

User's Manual [Version 2.30]



OPC®, the OPC-Logo and OPC[™] Foundation are trademarks of the OPC Foundation. (www.opcfoundation.org) Microsoft®, Microsoft .NET[™], VisualStudio.NET[™] and Microsoft Windows[™] are trademarks of the Microsoft Corporation (www.microsoft.com)

Warranty

All products manufactured by ICP DAS Inc. are warranted against defective materials for a period of one year from the date of delivery to the original purchaser.

Warning

ICP DAS Inc. assumes no liability for damages consequent to the use of this product. ICP DAS Inc. reserves the right to change this manual at any time without notice. The information furnished by ICP DAS Inc. is believed to be accurate and reliable. However, no responsibility is assumed by ICP DAS Inc. for its use, or for any infringements of patents or other rights of third parties resulting from its use.

Copyright

Copyright 2003-2011 by ICP DAS Inc., LTD. All rights reserved worldwide.

Trademark

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

License

The user can use, modify and backup this software on a single machine. The user may not reproduce, transfer or distribute this software, or any copy, in whole or in part.

Table of Contents

1	NAPOP	C_CE5 DA SERVER	4
1	I.1 INST	ALL NAPOPC_CE5 DA SERVER	5
1	I.2 Fun	CTION OVERVIEW	6
	1.2.1	Search Modules	6
	1.2.2	Monitoring Devices	
	1.2.3	Adding a New Device	
	1.2.3.1	Adding a New I-8K/I-87K Embedded Module	
	1.2.3.2	Adding a New XW I/O Embedded Module	
	1.2.3.3	Adding a New Remote I/O Module	
	1.2.3.4	Adding a New Enert Device	
	1236	Adding a New Modbus RTU Controller	21
	1.2.3.7	Adding a New Modbus ASCII Controller	
	1.2.3.8	Adding a New Modbus TCP Controller	
	1.2.4	Adding a New Group	
	1.2.5	Adding a New Tag	
	1.2.5.1	Adding New Tags For I-7K/8K/87K/ZigBee/FRnet/XW Module	
	1.2.5.2	Adding a New Tag For Internal Device	
	1.2.5.3	Adding a New Tag For Modbus Device	
	1.2.5.4	Scaling Settings	
	1.2.6	Adding Multi Tags for Modbus Device	
	1.2.7	Read/Write the Tags	
	1.2.8	Editing A Device/Group/Tag properties	
	1.2.9	Deleting A Device/Group/Tag	
	1.2.10	Generating Tags	
	1.2.11	Services Setup	
	1.2.12	Rule Script Editor	
	1.2.13	File	
	1.2.14	About	
	1.2.15	Minimize NAPOPC_CE5	
2	QUICK	START	
3	REMOT	E ACCESSING	
-	l Sve	TEM DEOLIDEMENT	17
-	3.1 5.13 3.2 Con	IEIGUDING DCOM	
-	321	Configuring On the Server Site (WinPAC)	
	3.2.1	Configuring On the Client Site (Will AC)	
	3.2.2	Configuring On the Client Site (TC)	
	22.5	Configuring On the Client Site (MinDAC)	
	3.2.4	Configuring On the Client Sile (winFAC)	
4	THE AP	PLICATION OF NAPOPC_CE5	72
4	4.1 NAI	POPC CE5 WITH OPC CLIENT	
4	1.2 NAI	POPC CE5 WITH MODBUS RTU/TCP CLIENT	
	4.2.1	Supported Modbus Commands	
4	1.3 NAI	POPC CE5 WITH NAPOPC ST/NAPOPC XPE	
2	14 NAI	POPC CE5 with User Application	79
	441	Ouicker API for eVC++ Developer	79
	4.4.1.1	System Function.	
	4.4.1.2	QuickerIO Function	
	4.4.1.3	Modbus Function	
	4.4.1.4	UserShare Function	
		Quicker API for VR NET/VC# NET Developer	112
	4.4.2	Quicker AI 1 Jor V B.INET/ V C#.INET Developer	
Z	4.4.2 4.5 NAI	POPC_CE5 with Rule Script	
2	4.4.2 4.5 NAI 4.5.1	POPC_CE5 WITH RULE SCRIPT	

1 NAPOPC CE5 DA Server

What is NAPOPC CE5 DA Server? NAPOPC CE5 DA Server is an integrated omnibus software package which combines OPC, Modbus TCP, Modbus RTU services, and Scankernel together. The particular design, "Rule Script", lets user can quickly establish a DCS control system with logic control, multi-communication services.

For UI design, NAPOPC_CE5 uses an explorer-style user interface to display a hierarchical tree of modules and groups with their associated tags. A group can be defined as a subdirectory containing one or more tags. A module may have many subgroups of tags. All tags belong to their module when they are scanned to perform I/O. (The "OPC" stands for "OLE for Process Control" and the "DA" stands for "Data Access".)

For software use, NAPOPC CE5 creates a set-up procedure requiring at most three steps for different kinds of users. This kind of procedure simplifies the designing process for the programmer, and ensures the stability and efficiency of control system.

NAPOPC CE5 not only can map the physical I/O to a specific Modbus address automatically, but also allows users to define their own variables into it. Therefore users can develop their own application program with eVC++, VB.NET, and VC#.NET programming language via Modbus RTU and Modbus TCP protocol to share their specific data with Modbus client. Moreover, users can operate the NAPOPC CE5 and NAPOPC ST/NAPOPC XPE in coordination to create a fantastic solution integrating SCADA software with on-line data.



The main program of NAPOPC_CE5 is "NAPOPCSvr_CE5.exe". It automatically loads dynamic libraries under \System_Disk\Tools\ NAPOPC_CE5 folder and calls functions in these libraries.

1.1 Install NAPOPC_CE5 DA Server

You have to execute "NAPOPC_CE5Boot.exe" in the \System_Disk\Tools\ NAPOPC_CE5 of WinPAC-8000/5000 when you use NAPOPC_CE5 first time, after that, "NAPOPC_CE5Boot.exe" will register NAPOPC_CE5 automatically. Moreover, if you want to execute the "NAPOPCSvr_CE5.exe " automatically while WinPAC-8000/5000 boots up, please refer to the "Auto Execution" function at "3.5 WinPAC Utility" of winpac_8x4x_user_manual_v1.9.0.pdf and add path of " NAPOPC_CE5Boot.exe" into "Auto Execution".

NOTE: After above steps, please use "Save and Reboot" function at WinPAC Utility to save registry exactly.

Address System_Disk\tools	WAPOPC_CE5			
Name	Size	Туре	Date Modified	
DCON_CE	18.5KB	Application Extension	6/29/2010 12:27 PM	
🖺 history_v2.14	2.55KB	Text Document	6/29/2010 12:27 PM	
NBTool	30.5KB	Application Extension	6/29/2010 12:27 PM	
🔊 module_7K	23.5KB	INI File	6/29/2010 12:27 PM	
module_87K	18.6KB	INI File	6/29/2010 12:27 PM	
🔊 module_8K	12.1KB	INI File	6/29/2010 12:27 PM	
module_ET	4.48KB	INI File	6/29/2010 12:27 PM	
module_FRnet	633 bytes	INI File	6/29/2010 12:27 PM	
NAPOPC CES	403 bytes	Batch File	6/29/2010 12:27 PM	
NAPOPC_CE5Boot)	4.50KB	Application	6/29/2010 12:27 PM	
NAPOPCSvr_CE5	900KB	Application	6/29/2010 12:27 PM	
opccomn_ps	60KB	Application Extension	6/29/2010 12:27 PM	
S OPCDAAuto	112KB	Application Extension	6/29/2010 12:27 PM	
S OPCProxy	72.5KB	Application Extension	6/29/2010 12:27 PM	
🔊 pac_i8017hw	18.5KB	Application Extension	6/29/2010 12:27 PM	
🔊 pac_i8024w	11KB	Application Extension	6/29/2010 12:27 PM	
🔊 pac_i8084w	16.5KB	Application Extension	6/29/2010 12:27 PM	
🔊 pac_i8172w	6.50KB	Application Extension	6/29/2010 12:27 PM	
N Quicker	11KB	Application Extension	6/29/2010 12:27 PM	
🔊 QuickerNet	5.50KB	Application Extension	6/29/2010 12:27 PM	
TRegEdit	25.5KB	Application	6/29/2010 12:27 PM	
megsvr32	6KB	Application	6/29/2010 12:27 PM	
🖺 Rule	0 bytes	Text Document	6/29/2010 12:28 PM	
StOleProxy	311KB	Application Extension	6/29/2010 12:27 PM	
🔊 Test1	24KB	TDB File	6/29/2010 12:36 PM	
JUARTCE	19.5KB	Application Extension	6/29/2010 12:27 PM	
NinpacSDK	96KB	Application Extension	6/29/2010 12:27 PM	

Fig 1.1-1

After that, you execute the main program "NAPOPCSvr_CE5.exe " which would load dynamic libraries under \System_Disk\Tools\ NAPOPC_CE5 folder by itself to start NAPOPC_CE5.

If the files under "\System_Disk\Tools\NAPOPC_CE5" loss or crash, please copy the files under "/napdos/wp-8x4x_ce50/system_disk/tools/NAPOPC_CE5/" in the CD to "\System_Disk\Tools\NAPOPC_CE5" by yourself.

1.2 Function Overview

1.2.1 Search Modules

The "Search Modules..." function lets you configure NAPOPC_CE5 automatically. It searches the RS-485 network and embedded modules to find modules and then generates tags automatically. This version of NAPOPC_CE5 not only generates AI/AO, DI/DO, Latched DI and Counter tags but also maps each tag to a unique modbus address.

Step 1: Click on the "Add/ Search Modules..." menu item or the **P** icon to search for modules.

Add	Edit	View	Services				
Nev	v Devic	e	Ctrl+D				
Nev	v Grou	Ctrl+G					
New Tag Ctrl+T							
Search Modules							
Gen	Generate Tags						
Mult	ti Tags						
	Fig	1.2.1	-1				



Eile Add Edit Vie	ew <u>Services Rule H</u> elp X 🗈 🛍
8052_2 Is 8057_3 Dos 87024_4 Aos 187040_5 DIS DICounter DICounter LatchLowDIs LatchHighDIs	Search Modules COM 0 ET-7000 Clear Modules COM 0 ET-7000 Clear Modules Baud Rate Searching 921600 460800 230400 115200 921600 460800 230400 115200 9600 921600 460800 230400 115200 921600 460800 230400 115200 921600 460800 230400 115200 921600 460800 230400 115200 921600 460800 19200 9600 921600 2400 1200 9600 4800 2400 1200 9600 Select All Clear All Clear All Address (0 to 255) Checksum Timeout (mSec) Start 0 Disabled 500 End 255 Enabled 500 Status 0% A:0 B:115200 S:0 ID:87040 Exit

Fig 1.2.1-2

Step 3: If you want to search the I-8K I/O modules or XW-board plugged in the WinPAC-8000/5000, you have to check the "Local Search" field. "COM 0" is for searching I-87K I/O modules or XW-board plugged in the WinPAC-8000/5000.

COM 0 Clear Module Baud Rate Searching 921600 460800 230400 115200 57600 38400 19200 9600 4800 2400 1200 Select All Clear All Address (0 to 255) Start 0 Disabled End 255 Checksum Status 0% A:1 B:115200 5:0 EC:13	rch Modules	
Baud Rate Searching 921600 460800 230400 115200 97600 38400 19200 9600 4800 2400 1200 9600 Select All Clear All Clear All Address (0 to 255) Checksum Timeout (mSec) Start 0 255 500 Status 0% A:1 B:115200 S:0 EC:13 500	сом	ET-7000 Clear Modules
921600 460800 230400 115200 57600 38400 19200 9600 4800 2400 1200 9600 4800 2400 1200 9600 Select All Clear All Clear All Timeout (mSec) Start 0 Disabled 500 500 End 255 Enabled 500 500 Status 0% A:1 B:115200 S:0 EC:13 500 500	Baud Rate Search	ning
57600 38400 19200 9600 4800 2400 1200 9600 Select All Clear All Clear All Address (0 to 255) Checksum Timeout (mSec) Start O Disabled 500 End 255 Enabled 500 Status 0% A:1 B:115200 S:0 EC:13 500	921600	460800 230400 🔽 115200
Address (0 to 255) Start 0 End 255 Status 0% A:1 B:115200 S:0 EC:13 1200 Clear All Clear All Timeout (mSec) 500 500 500	57600	38400 🔲 19200 📃 9600
Select All Clear All Address (0 to 255) Start 0 End 255 Checksum Disabled Enabled Status 0% A:1 B:115200 S:0 EC:13	4800	2400 1200
Address (0 to 255) Start 0 End 255 Status 0% A:1 B:115200 S:0 EC:13	Select A	
Address (0 to 255) Start 0 End 255 Status 0% A:1 B:115200 S:0 EC:13 Checksum Disabled Enabled Status 0% A:1 B:115200 S:0 EC:13		
Status 0% A:1 B:115200 S:0 EC:13	Address (0 to 25 Start 0 End 255	5) Checksum Timeout (mSec) Disabled 500 Enabled
Status U% A:1 B:115200 S:0 EC:13		
	Status U% A:1	B:115200 S:0 EC:13
	Search	Stop Exit
Search Stop Exit		
Search Stop Exit		Fig 1.2.1-3

Step 4: If you want to search the I-7K/I-87K remote I/O modules via RS-232, you have to choice "COM 1" and uncheck the "Local Search".

Search Modules	×
COM ET-7000 Clear Modules	1
Baud Rate Searching	,
921600 460800 230400 📝 115200	
57600 38400 19200 9600 4800 2400 1200	
Select All Clear All	
Address (0 to 255) Checksum Timeout (mSec) Start 0 Disabled 500 End 255 Enabled	
Status	
Search Stop Exit	

Fig 1.2.1-4

Step 5: If you want to search the I-7K/I-87K remote I/O modules via RS-485 and ET-7000 modules via Ethernet, you have to choice "COM 2" and "ET-7000", and uncheck the "Local Search".

