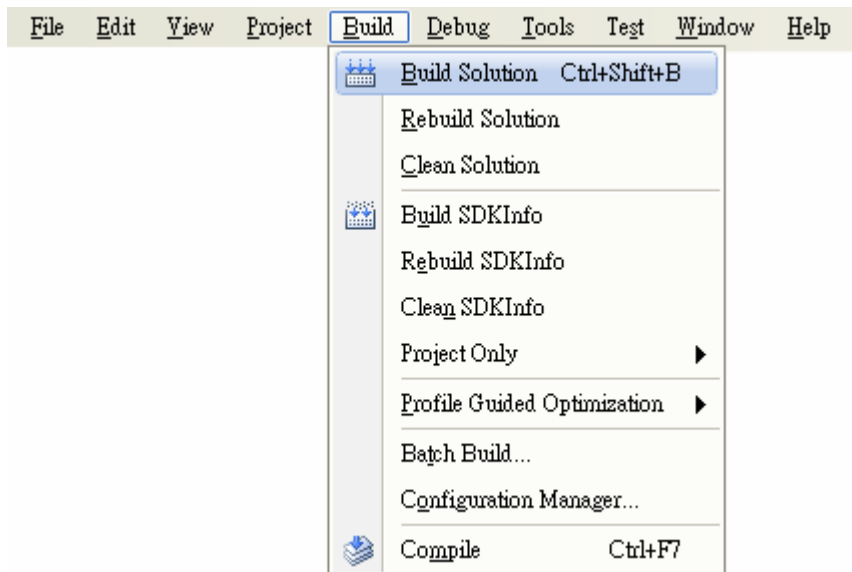
## 5.3.5. Upload the Application to iPPC

iPPC supports FTP server service. You can upload files to iPPC or download files from a public FTP server.



### To upload the application to iPPC

1. On the **Build** menu, click **Build Solution**



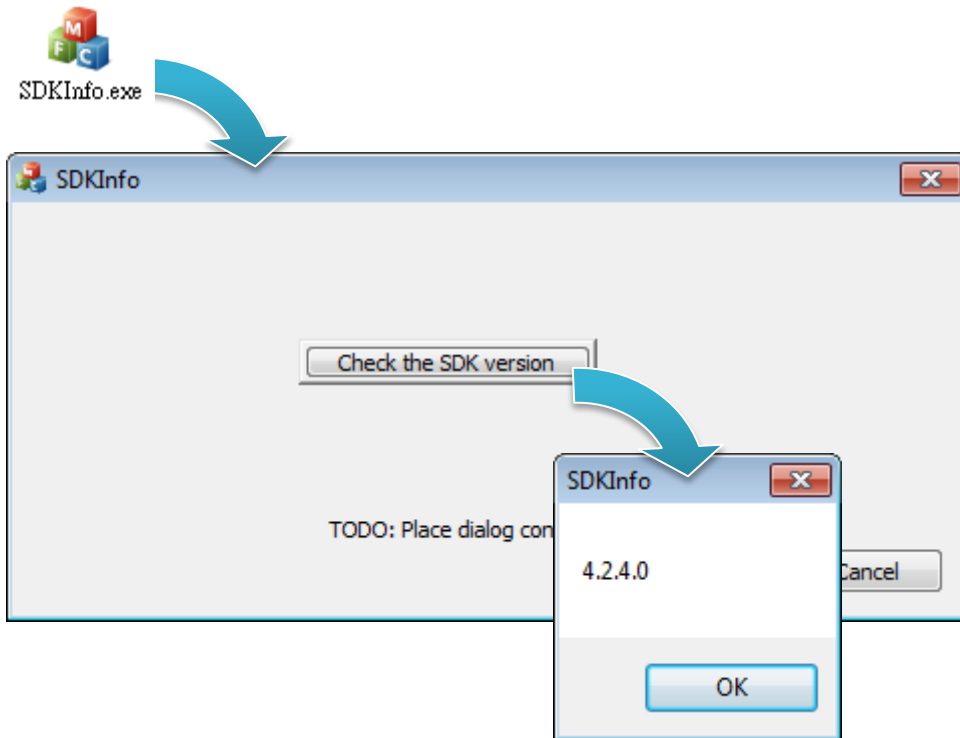
2. Open the **browser** and type the **IP address** of iPPC

3. Upload the SDKInfo.exe application to iPPC



### 5.3.6. Execute the Application on iPPC

After uploading the application to iPPC, you can just double-click it to execute it.

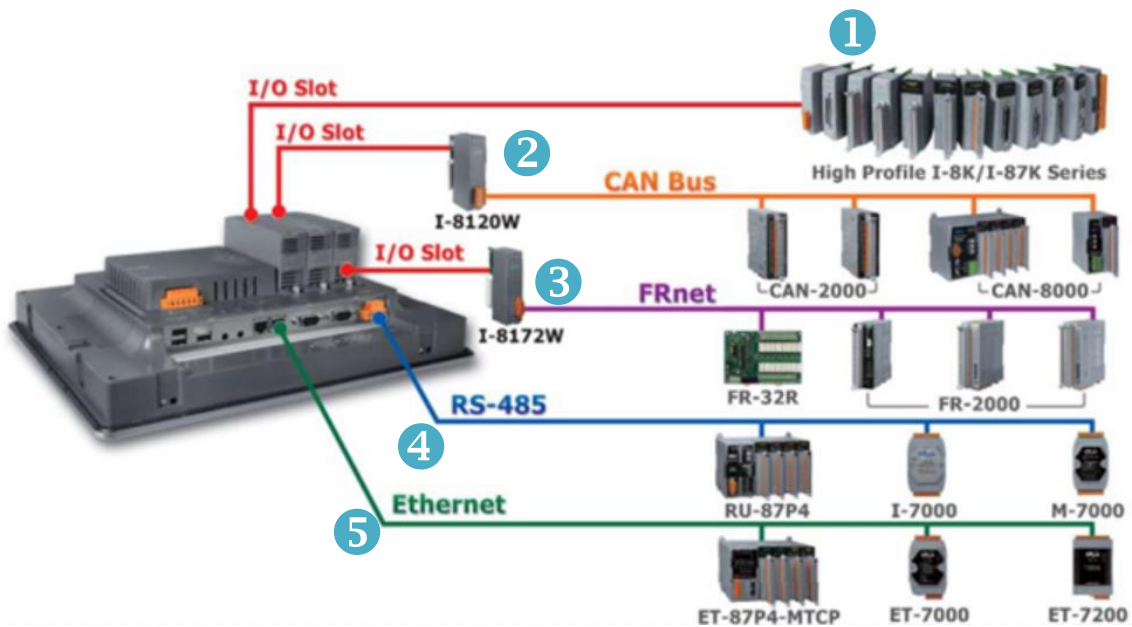


# 6. I/O Modules and SDK Selection

This chapter provides information on how to choose a suitable I/O expansion module and the corresponding SDK library to be used for developing programs on iPPC series modules.

## Choosing a SDK Library for an I/O Module

---



## 1 Local I/O Module (I-8K series and I-87K series)

There are two types of I/O modules that can be inserted into local bus of a iPPC device, Parallel and Serial. Parallel modules (I-8K Series) are high-speed modules and only support an MCU (Main Control Unit). Serial modules (I-87K Series) can support either an MCU or an I/O expansion unit. The following table shows the appropriate SDK library to be used for I/O modules.

### ➤ I-8K series I/O modules

Module	Native SDK	.NET CF SDK
I-8K series	PACSDK.dll	PACNET.dll
Specified I-8K series	Specified SDK	Specified SDK

For full details regarding I-8K series I/O modules and its demos, please refer to:  
CD:\ippc-wes7\demo\pacsdk\

### ➤ I-87K series I/O modules

Module	Native SDK	.NET CF SDK
I-87K series	PACSDK.dll	PACNET.dll
I-87K series with I-87088W	PACSDK_PWM.dll	PACNET.dll

For full details regarding I-87K series I/O modules and its demos, please refer to:  
CD:\ippc-wes7\demo\pacsdk\

➤ **Specified I-8K series modules**

Module	Native SDK	.NET CF SDK
I-8017HW	pac_i8017HW.dll	pac_i8017HWNet.dll
I-8014W	pac_i8014W.dll	pac_i8014WNet.dll
I-8024W	pac_i8024W.dll	pac8024WNet.dll
I-8026W	pac_i8026W.dll	pac_i8026WNet.dll
I-8048W	pac_i8048W.dll	pac_i8048WNet.dll
I-8050W	pac_i8050W.dll	pac_i8050WNet.dll
I-8084W	pac_i8084W.dll	pac_i8084WNet.dll
I-8088W	pac_i8088W.dll	pac_i8088WNet.dll
I-8093W	pac_i8093W.dll	pac8093WNet.dll
Others	PACSDK.dll	PACNET.dll

**2 CAN bus**

The Controller Area Network (CAN) is a serial communication way, which efficiently supports distributed real-time control with a very high level of security. It provides the error-processing mechanisms and concepts of message priority. These features can improve the network reliability and transmission efficiency.