COM 2			
Baud Rate Searching Local Searching 921600 460800 230400 ✓ 57600 38400 19200 9600 4800 2400 1200 9600 Select All Clear All Clear All Address (0 to 255) Checksum Timeout (mSe Start 0 Ø 500 End 255 Enabled 500	COM 2	Clear	Module
Baud Rate Searching	<u> </u>	Local	Search
921600 460800 230400 ♥ 115200 57600 38400 19200 9600 4800 2400 1200 9600 Select All Clear All Clear All Address (0 to 255) Checksum Timeout (mSe Start 0 Disabled 500 End 255 Enabled 500	Baud Rate Sear	ching 👘 👘	
□ 57600 □ 38400 □ 19200 □ 9600 □ 4800 □ 2400 □ 1200 □ 9600 Select All Clear All Clear All Address (0 to 255) Checksum Timeout (mSe Start □ □ □ End 255 □ □ Status □ □ □	921600	460800 230400 🔽 1:	15200
Address (0 to 255) Start 0 End 255 Status	57600	38400 19200 96	500
Select All Clear All Address (0 to 255) Start 0 End 255 Status	4800	2400 1200	
Address (0 to 255) Start 0 End 255 Status			
Address (0 to 255) Start 0 End 255 Checksum Disabled Enabled Status	Select /	All Clear All	
Address (0 to 255) Checksum Timeout (mSe Start 0 Disabled 500 End 255 Enabled Status			
Start 0 Disabled 500 End 255 Enabled	Address (0 to 25	55) – Checksum – Timeou	t (mSec
End 255 Enabled		Timood	- (111500
End 255	a	Disabled For	
Status	Start 0	Disabled 500)
Status	Start 0 End 255	Disabled 500)
	Start 0 End 255	Disabled 500	
	Start 0 End 255 Status	Disabled 500)
	Start 0 End 255 Status	Disabled 500 Enabled)

Fig 1.2.1-5

COM :

Specifies which "COM" port number to search. The default value is 0 and the valid range is from 0 to 255. Please verify the "COM" port number that the RS-485 network is connected to.

Modules	COM 0	COM 1	COM 2	COM 3	COM 4
Local I-87K/XW-board	Yes	-	-	-	-
Remote I-7K/I-87K via RS-232	-	Yes	-	-	-
Remote I-7K/I-87K via RS-485	-	-	Yes	-	-
Remote I-7K/I-87K via RS-232/485	-	-	-	Yes	-
Remote I-7K/I-87K via RS-232	-	-	-	-	Yes

ET-7000:

If this field is checked, NAPOPC can search not only the modules communicating via COM port but also ET-7000 modules via Ethernet automatically.

Clear Modules:

Modules can be added many times. If this field is checked, it removes all modules from the list window before searching. Checking this box prevents adding a duplicate module. The default setting is "not checked".

Local Search:

If this field is checked, it searches the I-8K modules or XW-board plugged in the WinPAC-8000/5000 first.

Baud Rate Searching:

Specifies which "Baud Rate" will be looking for. The default setting is "9600".

Naturally, if multiple baud rates are checked, the search will be longer. NAPOPC_CE5 has to close and then reopen the COM ports to communicate with modules when searching for multiple baud rates. This also reduces communication performance. Thus, using the same baud rate and COM port number for every module is highly recommended.

Select All:

Sets all the "Baud Rate" fields to be checked. Please refer to the above "Baud Rate Searching" section.

Clear All:

Sets all the "Baud Rate" fields to be unchecked (nothing to search). Please refer to the above "Baud Rate Searching" section.

Address/Start:

Specifies the starting address. The default value is 0 and the valid range is from 0 to 255. It won't search for an address below these settings.

Address/End:

Specifies the ending address. The default value is 255 and the valid range is from 0 to 255. It won't search for an address greater than these settings.

Checksum/Disabled:

If this field is checked, modules are searched with no checksum. If both the "Disabled" and "Enabled" fields were unchecked, the search would be undefined.

Checksum/Enabled:

If this field is checked, it searches modules with checksum. If both the "Disabled" and "Enabled" fields were unchecked, again, the search would be undefined.

Timeout:

Specifies the timeout value of communication to each module. The default value is 200 (equal to 0.2 Seconds), measured in millisecond(s) [0.001 Second(s)]. After a module has been found, this timeout value will also be recorded for further use.

Users can reduce this value to shorten the search time. Be careful. A shorter search time may cause communication failure.

Status:

It shows the searching status (includes: progress in %, Address in "A:??", Baud-Rate in "B:???", Checksum in "S:?" and Error-Code in "EC:??").

The timeout error code is 15. In most cases, it indicates no module has responded to the current command.

Search:

After setting the above options, click this button to search. The window will be closed automatically when completed.

Stop:

During the search, users can click the button to stop. The window will stay on the screen after the search is cancelled.

Exit:

Users can click the button to close the window.

Step 6: After the search, the discovered modules will be listed on the Device-Window (left side). Users can also see the tags on the Tag-Window (right side) generated by the "Search Modules..." function automatically.



The "Search Modules..." function generates "Digital Input", "Digital Output" "Bit Input" or "Bit Output" tags.

The "Digital Input" and "Digital Output" tags use one communication to read the status of all channels, while the "Bit Input" and "Bit Output" tags use one communication to read only one-channel status. The "Digital Input" and "Digital Output" tags have better performance than the "Bit Input" and "Bit Output" tags. Using the "Digital Input" and "Digital Output" tags to access modules is highly recommended.



Fig 1.2.1-7

1.2.2 Monitoring Devices

Use the "Monitor" function to see values of tags by checking the "View/ Monitor" menu item. Uncheck the item to stop monitoring.

Step 1: Click the "View/ Monitor" menu item to enable monitor.

<u>File Add Edit View Help</u>				
	ŀ.			
	Name	Type	Channel/Loca	Scal
📙 🂑 DOs	8 Ch00	Bit Input	0	· · · ·
B051_2	8 Ch01	Bit Input	1	
	8 Ch02	Bit Input	2	
P- 11 8056_3	8 Ch03	Bit Input	3	
DOs	8 Ch04	Bit Input	4	
P-11 8024_4	8 Ch05	Bit Input	5	
AOs	8 Ch06	Bit Input	6	
P-11 8017_5	8 Ch07	Bit Input	7	
AIs	8 Ch08	Bit Input	8	
⊡ 1 7017_6	8 Ch09	Bit Input	9	
📥 AIs	8 Ch10	Bit Input	10	
	8 Ch11	Bit Input	11	
	8 Ch12	Bit Input	12	
	8 Ch13	Bit Input	13	
	8 Ch14	Bit Input	14	
	8 Ch15	Bit Input	15	
	Fig 1.2.2	2-1		

Step 2: Select the "Als" group in the Device-Window (left side) to monitor its own Analog -Input tags.

8064_1	Name	Type	Channel/Loca	Scaling	Value
📙 🏀 DOs	8 Ch00	Analog Input	0	3	-0.00274
1 8051_2	8 Ch01	Analog Input	1		0.0012207
📙 🃸 DIs	8 Ch02	Analog Input	2		-0.00274
1 8056_3	8 Ch03	Analog Input	3		-0.00152
📙 🃸 DOs	8 Ch04	Analog Input	4		-0.00274
1 8024_4	8 Ch05	Analog Input	5		0.0109863
📥 🚵 AOs	8 Ch06	Analog Input	6		0.0012207
1 8017_5	8 Ch07	Analog Input	7		0.0024414:

Fig 1.2.2-2

Step 3: Select the "8064" module on the Device-Window to monitor its own Digital-Output tags.

<u>File A</u> dd <u>E</u> dit <u>V</u> iew	Help				
	3				
	Name	Type	Channel/Loca	Scaling	Value
Image: Constraint of the second se	8 DO	Digital Output	*	2	68

Fig 1.2.2-3

1.2.3 Adding a New Device

NAPOPC_CE5 provides three kinds of device, "DCON Device", "FRnet Device", and "Modbus Device" to be added. The "DCON Device" includes "I-8K/87K Embedded Modules", "Remote I/O Modules", and "Internal Device". The "Internal Device" could be the intermediary container between several user application programs or the intermediary device designing "Rule Script". The "FRnet Device" supports ICP DAS FRnet modules. The "Modbus Device" supports "Modbus RTU", "Modbus ASCII", and "Modbus TCP" protocol. NAPOPC_CE5 provides multi-thread communication via COM port and Ethernet. The maximum number of Modbus TCP master communication thread is limited to 32 by default.

1.2.3.1 Adding a New I-8K/I-87K Embedded Module

Step 1: Click on the "Add/ New Device..." menu item or the 🖻 icon to add a new module.

Add	Edit	View	Services					
Nev	v Devic	e	Ctrl+D					
Nev	v Grou	p	Ctrl+G					
Nev	v Tag		Ctrl+T					
Sea	Search Modules							
Gen	ierate i	Tags						
Mult	ti Tags							
	Fig 1	1.2.3.1	1-1					

Step 2: The "Select Device" window pops up.

Step 3: Click on the "DCON" radio button.

Step 4: Click the "I-8K/I-87K Embedded Modules" radio button.

DCON	(FRne	et	OMo	dbus	
evice Name	Devic	e8		Control of		
1-8K/87K	Embedd	ed Mod	ules			
Module Set	ing —					
Module	8013	~	Range	None		~
Slot	0	(0-7	for 8K/8	37K Mod	ules)	
Timeout	500	 mSe	с			
Module Set Module Address Slot	ing 0 0	(0-; (0-;	Timeou 255) Cl 7 for 8K/	t (mSec) necksum 87K Mod	500 Disabl	e 📈
COM Port S	etting					
COM Port	1	<u> </u>	Pari	ty Nor	ie	<u></u>
Baud Rate	1 9600		Data B Stop B	its 8 its 1		4
🔵 Internal	Device					
Simulate	I/O					
Pending	Time 0		n	ns		

Fig 1.2.3.1-2

Device Name:

Names with spaces or punctuation such as "|!.," cannot be used within a module name. The clients use the "Device Name" and "Tags" to access its value. The "Device Name" can not be the same as any other module.

Module:

User can click on the ComboBox to select a Module ID.

Timeout:

Specifies timeout (Response time) value for this module. A smaller timeout value may cause communication failure and a greater timeout value may reduce the performance of the client program.

Slot:

The WinPAC-8000 has 4 or 8 slots to plug in. This "slot" field indicates the slot number that the I/O module used. The valid range is from 0 to 7.

Range:

It is for I-8017 and I-8024 module settings. Please refer to module manual to choose correct range.

Simulate I/O:

The "Simulate I/O" checkbox switches to a simulator of reading I/O. Since the simulator does not open the TCP/IP or COM port, it is an easy way to work with the server, to configure tags or to connect clients without requiring any hardware.

Pending Time:

Minimum interval time between two access. To activate this function, NAPOPC_CE5 can work under optimized communication performance. If this module only needs to be accessed 1 time per 5 seconds. You can set pending time as 5000 ms. NAPOPC_CE5 will automatically spread time resource to other modules which are connected with each other.

Step 5: Click on the "OK" button to add this new module.

1.2.3.2 Adding a New XW I/O Embedded Module

Step 1: Click on the "Add/ New Device..." menu item or the 🖻 icon to add a new module.

Add	Edit	View	Services
Nev	v Devic	:e	Ctrl+D
New	v Grou	р	Ctrl+G
New	v Tag.	a -	Ctrl+T
Sear	rch Mo	dules	
Gen	erate	Tags	
Mult	i Tags		

Fig 1.2.3.2-1

Step 2: The "Select Device" window pops up.

Step 3: Click on the "DCON" radio button.

Step 4: Click the "XW I/O Embedded Modules" radio button.

DCON		ot		
JUCON	Orki	el		\$
evice Name	Device3			
💽 XW I/O Emi	bedded Modu	les		
Module Settin	g			
Module 🔀	W107 🔽			
Timeout 5	00 mSe			
inneede j	mov	~		
) Modules			
Module Settin	g g			
Module 7	7011 🔍	Timeout	(mSec) 50	0
		and and	t Die	
Address	(0-	-255) Che	ecksum (PPB	sable 🗸
Slot	(0-	-255) Che -7 for 8K/8	7K Modules	sable 🔛
Slot COM Port Set	(0- (0-	-255) Che -7 for 8K/8	7K Modules)
Slot COM Port Set	(0- (0- ting	-255) Che -7 for 8K/8 Parity	7K Modules)
Slot COM Port Set	(0- (0- ting L	-255) Che -7 for 8K/8 Parity Data Bit	7K Modules	
Slot COM Port Set	(0- (0- ting L V	-255) Che -7 for 8K/8 Parity Data Bit Stop Bit	7K Modules 7K Modules , None s 8 s 1	
Slot Slot COM Port Set	(0- (0- ting 1 2 2600 2	-255) Che -7 for 8K/8 Parity Data Bit Stop Bit	7K Modules 7K Modules 7K Mone 5 8 5 1	
Slot Slot COM Port Set	(0- ting L V P600 V	-255) Che -7 for 8K/8 Parity Data Bit Stop Bit	7K Modules 7K Modules 5 8 5 1	

Device Name:

Names with spaces or punctuation such as "|!.," cannot be used within a module name. The clients use the "Device Name" and "Tags" to access its value. The "Device Name" can not be the same as any other module.

Module:

User can click on the ComboBox to select a Module ID.

Timeout:

Specifies timeout (Response time) value for this module. A smaller timeout value may cause communication failure and a greater timeout value may reduce the performance of the client program.

Simulate I/O:

The "Simulate I/O" checkbox switches to a simulator of reading I/O. Since the simulator does not open the TCP/IP or COM port, it is an easy way to work with the server, to configure tags or to connect clients without requiring any hardware.

Pending Time:

Minimum interval time between two access. To activate this function, NAPOPC_CE5 can work under optimized communication performance. If this module only needs to be accessed 1 time per 5 seconds. You can set pending time as 5000 ms. NAPOPC_CE5 will automatically spread time resource to other modules which are connected with each other.

Step 5: Click on the "OK" button to add this new module.

1.2.3.3 Adding a New Remote I/O Module

Step 1: Click on the "Add/ New Device..." menu item or the 🔤 icon to add a new module.

Add	Edit	View	Services
Nev	/ Devic	:e	Ctrl+D
New	/ Grou	p	Ctrl+G
Nev	/ Tag.		Ctrl+T
Sear	rch Mo	idules	
Gen	erate	Tags	
Mult	i Tags		
		1 2 2 2	

Fig 1.2.3.3-1

Step 2: The "Select Device" window pops up.

Step 3: Click on the "DCON" radio button.

Step 4: Click the "Remote I/O Modules" radio button.

Hatt Dovito	0	
DCON	O FRnet O Modbu	ıs
evice Name	Device13	
] I-8K/87K Em	bedded Modules	
Module Setting		
Module 80	13 🔜 Range None	
Slot 0	(0-7 for 8K/87K Modules)
Timeout 50	0 mSec	
Module 20 Address 0 Slot 0	0111	oo isable 🔽 s)
COM Port Setti	ng	
COM Port 1	Parity None	2
Baud Rate 9	600 💽 Data Bits 8 Stop Bits 1	× ×
Internal Dev	vice	
Simulate I/C)	
10101		

Fig 1.2.3.3-2

Device Name:

Names with spaces or punctuation such as "|!.," cannot be used within a module name. The clients use the "Device Name" and "Tags" to access its value. The "Device Name" can not be the same as any other module.