Module	Native SDK	.NET CF SDK
I-8120W	I8120.dll	I8120net_pac.dll

For full details regarding CAN bus series modules and its demos, please refer to:  
 CD:\ippc-wes7\demo\pacsdk\

### 3 FRnet

FRnet is an innovative industrial field bus technology that uses twisted pair cable as the transmission medium. The status of all I/O devices is updated on a fixed cycle, no matter how many FRnet I/O modules are connected to the FRnet network.

Module	Native SDK	.NET CF SDK
I-8172W	pac_i8172W.dll	pac8172WNet.dll

### 4 RS-485 (I-7K series and M-7K series)

I-7000, M-7000, RU-87Pn and high profile I-87K series modules connect to iPPC series devices via a twisted-pair, multi-drop, 2-wire RS-485 network.

#### ➤ I-7K series I/O modules

Module	Native SDK	.NET CF SDK
I-7K series	PACSDK.dll	PACNET.dll
I-7K series with I-7088(D)	PACSDK_PWM.dll	PACNET.dll

For full details regarding I-7K series I/O modules and its demos, please refer to:

CD:\ippc-wes7\demo\pacsdk\

#### ➤ M-7K series I/O modules

Module	Native SDK	.NET CF SDK
M-7K series	Modbus Demo	Modbus Demo

For more detailed information about M-7K series modules using Modbus protocol and its demos, please refer to:

CD:\ippc-wes7\demo\pacsdk\



➤ **RU-87Pn + I-87K series I/O modules**

Module	Native SDK	.NET CF SDK
RU-87Pn + I-87K	PACSDK.dll	PACNET.dll

➤ **Other specified I/O**

Module	Native SDK	.NET CF SDK
Others	PACSDK.dll	PACNET.dll

**5 Ethernet (ET-7K series and I-8KE4/8-MTCP)**

The Ethernet I/O devices available include ET-7000 and I-8KE4/8-MTCP, and support either the DCON or the Modbus/TCP communication protocol.

Module	Native SDK	.NET CF SDK
ET-7K	Modbus Demo	Modbus Demo

For more detailed information about ET-7K and I-8KE4/8-MTCP series modules using Modbus protocol and its demos, please refer to:

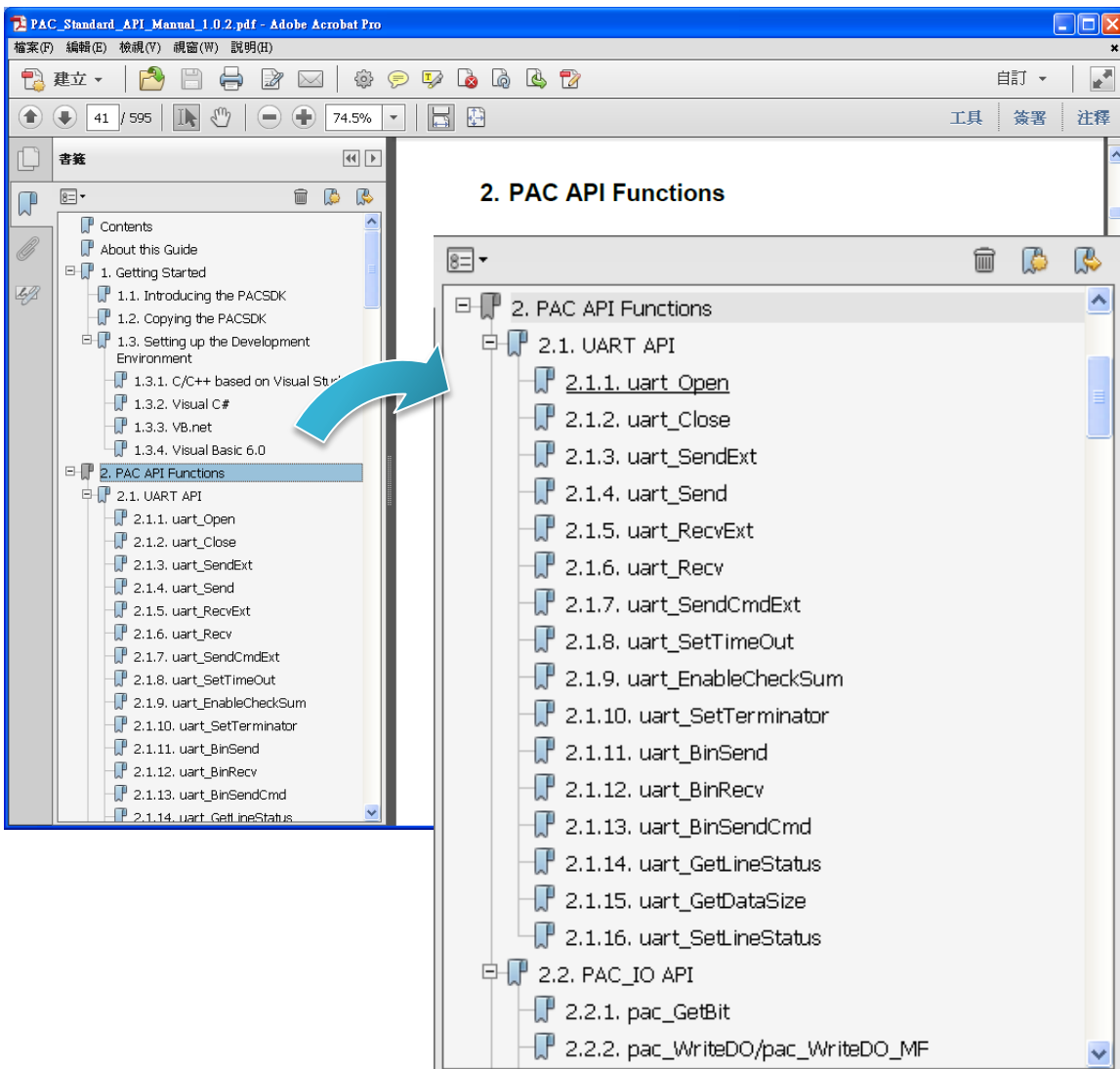
CD:\ippc-wes7\demo\pacsdk\

# 7. APIs and Samples

This chapter provides a brief overview of PAC APIs and samples that have been designed for iPPC.

ICP DAS provides a set of samples in different programming languages. You can examine the sample codes, which includes numerous comments, to familiarize yourself with the PAC APIs. This will allow developing your own applications quickly by modifying these demo programs.

For full usage information regarding the description, prototype and the arguments of the functions, please refer to the “PAC Standard API Manual”



## 7.1. PAC Standard APIs

The diagram below shows the set of each system operation API provided in the PACSDK.



## 7.1.1. VB.NET Samples for PAC Standard APIs

The PAC SDK includes the following samples that demonstrate the use of the PAC Standard APIs in a VB.NET language environment. The following samples can be found on the CD that was provided with the package or by downloading the latest version from ICP DAS web site.

For VB.NET applications, these demo programs can be obtained from:

CD:\ippc-wes7\demo\pacsdk\vb\standard\

<ftp://ftp.icpdas.com/pub/cd/ippc-wes7/demo/pacsdk/vb/standard/>

Folder	Demo	Explanation
deviceinformation	deviceinformation	Retrieves information about the OS version, the CPU version and the SDK version, etc.
diagnostic	diagnostic	Retrieves information about the slot count and the module inserted in the backplane.
dip	dip	Retrieves information about the status of the DIP switch.
getrotaryid	getrotaryid	Retrieves information about the status of the rotary switch.
memory	memory	Shows how to read/write data values from/to EEPROM.
uart	uart	Shows how to read the name of local I/O modules via UART
watchdog	watchdog	Displays information about how to operate the watchdog

## 7.1.2. C# Samples for PAC Standard APIs

The PAC SDK includes the following samples that demonstrate the use of the PAC Standard APIs in a C# language environment. The following samples can be found on the CD that was provided with the package or by downloading the latest version from ICP DAS web site.

For C# applications, these demo programs can be obtained from:

CD:\ippc-wes7\demo\pacsdk\csharp.net\standard\windows\_forms\

[ftp://ftp.icpdas.com/pub/cd/ippc-wes7/demo/pacsdk/csharp.net/standard/windows\\_forms/](ftp://ftp.icpdas.com/pub/cd/ippc-wes7/demo/pacsdk/csharp.net/standard/windows_forms/)

Folder	Demo	Explanation
deviceinformation	deviceinformation	Retrieves information about the OS version, the CPU version and the SDK version, etc.
diagnostic	diagnostic	Retrieves information about the slot count and the module inserted in the backplane.
dip	dip	Retrieves information about the status of the DIP switch.
getrotaryid	getrotaryid	Retrieves information about the status of the rotary switch.
memory	memory	Shows how to read/write date values from/to EEPROM.
uart	uart	Shows how to read the name of local I/O modules via UART
watchdog	watchdog	Displays information about how to operate the watchdog

### 7.1.3. Visual C++ Samples for PAC Standard APIs

The PAC SDK includes the following samples that demonstrate the use of the PAC Standard APIs in a Visual C++ language environment. The following samples can be found on the CD that was provided with the package or by downloading the latest version from ICP DAS web site.