Module:

User can click on the ComboBox to select a Module ID.

Address:

Specifies a Module Address for this module. The default value is 0 and the valid range is between 0 to 255. This field is disabled for the 8000 subdevices. It will use the 8000 main-device's address.

Slot:

The 8000 main-device has 4 or 8 slots for the 8000 sub-device to plug in. This "slot" field indicates the slot number that the 8000 sub-device is using. The valid range is from 0 to 7. This field is disabled for 8000 main-device and 7000 series modules.

Timeout:

Specifies timeout (Response time) value for this module. A smaller timeout value may cause communication failure and a greater timeout value may reduce the performance of the client program. This field is disabled for the 8000 sub-devices and it will use the 8000 main-device's timeout value.

Checksum:

This checksum field must match the hardware setting. A mismatch will always cause a communication failure with this module.

This field is disabled for the 8000 sub-devices and it will use the 8000 main-device's checksum.

COM Port:

Specifies the COM port to be used. Please verfiy which COM port number that the RS-485 network is using. Wrong settings will always cause communication failure.

This field is disabled for the 8000 sub-devices. It will use the 8000 main-device's COM port setting.

Baud Rate:

Specifies the baud rate to be used. Verify the module's current baud rate. A wrong setting will always cause communication error for this module. This field is disabled for the 8000 sub-devices. It will use the 8000

This field is disabled for the 8000 sub-devices. It will use the 8000 main-device's baud rate.

Simulate I/O:

The "Simulate I/O" checkbox switches to a simulator of reading I/O. Since the simulator does not open the TCP/IP or COM port, it is an easy way to work with the server, to configure tags or to connect clients without requiring any hardware. This field is disabled and not used for the 8000 maindevice.

Pending Time:

Minimum interval time between two access. To activate this function, NAPOPC_CE5 can work under optimized communication performance. If this module only needs to be accessed 1 time per 5 seconds. You can set pending time as 5000 ms. NAPOPC_CE5 will automatically spread time resource to other modules which are connected with each other.

OK:

Click on the "OK" button to add the new module setting.

Cancel:

Click on the "Cancel" button to avoid any changes.

Step 5: Click on the "OK" button to add this new module.

1.2.3.4 Adding a New Internal Device

Step 1: Click on the "Add/ New Device..." menu item or the 🖻 icon to add a new module.

Add	Edit	View	Services
Nev	v Devic	e	Ctrl+D
Nev	v Grou	p	Ctrl+G
Nev	v Tag		Ctrl+T
Sea	rch Mo	dules	
Gen	ierate i	Tags	
Mult	i Tags		
	Fig 1	1.2.3.4	4-1

- Step 2: The "Select Device" window pops up.
- Step 3: Click on the "DCON" radio button.
- **Step 4**: Click on the "Internal Device" radio button.

elect Device		OK
O DCON	O FRnet O Modbus	
Device Name	Device13	
O I-8K/87K Em	bedded Modules	
Module Setting		
Module 80	13 Sange None	~
Slot 0	(0-7 for 8K/87K Modules)	
Timeout 50	0 mSec	
Module Setting Module Setting Address 0 Slot 0	Modules D11 Timeout (mSec) 500 (0-255) Checksum Disab (0-7 for 8K/87K Modules)	ile 💉
COM Port Sett	ng	
COM Port	Parity None	~
Baud Rate 9	600 Data Bits 8 Stop Bits 1	
Internal De	vice	
Simulate I/C)	
Pending Tim	e 0 ms	

Fig 1.2.3.4-2

Device Name:

Names with spaces or punctuation such as "|!.," cannot be used within a module name. The clients use the "Device Name" and "Tags" to access its value. The "Device Name" can not be the same as any other module.

Step 5: Click on the "OK" button to add this new module.

1.2.3.5 Adding a New FRnet Device

Step 1: Click on the "Add/ New Device..." menu item or the 🖻 icon to add a new module.

Add	Edit	View	Services
Nev	v Devic	:e	Ctrl+D
Nev	v Grou	p	Ctrl+G
Nev	New Tag Ctrl+T		Ctrl+T
Sea	rch Mo	idules	
Gen	erate	Tags	
Mult	i Tags		
	Eia /	1 2 2 1	5 1

Fig 1.2.3.5-1

Step 2: The "Select Device" window pops up.

Step 3: Click on the "FRnet" radio button.

DCON	FRnet	O Modbus
Device Name De	vice13	
Board Setting Slot	Port 0	-
FRnet module S	etting	ess: 8
	Sender Addres	ss: 0

Fig 1.2.3.5-2

Device Name:

Names with spaces or punctuation such as "|!.," cannot be used within a module name. The clients use the "Device Name" and "Tags" to access its value. The "Device Name" can not be the same as any other module.

Slot:

The WinPAC has 4 or 8 slots to plug in. This "slot" field indicates the slot number that the I/O module used. The valid range is from 0 to 7.

Port:

The "Port" indicates the port number(0 or 1) of I-8172. Every FRnet I/O modules have to use I-8172 as FRnet communication module. Please refer to the I-8172 manual for more information.

FR-:

User can click on the Combo Box to select a FRnet module ID.

Receiver Address:

FRnet communication needs correct hardware configurations for the sender address (SA) and receiver address (RA) on the host controller and the remote module in the network. Please refer to the FRnet manual for more information.

Sender Address:

FRnet communication needs correct hardware configurations for the sender address (SA) and receiver address (RA) on the host controller and the remote module in the network. Please refer to the FRnet manual for more information..

Simulate I/O:

The "Simulate I/O" checkbox switches from reading I/O from the module to running a simulator. Since the simulator does not open the COM port, it is an easy way to work with the server, to configure tags or to connect clients without requiring any hardware.

1.2.3.6 Adding a New Modbus RTU Controller

Step 1: Click on the "Add/ New Device..." menu item or the 💆 icon to add a new module.

Add	Edit	View	Services
Nev	v Devic	e	Ctrl+D
Nev	v Grou	p	Ctrl+G
Nev	New Tag Ctrl+T		Ctrl+T
Sea	rch Mo	dules	
Ger	nerate i	Tags	
Mul	ti Tags		

Fig 1.2.3.6-1

Step 2: The "Select Device" window pops up.

Step 3: Click on the "Modbus" radio button.

Step 4: Click on the "Modbus RTU" radio button.

DEVICE Name Device Controller Setting) FRnet () Modbu	JS
Modbus RTU Modbus ASCII Modbus TCP	ISaGRAF General Modbus Dev IP Address 192.168.2 Port 502	vice Address 1 Timeout 500 55.1 Msg Delay 0 Word Swap
COM Port Setting COM Port 1 Baud Rate 9600		Parity None Data Bits 8 Stop Bits 1
Simulate I/O	ms	

Fig 1.2.3.6-2

Device Name:

Names with spaces or punctuation such as "|!.," cannot be used within a module name. The clients use the "Device Name" and "Tags" to access its value. The "Device Name" can not be the same as any other module.

ISaGRAF:

Connect ISaGRAF controller

General Modbus Device:

Connect general modbus device

Address:

Specifies a Address for this controller. The default value is 1 and the valid range is between 1 to 255.

Timeout:

Specifies timeout (Response time) value for this controller. A smaller timeout value may cause communication failure and a larger timeout value may reduce the performance of the client program.

Msg Delay:

Specifies message delay value for this controller. The default value is 0 ms. A smaller msg delay value may have a higher system loading, but it will have a faster data exchange speed.

Word Swap:

The "Word Swap" checkbox switches the interpretation of 4 Byte values. Sometimes we need to make the checkbox "TRUE" in order to achieve the purpose of Lo-Hi/Hi-Lo communication.

COM Port:

Specifies the COM port to be used. Please verfiy which COM port number that the RS-485 network is using. Wrong settings will always cause communication failure.

Baud Rate:

Specifies the baud rate to be used. Verify the module's current baud rate. A wrong setting will always cause communication error for this controller.

Parity:

Specifies the parity scheme to be used. It is one of the following values.

Value	Description
None	No parity
Even	Even
Odd	Odd

Data Bits:

Specifies the number of bits in the bytes transmitted and received.

Stop Bits:

Specifies the number of stop bits to be used. It is one of the following values.

Value	Description
1	1 stop bit
2	2 stop bits
1.5	1.5 stop bits

Simulate I/O:

The "Simulate I/O" checkbox switches from reading I/O from the module to running a simulator. Since the simulator does not open the COM port, it is an easy way to work with the server, to configure tags or to connect clients without requiring any hardware.

Pending Time:

Minimum interval time between two access. To activate this function, NAPOPC_CE5 can work under optimized communication performance. If this module only needs to be accessed 1 time per 5 seconds. You can set pending time as 5000 ms. NAPOPC_CE5 will automatically spread time resource to other modules which are connected with each other.

OK:

Click on the "OK" button to add the new controller setting.

Cancel:

Click on the "Cancel" button to avoid any changes.

Step 5: Click on the "OK" button to add this new device.

1.2.3.7 Adding a New Modbus ASCII Controller

Step 1: Click on the "Add/ New Device..." menu item or the 🙋 icon to add a new module.

Add	Edit	View	Services
Nev	v Devic	e	Ctrl+D
Nev	v Group	o	Ctrl+G
Nev	/ Tag	i i	Ctrl+T
Search Modules Generate Tags			
Mult	i Tags		
	Fig 1	.2.3.7	7-1

Step 2: The "Select Device" window pops up.

Step 3: Click on the "Modbus" radio button.

Step 4: Click on the "Modbus ASCII" radio button.

DCON OFRnet OModbus	
Device Name Device19	
Controller Setting	
O Modbus RTU O ISaGRAF Address	1
Modbus ASCII General Modbus Device Timeout	500
Modbus TCP IP Address I92.168.255.1 Msg Delay Port 502 Word St	j0 wap
COM Port Setting	
COM Port 1 Parity None	×
Baud Rate 9500 Data Bits 8 Stop Bits 1	×
Simulate I/O	
Pending Time 0 ms	

Fig 1.2.3.7-2

Device Name:

Names with spaces or punctuation such as "|!.," cannot be used within a module name. The clients use the "Device Name" and "Tags" to access its value. The "Device Name" can not be the same as any other module.

ISaGRAF:

Connect ISaGRAF controller

General Modbus Device:

Connect general modbus device

Address:

Specifies a Address for this controller. The default value is 1 and the valid range is between 1 to 255.

Timeout:

Specifies timeout (Response time) value for this controller. A smaller timeout value may cause communication failure and a larger timeout value may reduce the performance of the client program.

Msg Delay:

Specifies message delay value for this controller. The default value is 0 ms. A smaller msg delay value may have a higher system loading, but it will have a faster data exchange speed.

Word Swap:

The "Word Swap" checkbox switches the interpretation of 4 Byte values. Sometimes we need to make the checkbox "TRUE" in order to achieve the purpose of Lo-Hi/Hi-Lo communication.

COM Port:

Specifies the COM port to be used. Please verfiy which COM port number that the RS-485 network is using. Wrong settings will always cause communication failure.

Baud Rate:

Specifies the baud rate to be used. Verify the module's current baud rate. A wrong setting will always cause communication error for this controller.

Parity:

Specifies the parity scheme to be used. It is one of the following values.

Value	Description
None	No parity
Even	Even
Odd	Odd

Data Bits:

Specifies the number of bits in the bytes transmitted and received.

Stop Bits:

Specifies the number of stop bits to be used. It is one of the following values.

Value	Description
1	1 stop bit
2	2 stop bits
1.5	1.5 stop bits

Simulate I/O:

The "Simulate I/O" checkbox switches from reading I/O from the module to running a simulator. Since the simulator does not open the COM port, it is an easy way to work with the server, to configure tags or to connect clients without requiring any hardware.

Pending Time:

Minimum interval time between two access. To activate this function, NAPOPC_CE5 can work under optimized communication performance. If this module only needs to be accessed 1 time per 5 seconds. You can set pending time as 5000 ms. NAPOPC_CE5 will automatically spread time resource to other modules which are connected with each other.

OK:

Click on the "OK" button to add the new controller setting.

Cancel:

Click on the "Cancel" button to avoid any changes.

Step 5: Click on the "OK" button to add this new device.

1.2.3.8 Adding a New Modbus TCP Controller

Step 1: Click on the "Add/ New Device..." menu item or the 🖻 icon to add a new module.

Add	Edit	View	Services	
Nev	v Devic	e	Ctrl+D	
Nev	v Grou	p	Ctrl+G	
Nev	New Tag Ctrl+		Ctrl+T	
Sea	Search Modules			
Gen	Generate Tags			
Mult	Multi Tags			

Fig 1.2.3.8-1

Step 2: The "Select Device" window pops up.

Step 3: Click on the "Modbus" radio button.

Step 4: Click on the "Modbus TCP" radio button.

Device Name Device	919	
Modbus RTU Modbus ASCII	ISaGRAF General Modbus Device IP Address 192.168.255.1 Port 502	Address 1 Timeout 500 Msg Delay 0 Word Swap
COM Port Setting COM Port 1 Baud Rate 9600	Parity Data Stop F	None
Simulate I/O		

Fig 1.2.3.8-2

Device Name:

Names with spaces or punctuation such as "|!.," cannot be used within a module name. The clients use the "Device Name" and "Tags" to access its value. The "Device Name" can not be the same as any other module.

ISaGRAF:

Connect ISaGRAF controller

General Modbus Device:

Connect general modbus device

IP Address:

The uniqe IP address of your Modbus TCP controller.

Port:

You have to set up the value with "502" for communicating with ICP DAS Modbus TCP controller

Address:

Specifies a Address for this controller. The default value is 1 and the valid range is between 1 to 255.

Timeout:

Specifies timeout (Response time) value for this controller. A smaller timeout value may cause communication failure and a larger timeout value may reduce the performance of the client program.

Msg Delay:

Specifies message delay value for this controller. The default value is 0 ms. A smaller msg delay value may have a higher system loading, but it will have a faster data exchange speed.

Word Swap:

The "Word Swap" checkbox switches the interpretation of 4 Byte values. Sometimes we need to make the checkbox "TRUE" in order to achieve the purpose of Lo-Hi/Hi-Lo communication.

Simulate I/O:

The "Simulate I/O" checkbox switches from reading I/O from the module to running a simulator. Since the simulator does not open the COM port, it is an easy way to work with the server, to configure tags or to connect clients without requiring any hardware.

Pending Time:

Minimum interval time between two access. To activate this function, NAPOPC_CE5 can work under optimized communication performance. If this module only needs to be accessed 1 time per 5 seconds. You can set pending time as 5000 ms. NAPOPC_CE5 will automatically spread time resource to other modules which are connected with each other.

OK:

Click on the "OK" button to add the new controller setting.

Cancel:

Click on the "Cancel" button to avoid any changes.

1.2.4 Adding a New Group

Step 1: Click on the "Add/ New Group" menu item or the 🗟 icon to add a new group.

Step 2: The "Group" window pops up.

roup	
Name Group	ОК
	Cancel

Fig 1.2.4-1

Name:

A "Group Name" may have any name, but avoid names with spaces or punctuation such as "|!.,". The "Group Name" must not be used twice. A group can be defined as a subdirectory containing one or more tags. A device may have many subgroups of tags. All tags belong to their module when they are scanned to perform I/O.

1.2.5 Adding a New Tag

1.2.5.1 Adding New Tags For I-7K/8K/87K/ZigBee/FRnet/XW Module

Step 1: Click on the "Add/Generate Tags" menu item to add new tags.



- Step 2: "Generate Tags" function will generate tags for the device you choose.
- **Step 3**: Double click the tag to check the property.
- **Step 4**: Choice the "Settings" page. Because the tag belongs to the module-type device, the "I/O Module" radio button is active.

Tag Properties				ОК 🔀
Settings Read & Wr	ite			1.*
Name Tag1	Modbus address	17	Output Register	<u> </u>
Description		16		_
Device Type				
O Internal Device				
💿 I/O Module	Тур	e Anal	og Input	2
	Channe		_	
Controller	Locatio	n 1	Output Register	3
	Data	Bool	2	<u>.</u>
Scaling				
Enable				
Jocangstit				
	2			10000
Simulation Sin	e			×

Fig 1.2.5.1-2

Name:

Any "Tag Name" may be used, but avoid names with spaces or punctuation such as "|!.,". The clients will use the "Device Name" and "Tags" to access its value. Hence the "Tag Name" cannot be a duplicate of another tag in the same group.

Modbus address:

Specifies an unique modbus address for this tag in order to communicate with modbus client. The default address is already an unique one.

After that, you also need to choose the address type. There are four address types you can choose. They are "Input Coil", "Output Coil", "Input Register", and "Output Register" which depends on your tag property. It is important to give an appropriate modbus address type and address value.

Address Type	Range
Output Coil	000001 - 001000
Input Coil	100001 - 101000
Input Register	300001 - 301000
Output Register	400001 - 401000

Description:

Specifies the description text for this tag. This can be blank.

Type:

Shows the command to be used for this tag. Different modules support different commands.

Channel:

Specifies the channel number to be used for this tag. The "Digital Input" and "Digital Output" tags do not use this channel setting, because all channels are read with one communication.

Simulation:

The valid signal is SINE, RAMP and RANDOM. This field is validated when the module uses simulation I/O. Please refer to the "Adding A New Device" section.

OK:

Click on the "OK" button to add the new tag setting.

Cancel:

Click on the "Cancel" button to avoid any changes.

Scaling:

Enable:

Check this check-box to enable the "Settings..." button.

Settings:

Click on this button to set the scaling feature.

For more information, please refer to the section "1.2.5.4 Scaling Settings".

1.2.5.2 Adding a New Tag For Internal Device

Step 1: Click on the "Add/ New Tag" menu item or the sicon to add a new tag.

Step 2: The "Tag Properties" window pops up.

Step 3: Choice the "Settings" page. Because the tag belongs to the internal-type device, the "Internal Device" radio button is active.

Tag Properties					0К 🔀
Settings Read & Wr	ite				
Name Tagi	Modbus address	23	Output Regi	ster 🔽	
Description					
Device Type					
Internal Device					
I/O Module	Тур	e Ana	log Input	<u>.</u>	
	Channe	el 🔽			
O Controller	Locatio	n 1	Output Regi	ster 😾	
	Dati	Bool	1	~	
Scaling Enable Settings					
Simulation Sin	e			V	

Fig 1.2.5.2-1

Name:

Any "Tag Name" may be used, but avoid names with spaces or punctuation such as "|!.,". The clients will use the "Device Name" and "Tags" to access its value. Hence the "Tag Name" cannot be a duplicate of another tag in the same group.

Modbus address:

Specifies an unique modbus address for this tag in order to communicate with modbus client. The default address is already an unique one.

After that, you also need to choose the address type. There are four address types you can choose. They are "Input Coil", "Output Coil", "Input Register", and "Output Register" which depends on your tag property. It is important to give an appropriate modbus address type and address value.

Address Type	Range
Output Coil	001001 - 020999
Input Coil	101001 - 120999
Input Register	301001 - 320999
Output Register	401001 - 420999

Description:

Specifies the description text for this tag. This can be blank.

1.2.5.3 Adding a New Tag For Modbus Device

- Step 1: Click on the "Add/ New Tag" menu item or the 🔳 icon to add a new tag.
- **Step 2**: The "Tag Properties" window pops up.
- **Step 3**: Choice the "Settings" page. Because the tag belongs to the controller-type device, the "Controller" radio button is active.

Name Tag1 Modbu:	s address 23	Output Register
, in the second	120	Josephericegister
scription		
vice Type		
) Internal Device		
I/O Module	Type Anal	og Ipput
2		
	Channel JU	
Controller	Location 1	Output Register 🔽
	Data Short	
	Short	
aling	Long	
] Enable	Float	~
Settings	India	

Fig 1.2.5.3-1

Name:

Any "Tag Name" may be used, but avoid names with spaces or punctuation such as "[!.,". The clients will use the "Device Name" and "Tags" to access its value. Hence the "Tag Name" cannot be a duplicate of another tag in the same group.

Modbus address:

Specifies an unique modbus address for this tag in order to communicate with modbus client. The default address is already an unique one.

After that, you also need to choose the address type. There are four address types you can choose. They are "Input Coil", "Output Coil", "Input Register", and "Output Register" which depends on your tag property. It is important to give an appropriate modbus address type and address value.

Address Type	Range
Output Coil	000001 - 001000
Input Coil	100001 - 101000
Input Register	300001 - 301000
Output Register	400001 - 401000

Description:

Specifies the description text for this tag. This can be blank.

Location:

Specifies the tag address. It must be the same with the the variable address in the controller. Besides, you have to choice the location type. After you choice the location number, there are four location types you can choice. They are "Input Coil", "Output Coil", "Input Register", and "Output Register". When you monitor controller device(see 1.2.2 Monitoring Device), the "Channel/Location" field will show a value according to the location and location type as belows.

Location Type	Range
Output Coil	000001 - 065536
Input Coil	100001 - 165536
Input Register	300001 - 365536
Output Register	400001 - 465536

Data:

Specifies the data type of this tag which's location type is "Input Register" or "Output Register". NAPOPC_CE5 supports five kinds of data type which are "Short", "Long", "Float", "Word", and "DWord".

Data Type	Definition	Range
Short	16-bit signed integer	-32768~32767
Long	32-bit signed integer	-2147483648~2147483647
Float	Floating-point variable	-1.7E-308~1.7E+308
Word	16-bit unsigned integer	0~65535
DWord	32-bit unsigned integer	0~4294967295

The data type of "Input Coil" or "Output Coil" is "Bool".

Simulation:

The valid signal is SINE, RAMP and RANDOM. This field is validated when the module uses simulation I/O. Please refer to the "Adding A New Device" section.

OK:

Click on the "OK" button to add the new tag setting.

Cancel:

Click on the "Cancel" button to avoid any changes.

Scaling:

Enable:

Check this check-box to enable the "Settings..." button.

Settings:

Click on this button to set the scaling feature.

For more information, please refer to the section "1.2.5.4 Scaling Settings".

1.2.5.4 Scaling Settings

In general, the "Scaling" feature is only useful for the floating-point data type.

Tag Pro	ucess Settings	🥐 🕅
Scaling		
Min Max	Raw Units Scales to Min 0 1000 Max 10	
<u>D</u> eadba	Conversion Linear Square Root and: 0 %	

Fig 1.2.5.4-1

Raw Data:

Min: The original Minimum value. ([MinRaw]) Max: The original Maximum value. ([MaxRaw])

Scales to:

Units: The unit of the scaled value. (Just for reference only.) Min: The scaled Minimum value. ([MinScale]) Max: The scaled Maximum value. ([MaxScale])

Conversion:

Linear:

Scaled Value = ((Original Value – [MinRaw]) / ([MaxRaw] – [MinRaw])) * ([MaxScale] – [MinScale]) + [MinScale]

Square Root:

```
Scaled Value = ((sqrt(Original Value) – [MinRaw]) * ([MaxScale] – [MinScale]))
/ sqrt([MaxRaw] – [MinRaw]) + [MinScale]
```

Deadband(%):

In general, keep "0" in this field. Deadband will only apply to items in the group that have a dwEUType of Analog available. If the dwEUType is Analog, then the EU Low and EU High values for the item can be used to calculate the range for the item. This range will be multiplied with the Deadband to generate an exception limit. An exception is determined as follows:

Exception if (absolute value of (last cached value - current value) > PercentDeadband * (EU High –EU Low))

OK:

Click the "OK" button to save these settings.

Cancel:

Click the "Cancel" button to avoid any changes.

1.2.6 Adding Multi Tags for Modbus Device

This function only work when the device's protocol is Modbus.

Step 1: Click on the "Add/ Multi Tags" menu item

			-
Add	Edit	View	Services
Nev	v Devic	e	Ctrl+D
Nev	v Group	o	Ctrl+G
Nev	v Tag		Ctrl+T
Sea	rch Mo	dules	
Ger	nerate i	Tags	-
Mult	ti Tags		
	Fia	1.2.6	-1

Step 2: The "Add Multi Tags Dialog" dialog box pops up.

Step 3: Choose correct "Prototype", "Data Type" and key in Modbus address.

Coil Input	Coil Output	O Register Input	Register Output
)ata Type		Office Office	
(•) BOOL	Short ()Long	Ground Wo	rd () Dword
vioadus Address		Separation	

Fig 1.2.6 -2

Prototype:

There are four kinds of prototype for modbus tag. "Coil Input", "Coil Output", "Register Input" and "Register Output".

Data Type: "Bool": 8 bits, True or False "Short" : 16 bits, -32768 ~ 32767 "Long": 32 bits, -2147483648. ~ 2147483647 "Float": 32 bits, float numbers "Word": 16 bits, 0 ~ 65535 "DWORD" : 32 bits, 0 ~ 4294967295 Modbus Address: "From" : modbus address number of start tag, 1 ~ 65535 "To" : modbus address number of end tag. 1 ~ 65535 Separation: Separation numbers between each tag. 1 ~ 100 OK: Click on the "OK" button to add the new tag setting. Cancel:

Click on the "Cancel" button to avoid any changes.

1.2.7 Read/Write the Tags

First, you have to use the "Monitor" function to see values of tags by checking the "View/ Monitor" menu item. Select a tag and right click the mouse button. Then select the "Properties.." option. Choose the "Read & Write" page to read/write the tag.

- **Step 1**: Click the "View/ Monitor" menu item to enable monitor.
- Step 2: Select a tag and right click the mouse button. Then select the "Properties.." option.
- Step 3: Choose the "Read & Write" page. You can see the "Tag name" and "Access right" at the first. If the access right is "Read only!", the write function is disable.

Tag Properties	ОК 🔽
Settings Read & Write	
Read Value Value: Quality: Timestamp:	
Tag name: Ch04 Access right: Read&Write! Write Value Timestamp: Quality	
Value:	
Fig 1.2.7-1	

Ver: 2.30 Date: Aug-05 2011
Read Value/Value:

You can press the "Read!" button to read the tag value as you saw on the "Tag-Window".

Read Value/Quality:

Three kinds of qualities, "Good", "Bad", and "Uncertain", would be shown. If the communication status is good, the quality shows "Good". If the communication status has something wrong, the shows "Bad". And the other situation is "Uncertain".

Read Value/Timestamp:

It shows the time, when you read the tag.

Tag name:

It is the same with the "Name" at the "Settings" page. You can modify it at the "Settings" page.

Access right:

There are two kinds of access rights, "Read Only!" and "Read&Write!". The access right depends on what kind of tag property it is. Please refer to the "1.6 Adding A New Tag"

Write Value/Timestamp:

It shows the time that the tag is written.

Write Value/Quality:

Three kinds of qualities, "Good", "Bad", and "Uncertain", would be shown. If the communication status is good, the quality shows "Good". If the communication status has something wrong, the shows "Bad". And the other situation is "Uncertain".

Write Value/Value:

You can press the "Write!" button to write the value you key-in to the tag. If the tag data type is "Boolean" the write value "0" means "OFF" and the write value "not 0" means "ON".

1.2.8 Editing A Device/Group/Tag properties

To edit an existing Device or Group, just select the Device or Group and then select the "Properties..." option.



To edit an existing Tag, just select the Tag and right click mouse button to select "Properties..." option.



1.2.9 Deleting A Device/Group/Tag

To delete an existing Device or Group, just select the Device or Group and then select the "Delete..." option.



Fig 1.2.9-1

To delete an existing Tag, just select the Tag and right click mouse button to select "Delete..." option.



1.2.10 Generating Tags

This function lets you easily test the NAPOPC_CE5 in the simulation mode. It is only valid if the selected device of module type has no sub "Module", "Group" and "Tag".

Step 1: Select a device of module type you want to generate tags.Step 2: Click on the "Add/ Generate Tags" menu item.



Tags are generated depending on the Module-ID. Possible tags are "Analog Input", "Analog Output", "Digital Input", "Digital Output", "Latched DI" and "Counter".

1.2.11 Services Setup

This function lets you define which services you want to active for exchanging data with the other programs. NAPOPC_CE5 provides "RPC Server", "Modbus RTU", "Modbus ASCII", "Modbus TCP", and "Active ScanKernel" four services to be choosed. In them, the "RPC Server" is a mechanism which allows NAPOPC_ST/NAPOPC_XPE DA Server use NAPOPC_CE5 via "Remote Procedure Call". If you wanna create a "RPC" device at NAPOPC_ST/NAPOPC_XPE site, please check this at NAPOPC_CE5 site. "Modbus RTU", "Modbus ASCII", and "Modbus TCP" services would active immediately by checking. The "Active ScanKernel" service should check at all situation except to be the intermediary program between user application programs.

Step 1: Click on the "Services/Setup" menu item.

RPC Ser	ver 505 canKernel er 1		Port number	гСР r <mark>502</mark>	_
Slave1	ModbusASC		Slave2	1odbusRTU	
	IodbusRTU IodbusASC	II	COM port	СОМЗ	~
Baudrate	9600	~	Baudrate	9600	-
Parity	None	~	Parity	None	~
Data Bits	8 (RTU)	-	Data Bits	8 (RTU)	~
Stop Bits	1	~	Stop Bits	1	~

Fig 1.2.11-1

Step 2: Choose the services you want.