For Visual C++ applications, these demo programs can be obtained from:

CD:\ippc-wes7\demo\pacsdk\vc\standard\

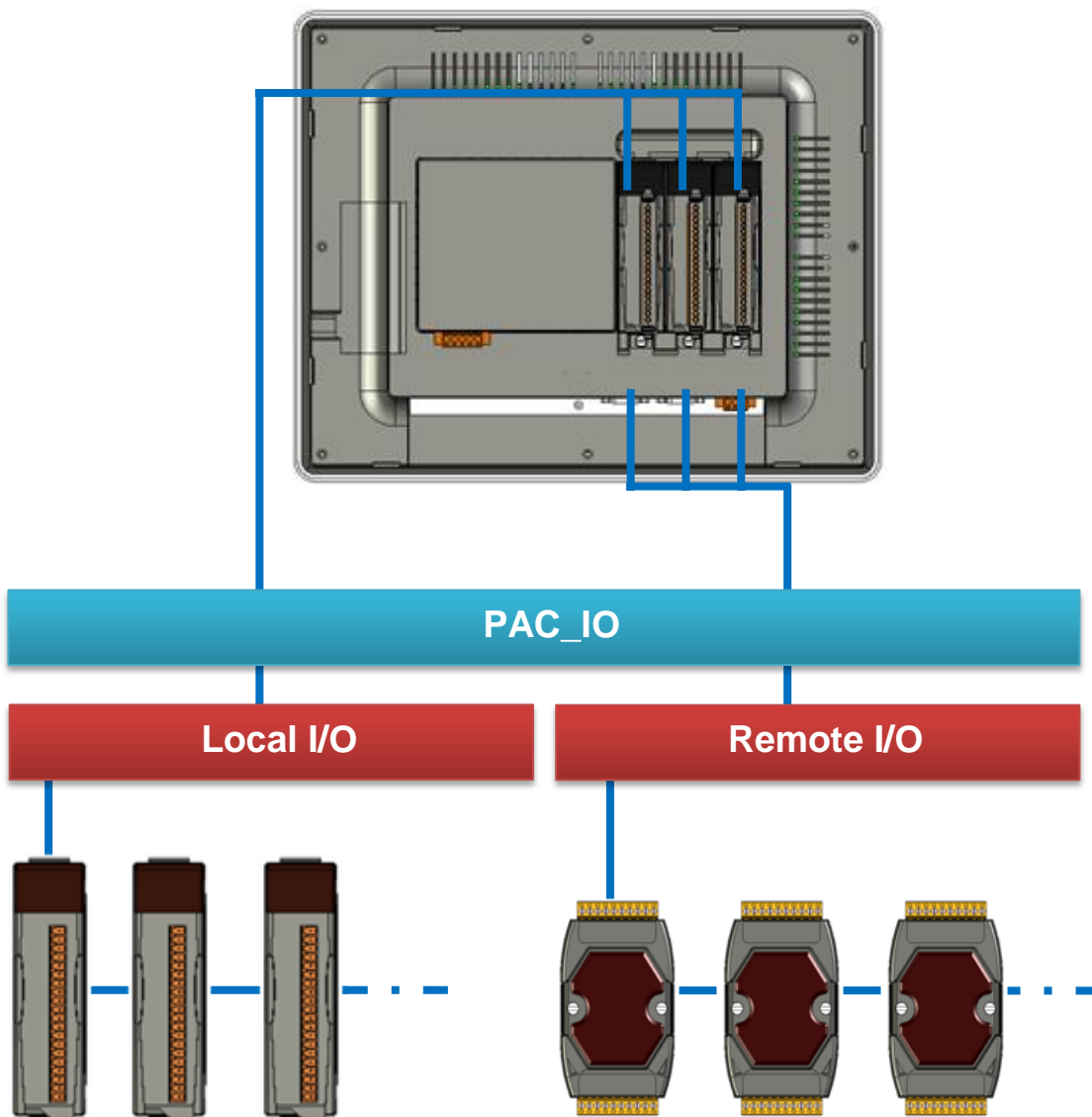
<ftp://ftp.icpdas.com/pub/cd/ippc-wes7/demo/pacsdk/vc/standard/>

Folder	Demo	Explanation
diagnostic	diagnostic	Retrieves information about the slot count and the module inserted in the backplane.
dip	dip	Retrieves information about the status of the DIP switch.
getdeviceinformation	getdeviceinformation	Retrieves information about the OS version, the CPU version and the SDK version, etc.
GetRotaryID	GetRotaryID	Retrieves information about the status of the rotary switch.
Memory	readmemory	Shows how to read date values from EEPROM.
	writememory	Shows how to write date values to EEPROM.
uart_sendcmd	uart_sendcmd	Shows how to read the name of local I/O modules via UART
WatchDog	WatchDog	Displays information about how to operate the watchdog

## 7.2. PAC IO APIs

The diagram below shows the types of the PAC IO APIs provided in the PACSDK or the specified SDK.

For more information about the APIs and samples provided by the expansion I/O modules, please refer to chapter 6. I/O Modules and SDK Selection



## 7.2.1. VB.NET Samples for PAC IO APIs

The PAC SDK includes the following samples that demonstrate the use of the PAC IO APIs in a VB.NET language environment. The following samples can be found on the CD that was provided with the package or by downloading the latest version from ICP DAS web site.

For VB.NET applications, these demo programs can be obtained from:

CD:\ippc-wes7\demo\pacsdk\vb\io\

<ftp://ftp.icpdas.com/pub/cd/ippc-wes7/demo/pacsdk/vb/io/>

Folder	Demo	Explanation
Local	find_io	Shows how to retrieve the module names and types which plugged in the iPPC.
	8k_di	Shows how to read the DI values of DI module. This demo program is used by 8K series DI modules.
	8k_do	Shows how to write the DO values to DO module. This demo program is used by 8K series DO modules.
	8k_dio	Shows how to read the DI and the DO values of the DIO module. This demo program is used by 8K series DIO modules.
	87k_basic	Shows how to send/receive a command/response application. This demo program is used by 87K series modules.
	87K_demo	Shows how use UART API and the IO modules located as slots. This demo program is used by 87K series modules.
	87k_ai	Shows how to read the AI values of AI module. This demo program is used by 87K series AI modules.
	87k_ao	Shows how to write the AO values to AO module. This demo program is used by 87K series AO modules.
	87k_di	Shows how to read the DI values of DI module. This demo program is used by 87K series DI modules.
	87k_do	Shows how to write the DO values to DO module. This demo program is used by 87K series DO modules.
	87k_dio	Shows how to read the DI and the DO values of the DIO



Folder	Demo	Explanation
		module. This demo program is used by 87K series DIO modules.
Remote	7k87k_basic	Shows how to send/receive a command/response application. This demo program is used by 7K or 87K series AI modules which connected through a COM port.
	7k87k_ai	Shows how to read the AI values of AI module. This demo program is used by 7K or 87K series AI modules which connected through a COM port.
	7k87k_ao	Shows how to write the AO values to AO module. This demo program is used by 7K or 87K series AI modules which connected through a COM port.
	7k87k_di	Shows how to read the DI values of DI module. This demo program is used by 7K or 87K series AI modules which connected through a COM port.
	7k87k_do	Shows how to write the DO values to DO module. This demo program is used by 7K or 87K series AI modules which connected through a COM port.
	7k87k_dio	Shows how to read the DI and the DO values of the DIO module. This demo program is used by 7K or 87K series AI modules which connected through a COM port.

## 7.2.2. C# Samples for PAC IO APIs

The PAC SDK includes the following samples that demonstrate the use of the PAC IO APIs in a C# language environment. The following samples can be found on the CD that was provided with the package or by downloading the latest version from ICP DAS web site.

For C# applications, these demo programs can be obtained from:

CD:\ippc-wes7\demo\pacsdk\csharp.net\io\

<ftp://ftp.icpdas.com/pub/cd/ippc-wes7/demo/pacsdk/csharp.net/io/>

Folder	Demo	Explanation
Local	find_io	Shows how to retrieve the module names and types which plugged in the iPPC.
	8k_di	Shows how to read the DI values of DI module. This demo program is used by 8K series DI modules.
	8k_do	Shows how to write the DO values to DO module. This demo program is used by 8K series DO modules.
	8k_dio	Shows how to read the DI and the DO values of the DIO module. This demo program is used by 8K series DIO modules.
	87k_basic	Shows how to send/receive a command/response application. This demo program is used by 87K series modules.
	87K_demo	Shows how use UART API and the IO modules located as slots. This demo program is used by 87K series modules.
	87k_ai	Shows how to read the AI values of AI module. This demo program is used by 87K series AI modules.
	87k_ao	Shows how to write the AO values to AO module. This demo program is used by 87K series AO modules.
	87k_di	Shows how to read the DI values of DI module. This demo program is used by 87K series DI modules.
	87k_do	Shows how to write the DO values to DO module. This demo program is used by 87K series DO modules.
	87k_dio	Shows how to read the DI and the DO values of the DIO

Folder	Demo	Explanation
		module. This demo program is used by 87K series DIO modules.
Remote	7k87k_basic	Shows how to send/receive a command/response application. This demo program is used by 7K or 87K series AI modules which connected through a COM port.
	7k87k_ai	Shows how to read the AI values of AI module. This demo program is used by 7K or 87K series AI modules which connected through a COM port.
	7k87k_ao	Shows how to write the AO values to AO module. This demo program is used by 7K or 87K series AI modules which connected through a COM port.
	7k87k_di	Shows how to read the DI values of DI module. This demo program is used by 7K or 87K series AI modules which connected through a COM port.
	7k87k_do	Shows how to write the DO values to DO module. This demo program is used by 7K or 87K series AI modules which connected through a COM port.
	7k87k_dio	Shows how to read the DI and the DO values of the DIO module. This demo program is used by 7K or 87K series AI modules which connected through a COM port.