RPC Server Port:

You have to set up the value with "505" for communicating with NAPOPC_ST or NAPOPC_XPE.

COM Port:

Specifies the COM port to be used. Please verfiy which COM port number that the RS-485 network is using. Wrong settings will always cause communication failure.

Baudrate:

Specifies the baud rate to be used. Verify the module's current baud rate. A wrong setting will always cause communication error for this controller.

Parity:

Specifies the parity scheme to be used. It is one of the following values.

Value	Description
None	No parity
Even	Even
Odd	Odd

Data Bits:

Specifies the number of bits in the bytes transmitted and received.

Stop Bits:

Specifies the number of stop bits to be used. It is one of the following values.

Value	Description
1	1 stop bit
2	2 stop bits
1.5	1.5 stop bits

1.2.12 Rule Script Editor

This function lets you design your rule base for making your WinPAC-8000/5000 to be a DCS via NAPOPC_CE5. The description of rule base of NAPOPC_CE5 is like "IF...THEN...". The left upper corner in the "Rule Script Editor" has four conditions behind "IF" in which the variables are showed as modbus address and combined with "AND/OR" each other. The right upper corner in the "Rule Script Editor" has four outputs behind "THEN" in which the variables are showed as modbus address and combined with "AND" each other. The relation between timer value and other variables is "AND".

If the variable behind "IF" is "0xxxxx" or "1xxxxx", the "Status" would be "0" or "1". The value "0" means "OFF" and the value "1" means "ON". If the variable is "3xxxxx" or "4xxxxx", the "Status" would depend on the data type of variable.

F: Timer 30	0 ms		^{TT}][HEN:	Assign Outp	out Logic AN	id/or	Add
Variable	Judge Sta	atus Logic Af	VD/OR	00002 💌	= 0	AND		Delete
None 💌	0 == 0			00003 🔽	= 0	AND		Edit
None 💌	>= ▼ 0 <= ▼ 0			00005 💌	= 0	AND	•	
None 💌			[[00006 💌] = 0			
Active Rule	Script (Hint: D	O:0xxxxx DI	:1xxxx A	[:3xxxxx A	D:4xxxxx)			_
		eeeeeb I III	decessed		Fer: 10			Save
IF THEN (000002' = '0.0 000002' = '0.0	100000") 'and' 100000") 'and'	(000003' = (000003' =	'0.000000')	[i imer = '3 'and' ('0000	00 JLI 105' = '0.00(), ('0000	Cancel
								_

Fig 1.2.12-1

Add:

Press this button to the "Rule list" after editing each rule.

Delete:

Check the rules in the "Rule list", and then press this button to delete.

Edit:

Click the rule in the "Rule list" to edit, and after that press this button to update. Save:

Save the "Rule list" to be "Rule.txt" after finishing editing.

Cancel:

Leave this editor.

Active Rule Script:

It would be active immediately after checking this option. If you wish to act the "Rule script" after rebooting NAPOPC_CE5, you should save file with "File/Save".

1.2.13 File

This function lets you save and load the configurations of NAPOPC_CE5. For taking the correct configuration file of NAPOPC_CE5 "*.tdb" after rebooting the WinPAC-8000/5000, you not only use "File/Save" to save in the NAPOPC_CE5 but also need to "Save and Reboot" in the "WinPAC Utility". Moreover, NAPOPC_CE5 will automatically load the last configuration file with every launch.

<u>File</u>	<u>A</u> dd	<u>E</u> dit	View	Services
Nev	v			Ctrl+N
Ope	en			Ctrl+O
Sav	ve			Ctrl+S
Sav	e <u>A</u> s			
<u>1</u> \9	iystem_	Disk\	Test1.to	lb
<u>2</u> g	gg.tdb			
<u>3</u> /S	iystem_	Disk\tes	st.tdb	
E <u>x</u> it				

New:

Clean current project and create a new project

<u>File</u>	<u>A</u> dd	<u>E</u> dit	View	Services
(<u>N</u> ev	V			Ctrl+N
Ope	en			Ctrl+O
Sav	е			Ctrl+S
Sav	e <u>A</u> s			
119	iystem_	Disk\'	Test1.to	lb
2 g	gg.tdb			
319	ystem_	Disk\te:	st.tdb	
E <u>x</u> it				-

Open: Load old NAPOPC_CE5 project

File	Add	Edit	View	<u>Services</u>
Nev	٧			Ctrl+N
Ope	en			Ctrl+O)
Sav	е			Ctrl+S
Sav	e <u>A</u> s			
119	ystem_	Disk\\	Test1.to	tb
2 g	gg.tdb			
319	ystem_	Disk\te:	st.tdb	
E <u>x</u> it				

Save:

Save current NAPOPC_CE5 project

<u>File</u>	<u>A</u> dd	<u>E</u> dit	View	Services
<u>N</u> e	Ŵ.			Ctrl+N
Ope	en			Ctrl+O
Sav	е			Ctrl+S)
Sav	e <u>A</u> s			
19	ystem_	Disk\'	Test1.to	lb
2g	gg.tdb			
319	System_	Disk\te:	st.tdb	
E <u>x</u> it				

Save as...:

Save NAPOPC_CE5 project as a new one

<u>File</u>	Add	<u>E</u> dit	View	Services
Nev	v			Ctrl+N
Ope	en			Ctrl+O
Sav	е			Ctrl+S
Sav	e <u>A</u> s			
119	ystem_	Disk\	Test1.to	tb
2 g	gg.tdb			
319	ystem_	Disk\tes	st.tdb	
E <u>x</u> it				

1.2.14 About

Click on the "Help/ About NAPOPC_CE5" menu item to see the "About NAPOPC_CE5" window. It shows the version number.

Step 1: Click on the "Help/ About NAPOPC_CE5" menu item.

Step 2: The "About NAPOPC_CE5" window pops up.

About	NAPOPC_CE5	
	NAPOPC_CE5	OK
	Version 2.30	
	Update : Jul-15 2011	
	Copyright 2011. All Rights Reserved	
	Fig 1.2.14-1	

1.2.15 Minimize NAPOPC_CE5

If you want to minimize NAPOPC_CE5, please click - on the top-right corner.

NAPO	PC_CE	5													\mathbf{X}
<u>F</u> ile	<u>A</u> dd	<u>E</u> dit	<u>V</u> iew	<u>S</u> ervices	<u>R</u> ule	Help	& 1	۶ 8	2	X [8	Ж	Pa 🛍		
							Fig	1.:	2.1	15-1					

After clicking the question mark, NAPOPC_CE5 will minimize itself at the status bar. It will be restored by double clicking it.



2 Quick Start

Please follow these steps:

- Wiring Modules or Controllers. Wiring modules in the RS-485 network. Wiring controllers to WinPAC-8000/5000 (Refer to winpac8000_user_manual_v2.0.2.pdf)
- Configuring Modules or Controllers. Using the DCON Utility to set modules. (Refer to winpac8000_user_manual_v2.0.2.pdf)
- Running NAPOPC_CE5 Launch NAPOPC_CE5 by means of executing the "NAPOPCSvr_CE5.exe" or "NAPCOP_CE5Boot.exe"
- 4. Searching Modules. Refer to the "1.2.1 Searching Modules.." section to search modules.
- Adding a new controller Refer to the "1.2.3 Adding A New Device" section to add a new modbus RTU or modbus TCP controller.
- 6. Saving Configuration. Refer to the "1.2.13 File Save" section to save the configuration.
- 7. Closing NAPOPC_CE5. Close NAPOPC_CE5 by clicking the "File/Exit" menu item.

NOTE: After above steps, please use "Save and Reboot" function at WinPAC Utility to save registry exactly.

3 Remote Accessing

OPC Client has two ways to access the OPC Server. One is called "Local Accessing", and the other is called "Remote Accessing". If the OPC Client and the OPC Server are at the same computer, we said this kind of architecture is "Local Accessing". In other words, if the OPC Client should access OPC Server through a network, we said this kind of architecture is "Remote Accessing".

The following figure shows the integrated architecture including "Local Accessing" and "Remote Accessing". At the real Process Industry, the two ways are often used at the same time. At the Process Management Layer, we often use "Local Accessing" architecture to monitor and control manufacturing processes. At the Business Management Layer, we just set up the OPC Client to collect the process information from the Process Management Layer. If you just want to construct the "Local Accessing" architecture, you do not need to read this chapter. If you want to construct the "Remote Accessing" architecture, you have to know how to set up the DCOM between OPC Client and OPC Server.



Figure 3-0-1 Local access and Remote access architecture.

3.1 System Requirement

To access a remote OPC server over a network, it is required to enable the DCOM mechanism on both stations, where the client and server are resided.

It is not possible to launch a secure process on a Windows 95 computer from a client computer. All processes in Windows 95 run in the security context of the currently logged-on user; therefore, DCOM on Windows 95 does not support remote activation. A server application on a Windows 95 computer will have to be launched manually or by some other mechanism to be accessed by a client application on another computer. Consequently, the "DefaultLaunchPermissions" and "LaunchPermissions" registry values have no affect on Windows 95.

Platform	Does the platform support the DCOM?
Windows 95	No. Users need to download and install the DCOM95.EXE and DCM95CFG.EXE from Microsoft's web site to enable the remote access.
Windows 98	Yes. Windows 98 supports the DCOM mechanism. It is recommended to upgrade to the newest version of DCOM98. The newest DCOM98 is also available at Microsoft's web site.
Windows NT 4.0	Yes. Windows NT 4.0 supports the DCOM mechanism. It is recommended to upgrade to the newest Service Pack for Windows NT 4.0 (Service Pack 3 or newer one).
Windows 2000	Yes. Windows 2000 supports the DCOM mechanism.
Windows XP	Yes. Windows XP supports the DCOM mechanism.

3.2 Configuring DCOM

Before making changes, register the server application in the registry of both the client and server computers. This may involve either running the server applications setup program or running the server application, then shutting it down on both computers. The server application does not need to reside on the client computer.

If the server uses custom interfaces, the marshaling code must be installed on the client and server computers. Automation servers that support "vtbl-binding" must install their type libraries on the client and server computers. Automation servers that do not support "vtbl-binding" do not need to install their type libraries on the client computer.

After changing the registry, run the client application on the client computer. The DCOM looks at the server application registry entries on the client computer and determines the name of the server computer. It will then connects to the server computer, use the server computer registry to determine the location of the server application, and start the server application on that computer.

You can change the registry with the DCOMCnfg.exe tool, the OLE Viewer tool, or manually. For more information on using OLE Viewer or manual changes, please refer to the "Q158582, HOWTO: Configure a Non-DCOM Server and Client to Use DCOM" article on Microsoft's web site. For more information on using DCOMCnfg.exe to configure the DCOM, please refer to "Inside Distributed COM", written by Guy Eddon and Henry Eddon in 1998 for Microsoft Press.

This section shows you how to configure the DCOM status with DCOMCnfg.exe graphic-driven utility (can be found in the Windows NT system32 folder or in the Windows95/98 system folder) on the client and server computer.

The following table shows three combinations of DCOM settings related to WinPAC. You can see WinPAC can be client site and server site with itself, but WinPAC only can be server site against XPAC and PC. The limitation is due to DCOM security. We only choose Windows XP for example to set up DCOM because there are too many kinds of OS on PC. You can use other Microsoft desktop operation system on our PC.

Client Site	Server Site
PC(NAPOPC_ST Server)	WinPAC(NAPOPC_CE5 Server)
XPAC(NAPOPC_XPE Server)	WinPAC(NAPOPC_CE5 Server)
WinPAC(NAPOPC_CE5 Server)	WinPAC(NAPOPC_CE5 Server)

3.2.1 Configuring On the Server Site (WinPAC) System Requirement

OS version: WinPAC OS 1.3.04 or later Program: NAPOPC_CE5 DCOMCnfg.exe WinPAC Utility 2.0.2.1 or later

Configuring DCOM

Step 1: Run the \\NAPOPC_CE5\napopc_ce5boot.exe program to register.

Step 2: Run the dcomcnfg.exe program and choose "Default".



Step 3: Select the "Access" button to add an account which is current connection account from client site.

ОК
Cancel
Add
Delete
Сок

Step 4: Select the "Launch" button to add an account which is current connection account from client site as above.

Step 5: Execute "WinPAC Utility->Network Setting->Users and Password"

WinPAC Utility [2.0.2.1]
File Help Configuration
System Setting Ethernet Setting Network Setting System Information Au
FTP Setting Users and Password
User name Password
Test **** Add Delete
User name Password Note: The accounts is used to login the servers search as Telnet, FTP, WebServer etc on WinPAC.
Setting

Step 6: Fill out "User name", "Password", and press "Add". The "User name" and "Password" must be the account we set at **Step 3.** After pressing "Add", press "Setting" to finish all settings.

Step 7: Run WinPAC Utility to save and reboot.

3.2.2 Configuring On the Client Site (PC)

Configuring the Firewall

Step1: By default the windows firewall is set to "On". This setting is recommended by Microsoft and by OPC to give your machine the highest possible protection. For trouble shooting, you may wish to temporarily turn off the firewall to prove or disprove that the firewall configuration is the source of any communication failure. **Note:** It may be appropriate to permanently turn off the firewall if the machine is sufficiently protected behind a corporate firewall. When turned off, the individual firewall settings outlined here need not be performed to allow OPC communication.



Step 2: Select the .Exceptions tab and add all OPC Clients and Servers to the exception list. Also add Microsoft Management Console (used by the DCOM configuration utility in the next section) and the OPC utility OPCEnum.exe found in the Windows\System32 directory.

Windows Firewa	u
Lieneral Exceptions	Advanced
Windows Firewall is b programs and service to work better but mig	locking incoming network connections, except for the s selected below. Adding exceptions allows some programs iht increase your security risk.
Programs and Service	35:
File and Printer 9	ibaring
GenAgent.exe	
GenRegistrarSe	(ver.exe
☑ GraphWorX32	
LASEngine.exe	
License Monitor	
Microsoft Manag	jement Console 🧮
🗹 OPC DataSpy	
OPC Simulator	
🛛 🗹 Remote Assistar	ice
	۰ 💌
Add Program	Add Port Edit Delete
<u></u>	
Display a potificati	ion when Windows Firewall blocks a program
E biopidy a Hermode	
What are the risks of	allowing exceptions?
	OK Cancel

In the Add a Program dialog, there is a listing of most applications on the machine, but note that not all of them show up on this list. Use the "Browse" button to find other executables installed on the computer.

Note: Only EXE files are added to the exceptions list. For in-process OPC Servers

and Clients (DLLs and OCXs) you will need to add the EXE applications that call them to the list instead.

S Lond London London London	logger Configurator	^
<mark>2</mark> Alarm 9	jerver Corfigurator	
🛂 Alarm V	/orX32	
Carous 🖬	el	
🔀 DataW	orX32	
DBOP	CServerConfigurator	
DrDCO	м	
S FreeCe	H. Contraction of the second se	
🛐 GenDC	IS3 to GFW16	
🔁 GenDC	IS4 to GFW16	
🖾 GenSta	atistics Viewer	~
	12	
GenDU GenSta	IS4 to GFW16 stistics Viewer	

Step 3: Add TCP port 135 as it is needed to initiate DCOM communications, and allow for incoming echo requests. In the Exceptions tab of the Windows Firewall, click on Add Port.

Windo	ows Firewa	ill				Σ
General	Exceptions	Advanced				
Windov program to work Program	vs Firewall is b ns and service better but mig ns and Servic	lockirg incon s selected be ght increase y es:	ning netwo Ilow. Addin our securit <u>i</u>	rk connecti g exceptior y risk.	ons, except fo ns allows some	r the programs
Name	e	1963				~
🗆 File	e and Printer !	Sharing				
⊡ Ge	enAgent.exe					
🗹 Ge	en RegistrarSe	rver.exe				
🗹 Gr	aphWorX32					
🗹 LA	SEngine.exe					
🗹 Lie	ense Monitor					
🗹 Mi	crosoft Manaj	gemert Conso	ole			-
🗹 OF	PC DataSpy					
⊡ OF	PC Simulator					
🗹 Re	emote Assista	nce				
_ Be	emote Deskto	n	10			
Add	Program	Add Port		Edit	D	elete
Disc	lau a notificat	ion when Win	idows Fireu	vall blocks	a program	
	ay a <u>n</u> otineat	Sen Yorsen YY I		an bioons i	a program	
<u>What a</u>	re the risks of	allowing exce	eptions?			
					ОК	Cancel

In the Add a Port dialog, fill out the fields as follows: Name: DCOM Port number: 135

Choose the TCP radio button

Add a Port	
Use these settings number and protoc want to use.	to open a port through Windows Firewall. To find the port ol, consult the documentation for the program or service you
<u>N</u> ame:	ОСОМ
Port number:	135
What are the risks	of cpening a port?
Change scope	OK Cancel

Creating the Account





Configuring DCOM

Step 1: Run the dcomcnfg.exe program to launch component services. Right clieck "My Computer" and choose "Properties".



- Step 2: Select the "Default Properties" tab page.
- Step 3: Use the following settings:

Field Name	Set to
Enable Distributed COM on this computer	Checked
Default Authentication Level:	Default
Default Impersonation Level:	Identify

My Computer Properties					
Default Protocols MSDTC COM Security General Options Default Properties					
Enable Distributed COM on this computer Enable COM Internet Services on this computer					
Default Distributed COM Communication Properties The Authentication Level specifies security at the packet level					
Default Authentication Level:					
The impersonation level specifies whether applications can determine who is calling them, and whether the application can do operations using the client's identity.					
Default Impersonation Level: Identify					
Security for reference tracking can be provided if authentication is used and that the default impersonation level is not anonymous.					
Provide additional security for reference tracking					

General	Options	Default Properties
Default Protocols	MSDTC	COM Security
ccess Permissions —		
You may edit who is also set limits on app	allowed default access dications that determine	to applications. You ma their own permissions.
	Edit Limits	Edit Default
determine their		
determine their own	Edit Limits	Edit Default
determine their own	Edit Limits	Edit Default
determine their own	Edit Limits	Edit Default
determine their own	Edit Limits	Edit Default

Step 4: Select the <u>"COM Security</u>" tab page.

Step 5: Click on the "Edit Limits..." of "Access Permissions" button to set.

roup or user names:		
😰 E veryone		
ermissions for ANDNYMDUS	Add	Remove Deny
Local Access Remote Access		
Hemote Access		

	? 🛛
Add	Remove
Allow	Deny
	-
ОК	Cancel
	Add Allow

Step 6: Click on the "Edit Default..." of "Access Permissions" button to set.

oup or user names:		
SELF SYSTEM		
ermissions for Everyone	Add Allow	Remove Deny
Local Access Remote Access		

Step 7: Click on the "Edit Limits..." of "Launch and Activation Permissions" button to set.

unch Permission		?
Group or user names:		
🕵 Administrators (ZIBET \Ad	dministrators)	
🕵 Everyone		
Permissions for Externione	Add	Remove
Bemote Launch		
Local Activation		
Remote Activation	~	

Step 8: Click on the "Edit Default..." of "Launch and Activation Permissions" button to set.

Launch Permission		? 🛛
Default Security		
Group or user names:		
Administrators (ZIBET\Adn	ninistrators)	
😥 Everyone		
SYSTEM		
	Add	Remove
Permissions for Everyone	Allow	Deny
Localtaunch		
Remote Launch		
Local Activation		님기
Temole Activation		
	ОК	Cancel

Step 9: Right click on the "NAPOPC DA Server" of "DCOM Config" button and select "Properties".



Step 10: Select the "Location" tab page and check "Run application on the following computer". And enter the Server IP here.

APOPC_XPE DA Server Properties	? >
General Location Security Endpoints Identity	
The following settings allow DCOM to locate the correct comp application. If you make more than one selection, then DCOM applicable one. Client applications may overide your selection:	uter for this uses the first s.
 Run application on the computer where the data is located Run application on this computer. 	d.
Run application on the following computer:	
192 168 1 91 Br	

Step 11: Select the "Identity" tab page and check "The launching user"

NAPOPC_ST DA Server	Properties	2 🛛
General Location Secur	ity Endpoints Identity	
Which user account do yo	ou want to use to run this appli	cation?
C		
I he interactive user.		
The launching user.		
C This user.	•	
User:		Browse
Password:		
Confirm password:		
C The system account (s	ervices only).	
	OK Cance	el Apply

Step 12: Restart PC

3.2.3 Configuring On the Client Site (XPAC)

Configuring the Firewall

Step1: By default the windows firewall is set to "On". This setting is recommended by Microsoft and by OPC to give your machine the highest possible protection. For trouble shooting, you may wish to temporarily turn off the firewall to prove or disprove that the firewall configuration is the source of any communication failure.

Note: It may be appropriate to permanently turn off the firewall if the machine is sufficiently protected behind a corporate firewall. When turned off, the individual firewall settings outlined here need not be performed to allow OPC communication.



Step 2: Select the .Exceptions tab and add all OPC Clients and Servers to the exception list. Also add Microsoft Management Console (used by the DCOM configuration utility in the next section) and the OPC utility OPCEnum.exe found in the Windows\System32 directory.

Windo	ows Firewa	all	
General	Exceptions	Advanced	
Windov program to work	vs Firewall is b ns and service better but mig	olocking incoming network connections, except for the es selected below. Adding exceptions allows some program: ght increase your security risk.	8
Name	e and servic	cs.	
D File	e and Printer :	Sharing	1
I Ge	enAgent.exe		
⊡Ge	en RegistrarSe	rver.exe	
🗹 Gr	aphWorX32		
🗹 LA	SEngine.exe		
🗹 Lie	ense Monitor		
🗹 Mi	crosoft Mana	gement Console	-
OF 🗹 OF	PC DataSpy		
🗹 OF	PC Simulator		
🗹 Re	emote Assista	nce	
□ Be	emote Neskto	n	
Add	Program.	Add Port Edit Delete	٦
U	السينية		
Dien	lau a notificat	ion when Windows Firewall blocks a program	
C Disp	лау а достоа	ant when windows the war blocks a plogram	
What a	re the risks of	allowing exceptions?	
			_
			J

In the Add a Program dialog, there is a listing of most applications on the machine, but note that not all of them show up on this list. Use the "Browse" button to find other executables installed on the computer.

Note: Only EXE files are added to the exceptions list. For in-process OPC Servers and Clients (DLLs and OCXs) you will need to add the EXE applications that call them to the list instead.



Step 3: Add TCP port 135 as it is needed to initiate DCOM communications, and allow for incoming echo requests. In the Exceptions tab of the Windows Firewall, click on Add Port.

Windows programs to work be Programs	Firewall is b and service atter but mig and Servic	blocking incomin is selected below ght increase you es:	g networ w. Adding r security	k connectior g exceptions risk.	ns, except for t allows some p	he rograms
Name						~
🗆 File a	nd Printer :	Sharing				
🗹 Gen4	Agent.exe					
🗹 GenF	RegistrarSe	rver.exe				
🗹 Grap	hWorX32					
✓ LASE	ngine.exe					
🗹 Licer	nse Monitor					
Micro	soft Mana	gemert Console				-
OPC 🗹	DataSpy					
OPC 🗹	Simulator					
🗹 Rem	ote Assista	nce				
	nte Deskto	n				~
Add Pro	ogram	Add Port		<u>E</u> dit	Dek	ete
🗹 Display) a <u>n</u> otificat	ion when Windo	ws Firew	all blocks a	program	
			and the second			

In the Add a Port dialog, fill out the fields as follows: **Name: DCOM Port number: 135** Choose the TCP radio button

want to use.	
<u>N</u> ame:	ОСОМ
<u>P</u> ort number:	135

Creating the Account



Configuring DCOM

Step 1: Run the dcomcnfg.exe program to launch component services. Right clieck "My Computer" and choose "Properties".

🎲 Component Se	rvices					_ 🗆 🗵
🚱 File Action	View Window Help					_8×
⇐ ⇒ 🗈 💽	X 🗗 🖻 😫 💷 ³			9		
Console Root	Services	Comput	er 4 object(s)			
📄 🧰 Compute	ers					
Event Viev	Stop MS DTC	EOM+	DCOM Config	Distributed	Running	
🗄 🍓 Services (I	Refresh all components	plications		Transacti	Processes	
	View 🕨					
	New Window from Here					
	Properties					
	Help]				
Opens the properties	dialog box for the current sel	ection.				

- Step 2: Select the "Default Properties" tab page.
- **Step 3:** Use the following settings:

Field Name	Set to
Enable Distributed COM on this computer	Checked
Default Authentication Level:	Default
Default Impersonation Level:	Identify

My Computer Properties
Default Protocols MSDTC COM Security General Options Default Properties
Enable Distributed COM on this computer
Enable COM Internet Services on this computer
Default Distributed COM Communication Properties
The Authentication Level specifies security at the packet level.
Default Authentication Level:
Default
The impersonation level specifies whether applications can determine who is calling them, and whether the application can do operations using the client's identity.
Default Impersonation Level:
Identify 🔽
Security for reference tracking can be provided if authentication is used and that the default impersonation level is not anonymous. Provide additional security for reference tracking
OK Cancel Apply

General	Options	Default Properties
Default Protocols	MSDTC	COM Security
ccess Permissions —		
You may edit who is also set limits on app	allowed default access plications that determine	s to applications. You ma s their own permissions.
	Edit Limits	Edit Default
You may edit who is activate objects. Yo determine their own	s allowed by default to la u may also set limits on permissions.	aunch applications or applications that
You may edit who is activate objects. Yo determine their own	s allowed by default to la u may also set limits on permissions. Edit Limits	aunch applications or applications that Edit Default
You may edit who is activate objects. Yo determine their own	s allowed by default to la su may also set limits on permissions. Edit Limits	aunch applications or applications that Edit Default
You may edit who is activate objects. Yo determine their own	s allowed by default to la su may also set limits on permissions. Edit Limits	aunch applications or applications that Edit Default
You may edit who is activate objects. Yo determine their own	s allowed by default to la ou may also set limits on permissions. Edit Limits	aunch applications or applications that Edit Default
You may edit who is activate objects. Yo determine their own	s allowed by default to la su may also set limits on permissions. Edit Limits	aunch applications or applications that Edit Default

Step 4: Select the "COM Security" tab page.

Step 5: Click on the "Edit Limits..." of "Access Permissions" button to set.

cess Permission		?
Group or user names:		
Permissions for ANONYMOUS LOGON Local Access Remote Access	Add Allow	Remove Deny
		-
	OK	Come

Access Permission		? 🛛
Security Limits		
Group or user names:		
ANONYMOUS LOGON		
👥 Everyone		
	Add	Remove
Permissions for Everyone	Allow	Deny
Local Access		
Remote Access		
		-
	ОК	Cancel

Step 6: Click on the "Edit Default..." of "Access Permissions" button to set.

Add	Remove Deny
	Add Allow

Step 7: Click on the "Edit Limits..." of "Launch and Activation Permissions" button to set.

ecurity Limits Group or user names: Administrators (ZIBET Var Everyone	dministrators)	
Permissions. <i>fer</i> Everyone	Add	Remove
Local Launch Remote Launch Local Activation Remote Activation	V V V V	
	OK	Cancel

Step 8: Click on the "Edit Default..." of "Launch and Activation Permissions" button to set.

Administrators (ZIBET \Adm Everyone INTERACTIVE	ninistrators)	
SYSTEM	Add	Remove
Local Launch Remote Launch Local Activation Remote Activation	XIOW V V V	

Step 9: Right click on the "NAPOPC_XPE DA Server" of "DCOM Config" button and select "Properties".



Step 10: Select the "Location" tab page and check "Run application on the following computer". And enter the Server IP here.

APOPC_XPE DA Server Properties	? :
General Location Security Endpoints Identity	
The following settings allow DCOM to locate the correct application. If you make more than one selection, then D applicable one. Client applications may overide your sele	computer for this COM uses the first ctions.
 Run application on the computer where the data is lo Run application on this computer. 	cated.
Run application on the following computer:	
192.168.1.91	Browse

Step 11: Select the "Identity" tab page and check "The launching user"

NAPOPC_ST DA Server	Properties 🛛 🛛 🛛
General Location Secu	rity Endpoints Identity
Which user account do ye	ou want to use to run this application?
C The interactive user.	
The launching user.	
C This user.	
User:	Browse
Password:	
Confirm password:	
C The system account (services onlu)
C The Ground Constant (sannaas angg.
	OK Cancel Apply

Step 12: Restart XPC

Shut Down	n Windows Windows Embedo Standard	⊻ ded Microsoft
	What do you want the computer to do?	
~	Log off Administrator	
	Log off Administrator Shut down	
	Restart	
	Stand by	
	OK Cancel	Help

3.2.4 Configuring On the Client Site (WinPAC) System Requirement

OS version: WinPAC OS 1.3.04 or later Program: NAPOPC_CE5 DCOMCnfg.exe WinPAC Utility 2.0.2.1 or later

Configuring DCOM

Step 1: Run the \\NAPOPC_CE5\napopc_ce5boot.exe program to register.