### 7.2.3. Visual C++ Samples for PAC IO APIs

The PAC SDK includes the following samples that demonstrate the use of the PAC IO APIs in a Visual C++ language environment. The following samples can be found on the CD that was provided with the package or by downloading the latest version from ICP DAS web site.

For Visual C++ applications, these demo programs can be obtained from:

CD:\ippc-wes7\demo\pacsdk\vc\io\

<ftp://ftp.icpdas.com/pub/cd/ippc-wes7/demo/pacsdk/vc/io/>

Folder	Demo	Explanation
Local	find_io	Shows how to retrieve the module names and types which plugged in the iPPC.
	8k_di	Shows how to read the DI values of DI module. This demo program is used by 8K series DI modules.
	8k_do	Shows how to write the DO values to DO module. This demo program is used by 8K series DO modules.
	8k_dio	Shows how to read the DI and the DO values of the DIO module. This demo program is used by 8K series DIO modules.
	87k_basic	Shows how to send/receive a command/response application. This demo program is used by 87K series modules.
	87K_demo	Shows how use UART API and the IO modules located as slots. This demo program is used by 87K series modules.
	87k_ai	Shows how to read the AI values of AI module. This demo program is used by 87K series AI modules.
	87k_ao	Shows how to write the AO values to AO module. This demo program is used by 87K series AO modules.
	87k_di	Shows how to read the DI values of DI module. This demo program is used by 87K series DI modules.
	87k_do	Shows how to write the DO values to DO module. This demo program is used by 87K series DO modules.

Folder	Demo	Explanation
	87k_dio	Shows how to read the DI and the DO values of the DIO module. This demo program is used by 87K series DIO modules.
Remote	7k87k_basic	Shows how to send/receive a command/response application. This demo program is used by 7K or 87K series AI modules which connected through a COM port.
	7k87k_ai	Shows how to read the AI values of AI module. This demo program is used by 7K or 87K series AI modules which connected through a COM port.
	7k87k_ao	Shows how to write the AO values to AO module. This demo program is used by 7K or 87K series AI modules which connected through a COM port.
	7k87k_di	Shows how to read the DI values of DI module. This demo program is used by 7K or 87K series AI modules which connected through a COM port.
	7k87k_do	Shows how to write the DO values to DO module. This demo program is used by 7K or 87K series AI modules which connected through a COM port.
	7k87k_dio	Shows how to read the DI and the DO values of the DIO module. This demo program is used by 7K or 87K series AI modules which connected through a COM port.

## 8. Restore and Recovery

This chapter provides information of the iPPC restore and recovery, and a guided tour that describes the steps needed to restore and recovery the iPPC.

The iPPC come with a rescue CF card that can be used to not only boot the iPPC when the OS fails to load, but also recover files.

The recovery file of the rescue CF card can be found separately on the CD that was provided with the package or by downloading the latest version from ICP DAS web site.

CD:\ippc-wes7\rescue\_disk\

[ftp://ftp.icpdas.com/pub/cd/ippc-wes7/rescue\\_disk/](ftp://ftp.icpdas.com/pub/cd/ippc-wes7/rescue_disk/)

## 8.1. Recovering the iPPC

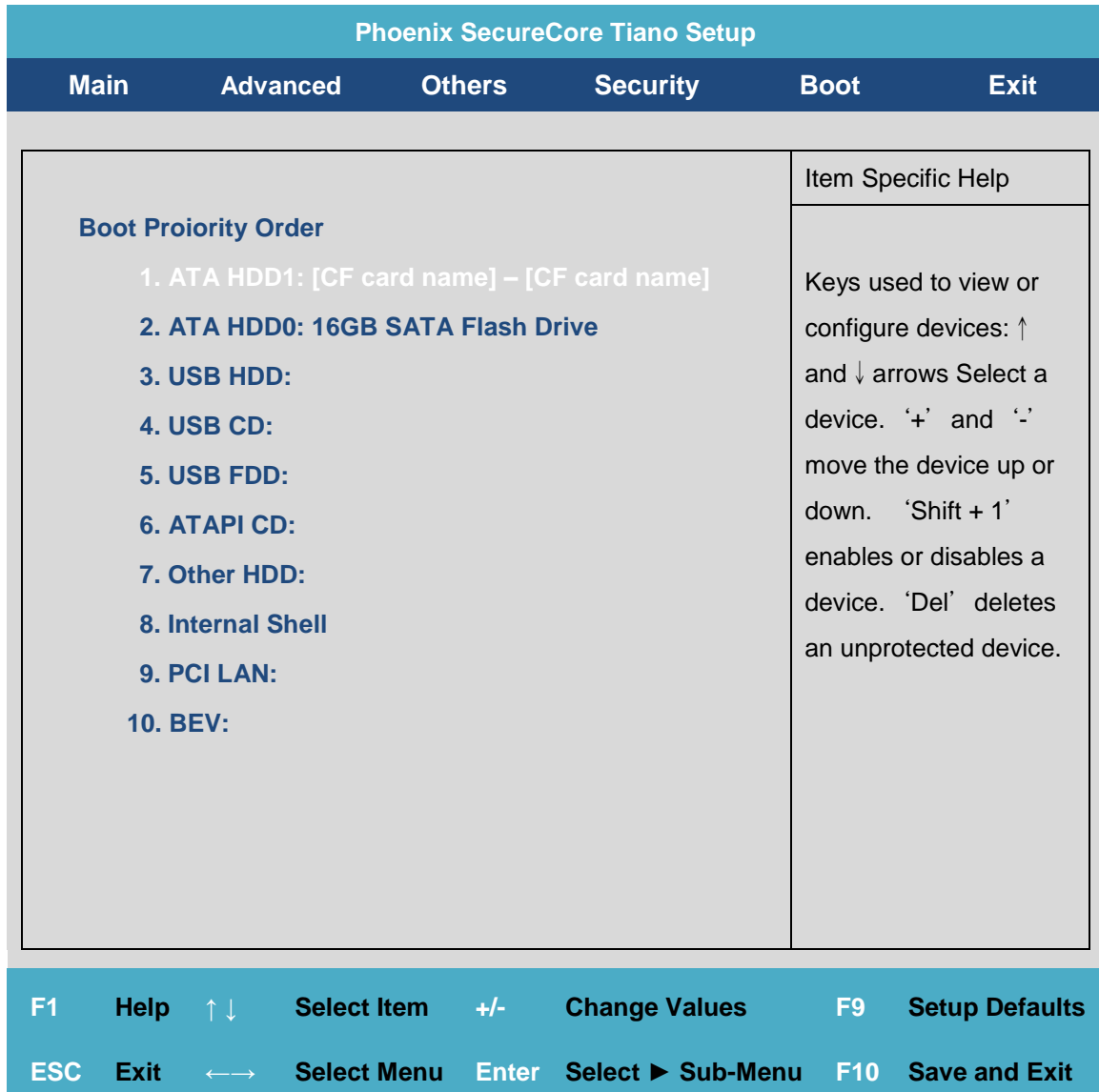
The iPPC comes with a rescue CF card that can be used to restore the iPPC to factory default settings by reinstalling the iPPC OS image.

If the iPPC crashes and won't start up, you can use the rescue CF card to start up the iPPC and then fix the problem that caused the crash.