Step 2: Run the dcomcnfg.exe program and choose "Default".

seeu eannalan	ntion Utility 🛛 🔛
Active SSCE Rej SSCE DropTable SSCE Error Obje Microsoft SQL S Yuv Video Rend XviD MPEG-4 Vid MP3 Audio Decc PSFactoryBuffe NAPOPC_CES D	Difation Object
Default Securit	i 🛛
Authentication:	
T	
impersonation:	ANONYMOUS 🔽
Enable DCOM	ANONYMOUS Secure references
Enable DCOM	ANONYMOUS Secure references

Step 3: Select the "Access" button to add an account which is identical to the account on the server site.

	ОК
	Cancel
	Add
	Delete
add Permissions	
Add Permissions Principal: Test	ок
Add Permissions Principal: Test Grant 💽 🔲 Group	OK Cancel

- **Step 4:** Select the "Launch" button to add an account which is identical to the account on the server site as above.
- Step 5: Select "Class" button of "DCOM Configuration Utility" to setup "Class Activation". Uncheck "Run Locally" and check "Run remotelly". Enter IP address of server site.

OK
Cancel

Step 6: Execute "WinPAC Utility->Network Setting->Users and Password"

WinPAC Utility [2.0.2.1]
File Help Configuration
System Setting Ethernet Setting Network Setting System Information Au
FTP Setting Users and Password
User name Password
Test **** Add Delete
User name Password Note: The accounts is used to login the servers search as Telnet, FTP, WebServer etc on WinPAC.
Setting

Step 7: Fill out "User name", "Password", and press "Add". The "User name" and "Password" must be the account we set at **Step 3.** After pressing "Add", press "Setting" to finish all settings.

Step 8: Select "Control Panel" → "Owner Properties" → "Network ID" and fill out the User name/Password which is identical to the account on the server site.

Owner Prope	erties 🛛 🕜 OK 🔀
Identification	Notes Network ID
Windows CE gain access t Enter the use domain provie administrator	uses this information to o network resources. er name, password, and ded by your network
User Name:	Test
Password:	****
Domain:	

Step 9: Run WinPAC Utility to save and reboot.

Step 10: Execute OPC client for testing.

Eile	OPC	<u>V</u> iew	Help	A	¥	lands		1000	I	8		
Tag						V	alue				, i	Qual Flag

Rheid	ОК
	Cancel

4 The Application of NAPOPC_CE5

User can develop an incredible application combining with OPC client, Modbus RTU/TCP client, NAPOPC_ST, and NAPOPC_XPE. If using "Rule Script" inside the NAPOPC_CE5, user can not only save lots of time developing system, but also create a more stable and safer system. The five sections below describe the timing and method to apply in different kind of situation.

4.1 NAPOPC_CE5 with OPC Client

NAPOPC_CE5 is designed as OPC based architecture, therefore it supports OPC client naturally. Many WinCE based OPC clients in the world can apply with it. Please refer to its user manual for detail information. The following sections show you how "InduSoft Web Studio Version 6.0" connects to Quicker.

InduSoft Web Studio is a powerful, integrated collection of automation tools that includes all the building blocks needed to develop human machine interfaces (HMIs), supervisory control and data acquisition (SCADA) systems, and embedded instrumentation and control applications. Web Studio runs in native Windows NT, 2000, XP and CE.Net 5.0 environments and conforms to industry standards such as Microsoft DNA, OPC, DDE, ODBC, XML, SOAP and ActiveX. For more information please visit: <u>http://www.indusoft.com/</u>

	Type	Channel/Location	Mil Address Value	Scaling	Description	
8 Ch00	Bit Output	0	9	100100000		
\$ Ch01	Bt Output	1	10			
8 Ch02	Bit Output	2	11			
\$ 0103	Bit Output	3	12			
\$ Ch04	Bit Output	4	13			
8 Ch05	Bit Cutput	5	14			
\$ 0106	Bit Output	6	15			
\$ 0.07	Bit Output	7	16			
\$ Ch08	Bit Output		17			
\$ 0.09	Bt Output	9	18			
8 Ch10	Bt Output	10	19			
# Chill	Bit Output	11	20			
8 Oh12	Bit Output	12	21			
8 Ch13	Bit Output	10	22			
\$ (h14	BI CUIDUE	14	23			
\$ Ch15	BI OUDUL	15	29			

Step 1: Before using the InduSoft OPC Client module, you need to configure the NAPOPC_CE5 on the WinPAC-8000/5000 first.
Step 2: Run InduSoft Web Studio version 6.0



Step 3: Create a new project.



Step 4: In the Studio Workspace window, click the OPC tab, right-click the OPC folder, and click "Insert":

*						InduSe	e We	ch Studie	5 - E)is pile	72											00
Sie Stie y	Sew Lasert Booles	t Jook 3	Lindow	8-6					_							_	_	_				
) 4 日	# X % 6	X D	-	B H	00		4:00	101	12	1.44	$\rightarrow i$	e i	- A						10			
Data	. × .	145-114			1.00	. 1		100.000	-	and I	Los S	2.1	L. 7	1+	ti z	s D	D. D.	1.00	- 34.			
Workstore	10 107 11			×. •	1.00	5- 1 m		200, 204	1	100	100.2	PO1	~ 4		14.5	<		1	6.24			1
mouspace -									11													
* an Project	C. LOULAPP								11													
	Period.								12													A 1
	luert a					1111	111		11	111					111	111			100	1111		- 1 - 1
-																						- A- 1
									22													1.2
						1111	111		11	111					111	111			111	1111		<u> </u>
									12						111	223			222			
									22													-
						1111			12.	111					123	111			2.2.2	1111		: 🖻 :
						1111			12	111					111	2.2.3			222			
						$\boldsymbol{x}_{i} = \boldsymbol{x}_{i} + \boldsymbol{x}_{i}$			1.1						1.1							88
									11	111						2.2.3						8
						1.1.1.1			1.1													1 m 4
									11						111	111						1.0
				1111		1010			11	1.1.1					111	2.2.2			111	2223		
									11							111						128
						1111			12.	111					111	2.23			10.0	1111		i 🚍 L
									• •													
									11													
																						- R
									22													9
									22													
						1111			1.1	111					1.11	111		1.1.1	1.1.1	1111		
									11													
Con 15	One 1 📫 Take	2 Conn																				
		_																				
×	Less Le								_	112	-											
. Nette	1000 Q	Swath C	ostavo	104						11.												12
3																						- 8
2	1 /2001 2003	I DD A	_				_		-		-	-	Lon C	10.0								1
E 1404 (10	VIGT VDB3	1084		0.4	_	_	_		1	j o	140	-1/	FOR V	and /				0.40	_	_	_	18.1
-							_		_	_		_				_						

Step 5: OPC Attributes window pops up.

		OPCCL002.OPC	000
Description	Server Identifier:	Disable:	
Read Update Rate (ms)	Percent Deadband:	Status:	
Renote Server Name:	(Browse)		
	Tag Name	Item	Scan
1			*
2			•
3			•
4			•
5			•

Step 6: Click on the Server Identifier: Write "NAPOPC.Svr".

Description:	Server Identifier:	Disable:
ICPDAS	NAPOPC.Svr	÷
Read Update Rate (ms):	Percent Deadband:	Status:
100		
Remote Server Name:		
	Browse	

The configuration table for OPC has the following entries:

- Description: this field is used for documentation only. The OPC Client module ignores it.
- Server Identifier: this field should contain the name of the server you want to connect. If the server is installed in the computer, its name can be selected through the list box.
- Disable: this field should contain a tag or a constant. If its value is different of zero, the communication with the OPC server is disabled.
- Update Rate: this field indicates how often the server will update this group in milliseconds. If it is zero indicates the server should use the fastest practical rate.
- Percent Deadband: this field indicates the percent change in an item value that will cause a notification by the server. It's only valid for analog items.
- Tag Name: these fields should contain the tags linked to the server items.
- Item: these fields should contain the name of the server's items
- **Step 7**: In the first cell of the Tag Name column type the tag name created in database.
- **Step 8**: In the first cell of the item, you have to write it the same as the NAPOPC_CE5 configuration. Please refer to the demo at "CD:\Compact Flash\NAPOPC_CE5\Demo\InduSoft\Full"

e		OPCCL001.OPC	0	000	
Description	Canel Identifier	Disable			
Basiling Ba	INAPUPC.SVF	Statur			
100					
Remote Server N	lame:				
	Biowse				
	Tag Name	Item	Scan		
1	Tag Name 18064_Ch00	8064_1.DOs.Ch00	Scan Always -		
1 2	Tag Name 18064_Ch00 18064_Ch01	8054_1.DOS.Ch00 8054_1.DOS.Ch01	Scan Always - Always -	ĺ	
1 2 3	Tag Name 18064_Ch00 18064_Ch01 18064_Ch02	8064_1.DOs.Ch00 8064_1.DOs.Ch01 8064_1.DOs.Ch01 8064_1.DOs.Ch02	Scan Always - Always - Always -		
1 2 3 4	Tag Name 18064_Ch00 18064_Ch01 18064_Ch02 18064_Ch03	8064_1.DOs.Ch00 8064_1.DOs.Ch01 8064_1.DOs.Ch01 8064_1.DOs.Ch02 8064_1.DOs.Ch03	Scan Always - Always - Always - Always -		
1 2 3 4 5	Tag Name 18064_Ch00 18064_Ch01 18064_Ch02 18064_Ch03 18064_Ch04	8064_1.DOs.Ch00 8064_1.DOs.Ch01 8064_1.DOs.Ch01 8064_1.DOs.Ch02 8064_1.DOs.Ch03 8064_1.DOs.Ch04	Scan Always - Always - Always - Always - Always -		
1 2 3 4 5 6	Tag Name 18064_Ch00 18064_Ch01 18064_Ch02 18064_Ch03 18064_Ch04 18064_Ch05	8064_1.DOs.Ch00 8064_1.DOs.Ch01 8064_1.DOs.Ch01 8064_1.DOs.Ch02 8064_1.DOs.Ch03 8064_1.DOs.Ch04 8064_1.DOs.Ch05	Scan Always - Always - Always - Always - Always - Always -		
1 2 3 4 5 6 7	Tag Name 18064_Ch00 18064_Ch01 18064_Ch02 18064_Ch03 18064_Ch03 18064_Ch04 18064_Ch05 18064_Ch05 18064_Ch06	8064_1.DOs.Ch00 8064_1.DOs.Ch01 8064_1.DOs.Ch02 8064_1.DOs.Ch02 8064_1.DOs.Ch03 8064_1.DOs.Ch04 8064_1.DOs.Ch05 8064_1.DOs.Ch06	Scan Always - Always - Always - Always - Always - Always - Always -		
1 2 3 4 5 6 7 8	Tag Name 18064_Ch00 18064_Ch01 18064_Ch02 18064_Ch03 18064_Ch03 18064_Ch05 18064_Ch05 18064_Ch05 18064_Ch05 18064_Ch05 18064_Ch05 18064_Ch07	8064_1.Dos.Ch00 8064_1.Dos.Ch01 8064_1.Dos.Ch01 8064_1.Dos.Ch02 8064_1.Dos.Ch03 8064_1.Dos.Ch04 8064_1.Dos.Ch05 8064_1.Dos.Ch06 8064_1.Dos.Ch06	Scan Always Always Always Always Always Always Always Always Always		

Step 9: Repeat the step between 7 and 8 to add more tags.

- Step 10: Creating a Text String for the Input/Output Dynamic. Click the Text icon on the Object Editing toolbar. Position the crosshairs in the MAIN.SCR. Press the"#" key three times to display "###" in the gray square.
- **Step 11**: Click the Text Input/Output property icon on the Object Editing toolbar. *Text I/O* appears in the drop-down menu of the Object Properties window. In the Tag/Expression field type the tag name you want to link.

ICPDAS			? F1-Help	F2-Logon	F5-Exit	*******
WinCon(I-80	64/1-805	1/1-80	56/1-8024/1	-8017H) D	emo Progra	m
1-8064	1-8051		1-8056	1-8024	1-80171	n see see s
DO_Ch00 ###	DI_Ch00	###	DO_Ch00 ###	AO_Ch00	### ### Al_Ch00	###.###
DO_Ch01 ###	DI_Ch01	###	DO_Ch01 ###	AO_Ch01	### ### Al_Ch01	1. ### ### 1.1.1.1
DO_Ch02 ###	DI_Ch02	### 1	DO_Ch02 ###	AO_Ch02	### ### Al_Ch02	
DO_Ch03 ###	DI_Ch03	###	DO_Ch03 ###	AO_Ch03	###.### Al_Ch03	###_###
DO_Ch04 ###	DI_Ch04	###	DO_Ch04 ###		AI_Ch04	###
DO_Ch05 ###	DI_Ch05	#### \ \	DO_Ch05 ###		AI_Ch05	i i 🛲 i si si si s
DO_Ch06 ###	DI_Ch06	###	DO_Ch06 ###		AI_Ch06	1 *** * 111111
DO_Ch07 ###	DI_Ch07	###	DO_Ch07 ###		AI_Ch07	1. ### 1.111111
	DI_Ch08	###	DO_Ch08 ###			
an anar ann :	DI_Ch09	###	DO_Ch09 ###			
	DI_Ch10	###	DO_Ch10 ###			
an ann ann i	DI_Ch11	###	DO_Ch11 ###			
	DI_Ch12	###	DO_Ch12 ###			
an ann ann :	DI_Ch13	###	DO_Ch13 ###			
	DI_Ch14	###	DO_Ch14 ###			
	DI_Ch15	###	DO_Ch15 ###			

Step 12: After you finish the configuration. Execute the InduSoft Remote Agent by clicking "Compact Flash\Indusoft\CEServer.exe"

<u>File E</u> dit <u>V</u> iew	<u>G</u> o F <u>a</u> vorites	; 🛛 🗢 🗕 🗈					
Address \Compact Flash\Indusoft							
Name	Size	Туре	Modified				
🗁 Bin		Folder					
🗁 Drv		Folder					
🗁 TEST		Folder					
💌 CEApp.INI	109 bytes	INI File	1/27/2005 2:16:20 PM				
🔊 CEServer.exe	218KB	Application	11/12/2004 4:34:48 PM				
🛋 CEServer.INI	103 bytes	INI File	1/27/2005 2:46:48 PM				
🎎 Recovery.exe	11.0KB	Application	11/18/2004 6:17:30 PM				
Salaregview.exe	11.5KB	Application	11/25/2002 6:43:54 PM				

Step 13: Click "Project → Execution Environment" then select "Network IP" to press the IP of WinPAC-8000/5000.