### To restore the iPPC OS

1. Plug the Rescue CF card into CF slot
2. Restart the iPPC, and then enter the BIOS by pressing **F2** key

- Press the → key to highlight the **Boot** tab, and then press ↓ key to select **ATA HDD1: [CF card name] – [CF card name]**
- Press + key to move the **ATA HDD1: [CF card name] – [CF card name]** to the top

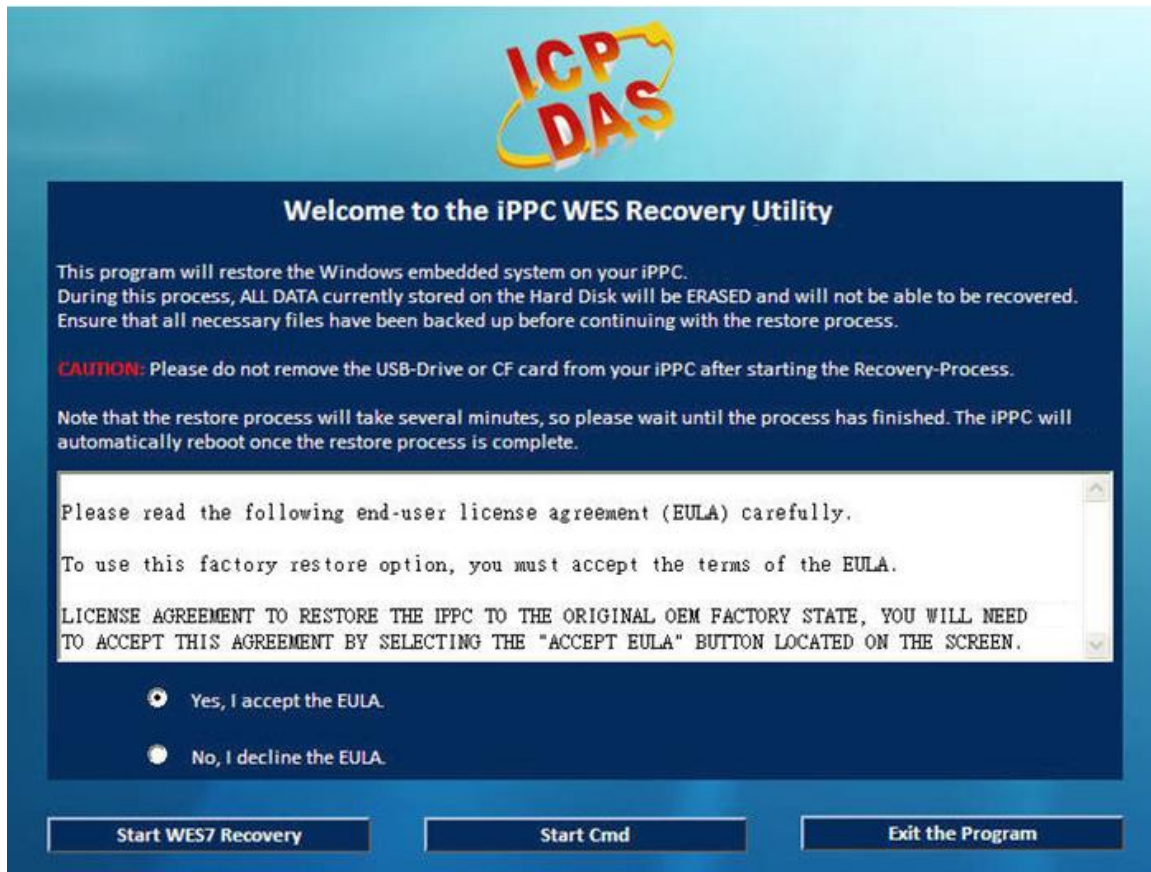


- Press the **F10** key, and then select **Yes** to save settings and exit the utility.  
The iPPC will restart and then enter to the iPPC WES Recovery Utility.



6. Select the **Yes, I accept the EULA** check box, and then click the **Start WES7 Recovery** button for start the recovery process

The process will take a few minutes until this utility is displayed again.



7. Click the **Exit the Program** button, and then repeat the step 2 to 5 to use the **ATAHDD0: [Flash (SATA) name]** as a boot drive and restart the iPPC

## 8.2. Restoring the Rescue CF Card

The rescue CF card is rescue equipment that allows you to perform some maintenance tasks on your system in case of failure.

Once the rescue CF card are partitioned or formatted, you must restore the rescue CF card.

### Requirements

For restoring the Rescue CF card, you should prepare Ghost 11 or later, which you could obtain by contacting Symantec (<http://www.symantec.com>)

In this article, we will use Symantec Norton Ghost32 V.11 (The Symantec Norton Ghost V.11 or above version are recommend) to restore the rescue CF card.

### To restore the rescue CF card

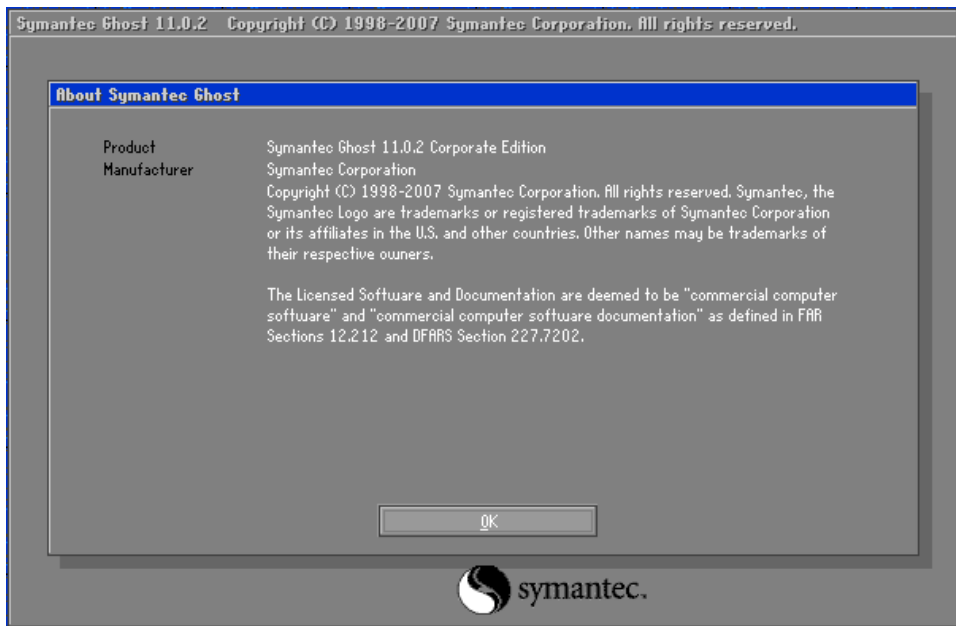
1. Get the rescue ghost file, rescue.gho

The rescue.gho file can be found on the CD that was provided with the package or by downloading the latest version from ICP DAS web site.

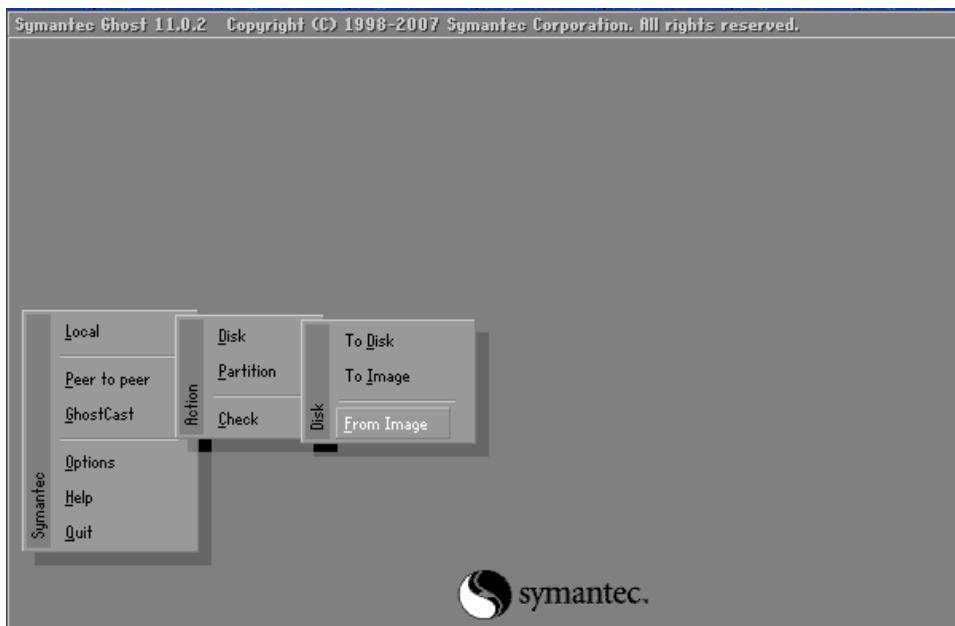
CD:\ippc-wes7\rescue\_disk\

[ftp://ftp.icpdas.com/pub/cd/ippc-wes7/rescue\\_disk/](ftp://ftp.icpdas.com/pub/cd/ippc-wes7/rescue_disk/)

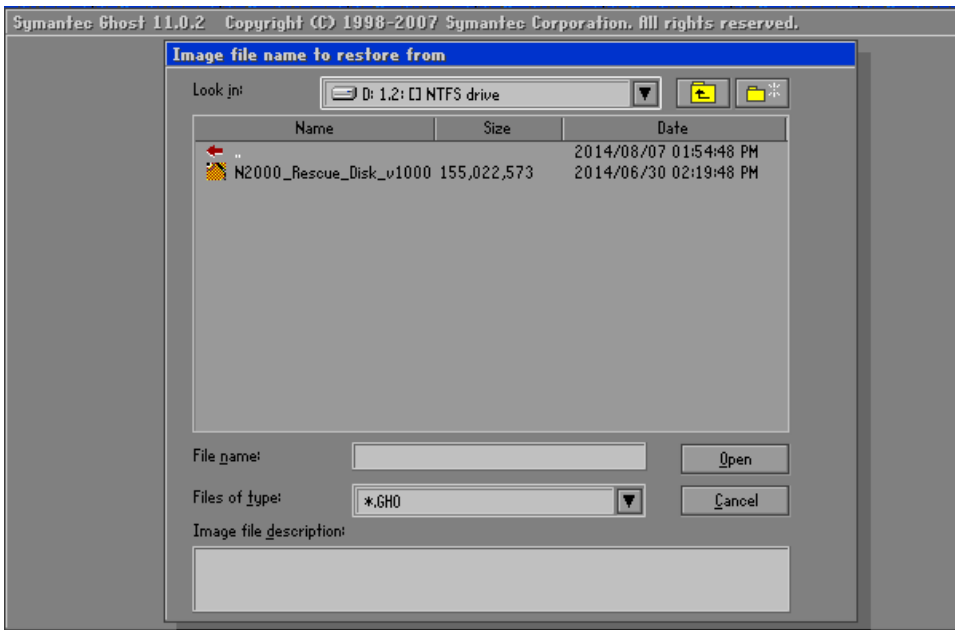
2. Run the **Symantec Ghost32**, and then click **OK** button



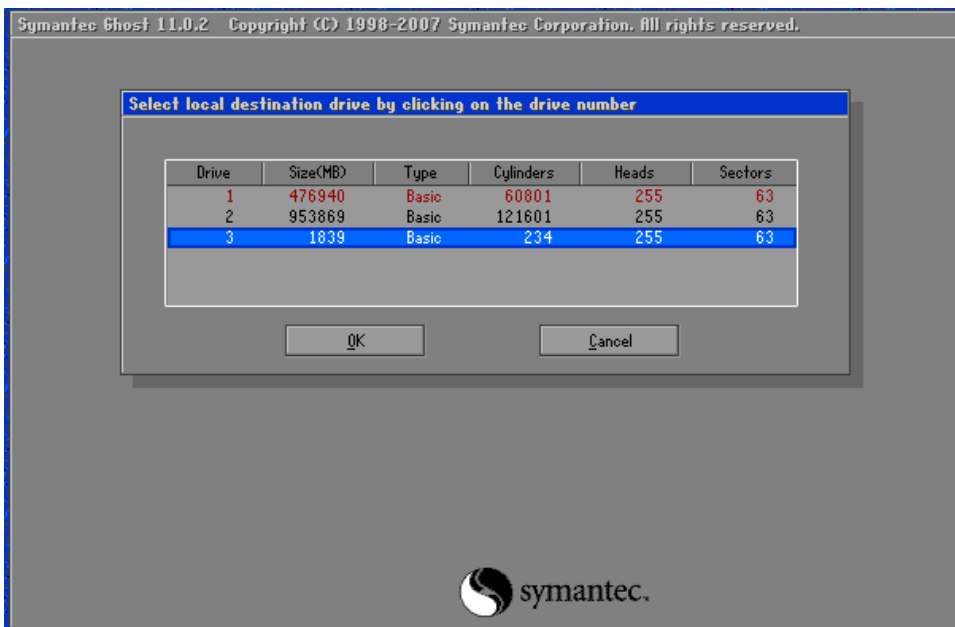
3. Click Function Menu, point to **Local**, point to **Disk**, and then click **From Image**



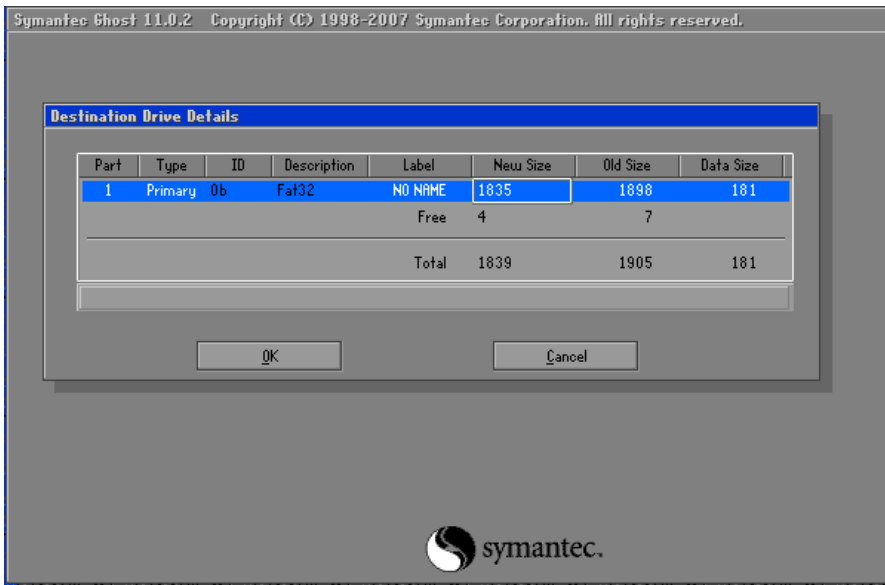
4. Select the rescue ghost file, **rescue.gho**, that you saved and then click **Open**



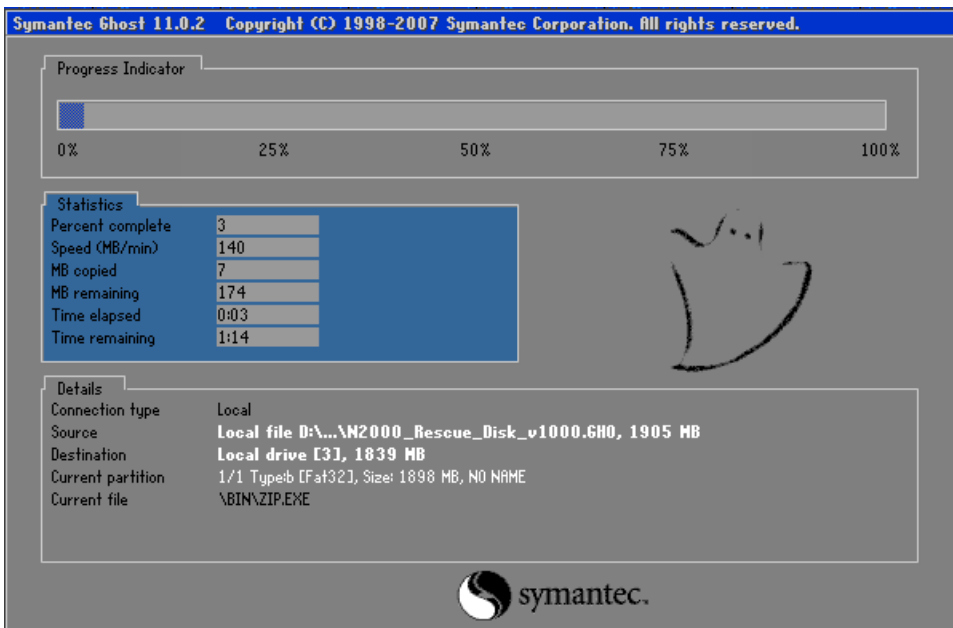
5. Select the destination to CF card and click then **OK**



6. Recover the rescue ghost file, **rescue.gho**, into CF card and then click **OK**



7. The rescue CF card has been done



# 9. iPPC Updates

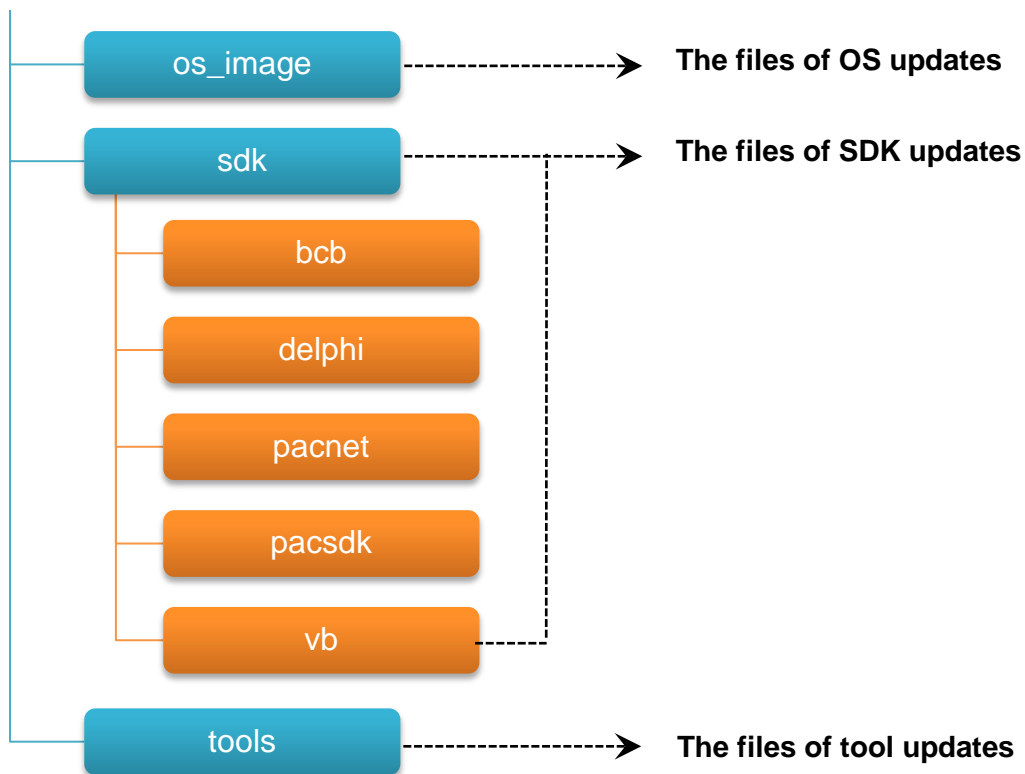
This chapter provides information of the iPPC OS and SDKs, and a guided tour that demonstrates the steps needed to update the iPPC OS and SDKs.

ICP DAS will continue to add additional features to iPPC OS and SDKs in the future, so we advise you to periodically check the ICP DAS web site for the latest updates.

Both the files of OS updates and SDK updates can be found on the CD that was provided with the package or by downloading the latest version from ICP DAS web site.

CD:\ippc-wes7\

<ftp://ftp.icpdas.com/pub/cd/ippc-wes7/>



## 9.1. Updating the iPPC OS

ICP DAS will continue to add additional features and improve performances to iPPC OS in the future, so we advise you to periodically check the ICP DAS web site for the latest updates.

The information can be obtained from:

[http://ftp.icpdas.com/pub/cd/ippc-wes7/os\\_image/](http://ftp.icpdas.com/pub/cd/ippc-wes7/os_image/)

Free feel to contact us to get the latest version of OS image.

E-mail: [service@icpdas.com](mailto:service@icpdas.com)

## 9.2. Updating the iPPC SDK

ICP DAS will continue to include more functionality and API calls to iPPC SDK in the future, so we advise you to periodically check the ICP DAS web site for the latest updates.

### To update the iPPC SDK

1. Download the latest version of the pacsdk.dll file

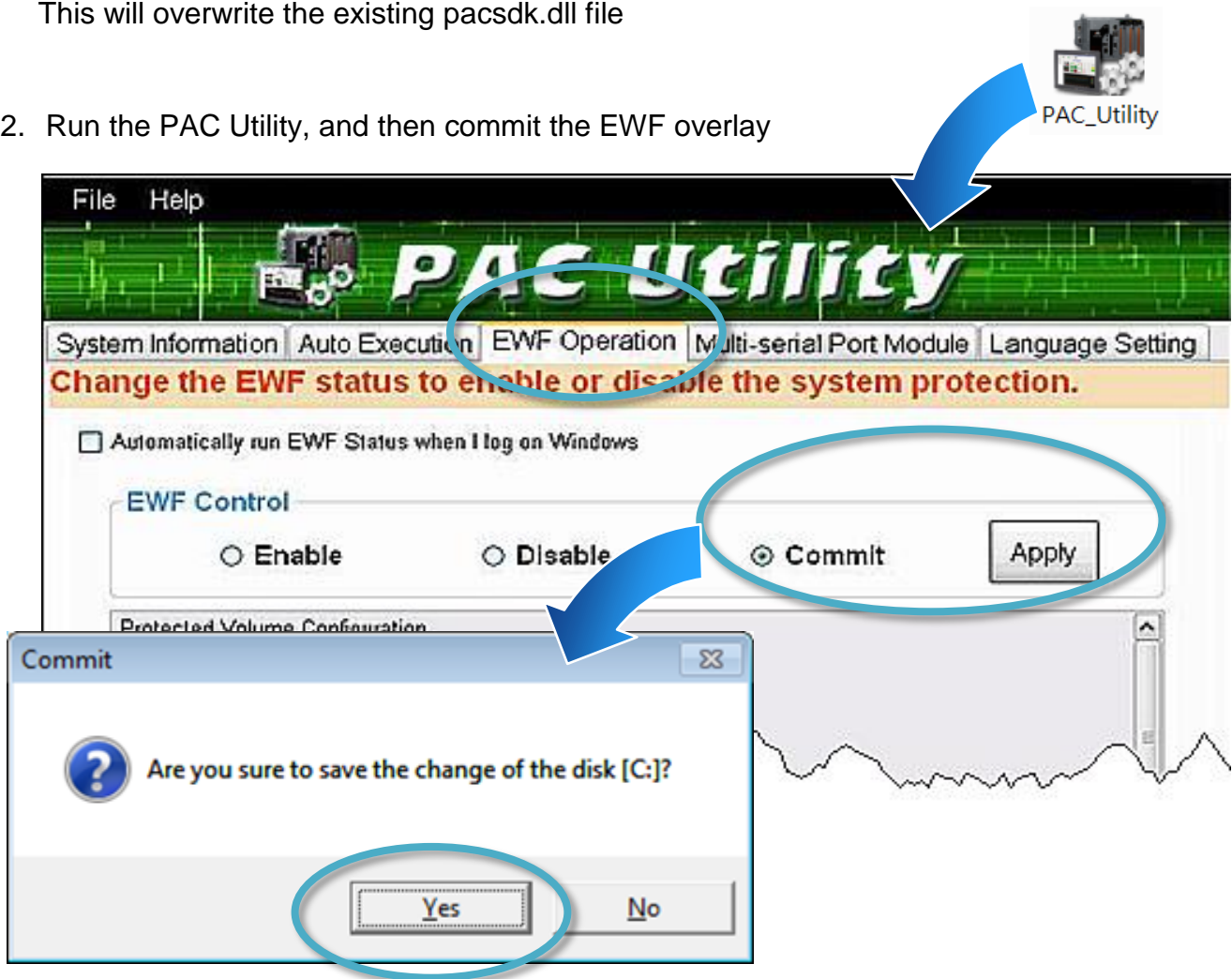
The latest version of the pacsdk.dll file can be obtained from ICP DAS web site.

<http://ftp.icpdas.com/pub/cd/ippc-wes7/sdk/pacsdk/>

Copy the downloaded file, pacsdk.dll into the C:\Windows\System32\ folder.

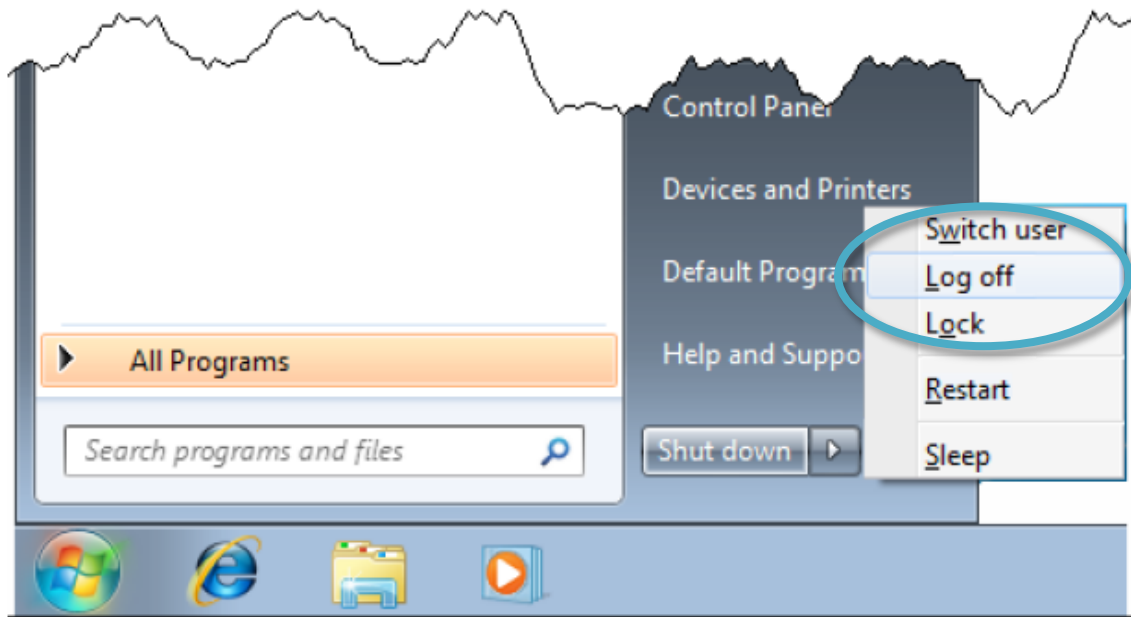
This will overwrite the existing pacsdk.dll file

2. Run the PAC Utility, and then commit the EWF overlay





3. Log off iPPC and then login again



## 9.3. Updating the PAC Utility

ICP DAS will continue to add more functionality and support to the PAC utility in the future, so we advise you to periodically check the ICP DAS web site for the latest updates.

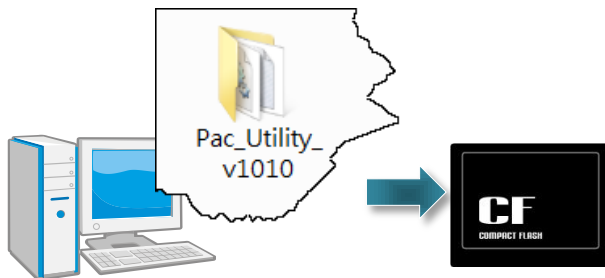
### To update the PAC Utility

1. Download the latest version of the PAC utility file in PC or a laptop

The latest version of the PAC utility file can be obtained from ICP DAS web site.

[http://ftp.icpdas.com/pub/cd/ippc-wes7/tools/pac\\_utility/](http://ftp.icpdas.com/pub/cd/ippc-wes7/tools/pac_utility/)

2. Extract the downloaded file, and then copy the file folder to the CF card

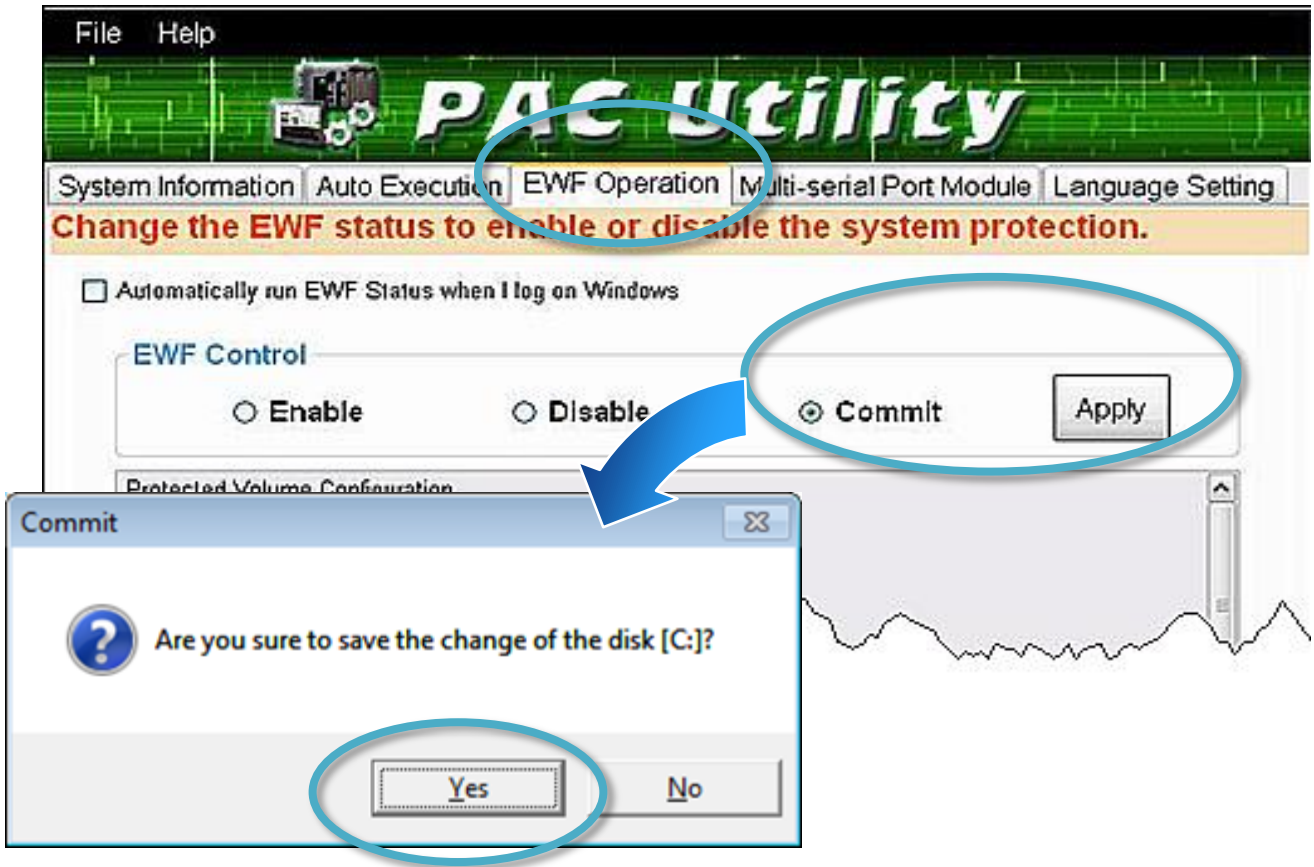


3. Plug the Rescue CF card into CF slot in iPPC

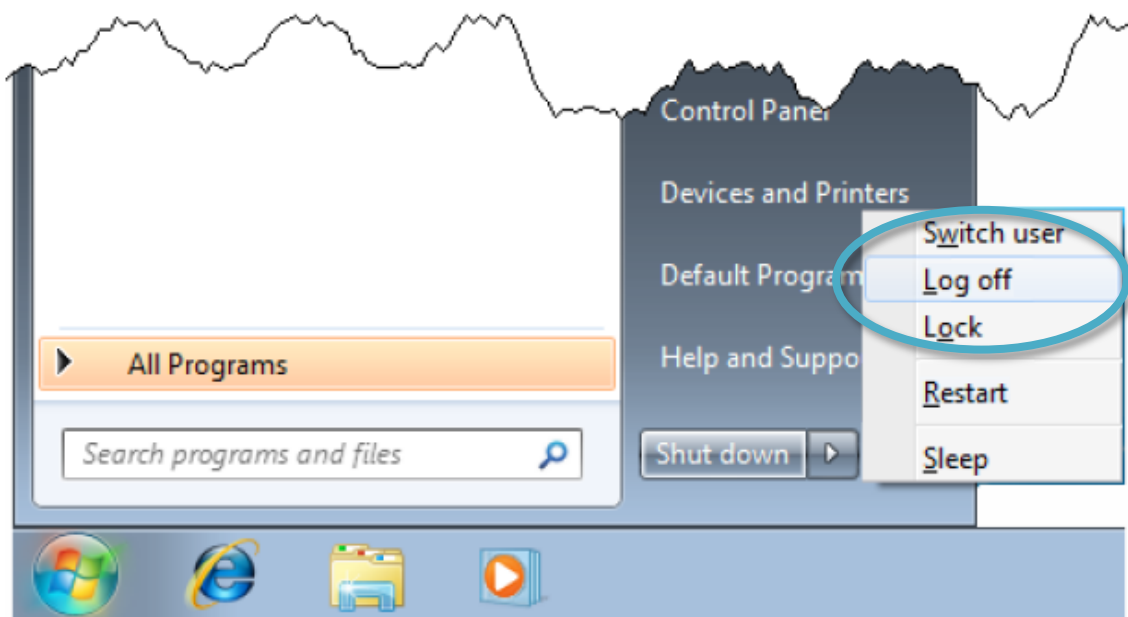


4. Copy the file folder into C:\icpdas\, and then delete the older, existing file folder

5. Run the PAC Utility, and then commit the EWF overlay



6. Log off iPPC and then login again



# 10. iPPC Download Center

This chapter provides a brief introduction of the iPPC download center.

iPPC has a download center where you can access the latest version of the software, tools, demo programs, and related information.

The iPPC Download Center can be found at:

[http://www.icpdas.com/root/support/download/pac/ippc/ippc\\_download\\_driver\\_os\\_image.html](http://www.icpdas.com/root/support/download/pac/ippc/ippc_download_driver_os_image.html)

## *iPPC Download Center*

**Note:**

Before you download any program, please read the notes of each online program first to avoid the confused situation. The published date and indicated requirement of a program can help user to determine the compatibility for your iPPC.

The screenshot shows a web interface with five navigation tabs: "Driver & OS Image", "PacSDK", "Tools", "Demo", and "Documents". The "Driver & OS Image" tab is selected. Below the tabs, the heading "OS images and driver" is displayed. A prominent orange banner reads "OS Image". Underneath, the version "1.0.1.0." is listed with the release date "(Released at Mar. 2015)". A message states "Please contact ICP DAS to get OS recovery image" with the email "service@icpdas.com". A "Features" section follows, listing: Full Win32 API, Remote Desktop Protocol 5.1, Silverlight 5, Enhanced Write Filter, and MS SQL Server 2012 Express.

# Appendix

## A. I-8K Modules and I-87K Modules

This chapter provides a brief overview of the different between the I-8K series modules and I-87K series modules.

I-8K and I-87K modules provide the option to expand the local I/O to expansion I/O slots and the bus type for the modules can be either parallel (high profile I-8K series) or serial (high profile I-87K series).

The differences between the I-8K series modules and I-87K series modules are as follows.

Item	I-8K Series	I-87K Series
Microprocessor	No	Yes (8051)
Communication Interface	Parallel Bus	Serial Bus
Communication Speed	Fast	Slow
Latched DI Function	No	Yes
Counter Input (for digital input modules)	No	Yes (100 Hz)
Power-on Value	No	Yes
Safe Value	No	Yes
Programmable Slew-Rate for AO modules	No	Yes

## B. Revision History

This chapter provides revision history information to this document.

The table below shows the revision history.

Revision	Date	Description
V1.0.0	May 2015	Initial issue
V1.0.1	December 2015	Deleted the product information about the iPPC-5731-WES7
V1.0.2	January 2016	Modified the directory name of the CD and FTP.