Target Application Import CE License	
- Target Station	Connect Disconnect
Network IP: 10.0.0.80 Serial Port: COM1 Advanced	Platform:
○ <u>M</u> icrosoft ActiveSync	Install system files) 🗹 Only newer files

Step 14: Click "Connect" then select "Application → Send to Target"

Execution Environment	0
Target Application Import CE License	
Application Path	
Local: 3:VOPCVOPCVINDUSO~1\Test\	
Target: \Compact Flash\Indusoft\Test\	
Send To TargetOnly newer filesRunStatus:Send FileStopApplication was started.	
	Close

Step 15: Execute your application by clicking "Start". After that, you will see your runtime HMI.

Remote Agent (v4.3)	×
Connection status:	
Log:	
Runtime was started.	4
Device connection via Network (TCP/IP)	S <u>e</u> tup
nothon (rei /ir /	<u>S</u> tart
Local IP: 10.0.0.80	E <u>×</u> it

<u>File</u> <u>S</u> ecurity	Tools 🖻 🙀	1			
ICPDAS		📪 F1-Help	F2-Logon	FS-Exit	01/27/2005
WinCon(I-8	064/I-8051/	I-8056/I-80 2	24/I-8017H)	Demo Prog	ıram
I-8064	I-8051	I-8056	I-8024	I-801	7H
DO_Ch00 1	DI_Ch00 0	DO_Ch00 1	AO_Ch00	3.567 AI_Ch(0 3.571
DO_Ch01 U DO Ch02 1	DI_Ch01 0 DI Ch02 0	DO_Ch01 DO Ch02 DO CH0	AO_Ch01 AO_Ch02	1.234 AI_Ch(5.678 AI_Ch(J1 1.236 J 2 2.732
DO_Ch03 0	DI_Ch03 0	DO_Ch03 0	AO_Ch03	8.642 AI_Ch	13 2.697
DO_Ch04 0	DI_Ch04 0	DO_Ch04 0		AI_Chi)4 2 15 3
DO_Ch06 0	DI_Ch06 0	DO_Ch06 0		AI_Chi	JG 5
DO_Ch07 0	DI_Ch07 0	DO_Ch07 0		AI_Chi)7 8
	DI_Ch08 U DI Ch09 O	DO_Ch08 U DO Ch09 1			
	DI_Ch10 0	DO_Ch10 0			
	DI_Ch11 0	DO_Ch11 0			
	DI_Ch12 0	DO_Ch12 0			
	DI_Ch14 0	DO_Ch14 0			
	DI_Ch15 0	DO_Ch15 0			
Main	Trend	Alarms	Recipe	PID	Monitor

4.2 NAPOPC_CE5 with Modbus RTU/TCP Client

If the third party software which supports Modbus RTU/TCP client wants to connect to NAPOPC_CE5, just remember to check the services "Modbus RTU" and "Modbus TCP". Please refer to the user manual of the third party made for setting. And for NAPOPC_CE5, please refer to the section "1.2.11 Services Setup".

4.2.1 Supported Modbus Commands

The Modbus protocol establishes the format for the master's query by placing into the device (or broadcast) address, a function code defining the requested action, any data to be sent, and an error checking field. The slave's response message is also constructed using the Modbus protocol. It contains fields confirming the action taken, any data to be returned, and an error-checking field. If an error occurred in receipt of the message, or if the slave is unable to perform the requested action, the slave will construct an error message and send it as its response.

	Code Description I/O Unit Min Max									
Code	Description	I/O	Unit	Min	Max					
01(0x01)	Read Coil	Status In	Bit	1	2000(0x7D0)					
02(0x02)	Read Discrete Inputs	Status In	Bit	1	2000(0x7D0)					
03(0x03)	Read Holding Registers	Registers In	Word	1	125(0x7D)					
04(0x04)	Read Input Registers	Registers In	Word	1	125(0x7D)					
05(0x05)	Write Single Coil	Coil Out	Bit	1	1					
06(0x06)	Write Single Register	Register Out	Word	1	1					
15(0x0F)	Write Multiple Coils	Coils Out Bit	Bit	1	800					
16(0x10)	Write Multiple registers	Registers Out Word	Word	1	100					

4.3 NAPOPC_CE5 with NAPOPC_ST/NAPOPC_XPE

You can construct a complete control system from top to bottom via NAPOPC_CE5 combining with NAPOPC_ST/NAPOPC_XPE and SCADA software. Please refer to the "1.2.11 Services Setup" to set up NAPOPC_CE5 services depending on which communication way that NAPOPC_ST/NAPOPC_XPE used. NAPOPC_CE5 provides three ways, "Modbus TCP", "Modbus RTU", and "RPC Server", to communicate with NAPOPC_ST/NAPOPC_XPE. At NAPOPC_ST/NAPOPC_ST/NAPOPC_XPE site, please refer to the "Adding A New Modbus TCP Controller", "Adding A New Modbus RTU Controller" and "Adding A New RPC Controller" in the NAPOPC_ST/NAPOPC_XPE user manual.

4.4 NAPOPC_CE5 with User Application

Users can develop their own application program with eVC++, VB.NET, or VC#.NET and share data with NAPOPC_CE5 via Quicker API. User can use the Modbus RTU/TCP services, or just use the share memory inside NAPOPC_CE5 to exchange data between different programs. We do not focus on the programming skill of eVC++/VB.NET/VC#.NET. We just focus on the Quicker API below.

4.4.1 Quicker API for eVC++ Developer

```
Step 1:
```

Install pac270_sdk_2008xxxx.msi

```
Step 2:
```

Create a new eVC++ project with choosing "Win32[WCE ARMV4I] CPU" option **Step 3:**

#include "WinConAgent.h"

Step 4:

Refer to the following functions to design your own program

Step 5:

Build your project with release mode.

Note: Quicker.dll and eVC++ application program must be copied to the same folder in the WinPAC-8000/5000

System Function

unsigned char StartQuicker(unsigned char iMode) unsigned char StopQuicker(void) unsigned char GetVersion()

QuickerIO Function

unsigned char GetDIO(unsigned short iMBAddr, unsigned char *iRecv, unsigned char iAttribute); unsigned char GetAIO_Short(unsigned short iMBAddr, short *iRecv, unsigned char iAttribute); unsigned char GetAIO_Long(unsigned short iMBAddr, flong *iRecv, unsigned char iAttribute); unsigned char GetAIO_Float(unsigned short iMBAddr, float *iRecv, unsigned char iAttribute); unsigned char GetAIO_Word(unsigned short iMBAddr, unsigned short *iRecv, unsigned char iAttribute); unsigned char GetAIO_DWord(unsigned short iMBAddr, unsigned short *iRecv, unsigned char iAttribute); unsigned char GetAIO_DWord(unsigned short iMBAddr, unsigned long *iRecv, unsigned char iAttribute); unsigned char SetDO(unsigned short iMBAddr, unsigned char iSend); unsigned char SetAO_Short(unsigned short iMBAddr, long *iSend); unsigned char SetAO_Float(unsigned short iMBAddr, float *iSend); unsigned char SetAO_Float(unsigned short iMBAddr, unsigned short *iSend); unsigned char SetAO_Word(unsigned short iMBAddr, unsigned long *iSend);

Modbus Function

unsigned char MBSetCoil(unsigned short iMBAddress, unsigned char iStatus, unsigned char iAttr) unsigned char MBGetCoil(unsigned short iMBAddress, unsigned char *iStatus, unsigned char iAttr) unsigned char MBSetReg(unsigned short iMBAddress, short iStatus, unsigned char iAttr) unsigned char MBGetReg(unsigned short iMBAddress, short *iStatus, unsigned char iAttr) unsigned char MBSetReg_Long(unsigned short iMBAddress, long iStatus, unsigned char iAttr) unsigned char MBGetReg_Long(unsigned short iMBAddress, long *iStatus, unsigned char iAttr) unsigned char MBGetReg_Long(unsigned short iMBAddress, long *iStatus, unsigned char iAttr) unsigned char MBGetReg_DWord(unsigned short iMBAddress, unsigned long iStatus, unsigned char iAttr) unsigned char MBGetReg_DWord(unsigned short iMBAddress, unsigned long *iStatus, unsigned char iAttr)

UserShare Function

unsigned char UserSetCoil(unsigned short iUserAddress, unsigned char iStatus); unsigned char UserGetCoil(unsigned short iUserAddress, unsigned char *iStatus); unsigned char UserSetReg_Str(unsigned short iUserAddress, char *iStatus); unsigned char UserGetReg_Str(unsigned short iUserAddress, char *iStatus); unsigned char UserSetReg_Float(unsigned short iUserAddress, float *iStatus); unsigned char UserGetReg_Float(unsigned short iUserAddress, float *iStatus); unsigned char UserGetReg_Float(unsigned short iUserAddress, float *iStatus); unsigned char UserSetReg_Short(unsigned short iUserAddress, short *iStatus); unsigned char UserGetReg_Short(unsigned short iUserAddress, short *iStatus); unsigned char UserGetReg_Long(unsigned short iUserAddress, long *iStatus); unsigned char UserGetReg_Long(unsigned short iUserAddress, long *iStatus);

4.4.1.1 System Function

This group provides three functions for users to start and stop the "NAPOPCSvr_CE5.exe" and get NAPOPC_CE5 version before using "QuickerIO Function" and "Modbus Function".

StartQuicker

This function launches the NAPOPC_CE5 with different mode. **Syntax**

[eVC++] unsigned char StartQuicker(unsigned char iMode)

[VB.NET/VC#.NET]

byte Quicker.System.StartQuicker(byte iMode)

Parameters

iMode

[in] The decimal number of kernel mode. It is always 1 now. It will provide another mode in the future.

Return Values

0 indicates success. If the NAPOPC_CE5 has been run, the function will return mode number. (Please refer to the Appendix 2.1)

Remarks

You **have to** call this function to launch the NAPOPC_CE5 before using the QuickerIO and Modbus functions.

Requirements

Runs on	Versions	Defined in	Include	Link to
WinPAC 8000/5000	4.1.0.01 and later	Quicker.lib	WinConAgent.h	
Example [eVC++] //Start up the] if (StartQuick AfxMa } else{ AfxMa	NAPOPC_CE5 wi er(1) == 0){ essageBox(_T("Sta essageBox(_T("Na	th mode 1 art NAPOPC_CE5 suc APOPC_CE5 has been	cessfully!")); started!"));	
}			, started:)),	

[VB.NET]

Quicker.System.StartQuicker(1)

[VC#.NET]

Quicker.System.StartQuicker(1)

StopQuicker

This function stops the NAPOPC_CE5. Syntax

[eVC++]				
unsigned char StopQuicker(void)				
[VB.NET/VC#.NET]				
byte Quicker.System.StopQuicker()				

Parameters

Return Values

0 indicates success. WCA_Stop means NAPOPC_CE5 has been stopped. WCA_NOT_MASTER means not the main AP which calls NAPOPC_CE5 (Please refer to the Appendix 2.1)

Remarks

NAPOPC_CE5 only can be stopped by the AP which launched it.

Requirements

Runs on	Versions	Defined in	Include	Link to
WinPAC 8000/5000	4.1.0.01 and later	Quicker.lib	WinConAgent.h	

Example

```
[eVC++]
   //Stop the NAPOPC_CE5
   if(StopQuicker() == 0){
        AfxMessageBox(_T("Stop NAPOPC_CE5 successfully!"));
    }
   else if(StopQuicker() == WCA_Stop){
        AfxMessageBox(_T("NAPOPC_CE5 has been stopped!"));
    }
   else{
        AfxMessageBox(_T("Can not terminate the NAPOPC_CE5!"));
    }
```

[VB.NET]

Quicker.System.StopQuicker()

[VC#.NET]

Quicker.System.StopQuicker()

GetVersion

This function gets the NAPOPC_CE5 version. **Syntax**

[eVC++]				
unsigned char GetVersion(void)				
[VB.NET/VC#.NET]				
byte Quicker.System.GetVersion()				

Parameters

Return Values

The return value means the version value. Ex. 209 means v2.09.

Remarks

Requirements

Runs on	Versions	Defined in	Include	Link to
WinPAC 8000/5000	4.1.0.01 and later	Quicker.lib	WinConAgent.h	

Example [eVC++]

//Get the NAPOPC_CE5 version unsigned char iQversion; iQversion = GetVersion();

[VB.NET]

Dim iQversion As Byte iQversion = Quicker.System.GetVerison()

[VC#.NET]

byte iQversion = 0; iQversion = Quicker.System.GetVersion();

4.4.1.2 QuickerIO Function

This group provides 12 functions for users to Get/Set data which's modbus address is mapping from 1 to 1000 in NAPOPCSvr_CE5. The data which's modbus address is mapping from 1 to 1000 can be accessed by OPC client and modbus master via NAPOPC_CE5.

GetDIO

This function can get a single digital I/O status from a specific modbus address.

Syntax

[eVC++] unsigned char GetDIO(unsigned short iMBAddr, unsigned char *iRecv, unsigned char iAttribute

[VB.NET/VC#.NET] byte GetDIO(ushort iMBAddr, out byte iRecv, byte iAttribute)

Parameters

iMBAddr

[in] The modbus address of specific tag in the NAPOPC_CE5. The range is from 1 to 1000.

iRecv

[out] The digital status of specific tag. 1 means ON. 0 means OFF.

iAttribute

[in] Assign which kind of digital status you want get. 1 means digital input. 0 means digital output.

Return Values

0 indicates success. WCA_ATT_ERROR means the iAttibute is neither 0 nor 1. Remarks

Requirements

Runs on	Versions	Defined in	Include	Link to
WinPAC 8000/5000	4.1.0.01 and later	Quicker.lib	WinConAgent.h	

Example

[eVC++]

//Get the digital I/O status //Get the digital input status from modbus address 1 unsigned char iRecvIn; GetDIO(1,&iRecvIn,1); //Get the digital output status from modbus address 2 unsigned char iRecvOut; GetDIO(2,&iRecvOut,0);

[VB.NET]

Dim m_GetDIOVal As Byte Quicker.QuickerIO.GetDIO(7, m_GetDIOVal, 0)

[VC#.NET]

byte m_GetDIOVal; Quicker.QuickerIO.GetDIO(7,out m_GetDIOVal, 0);

GetAIO_Short

This function can get a single analog I/O value from a specific modbus address. **Syntax**

[eVC++] unsigned char GetAIO_Short(unsigned short iMBAddr, short *iRecv, unsigned char iAttribute)

[VB.NET/VC#.NET] byte GetAIO Short(ushort iMBAddr, out short fRecv, byte iAttribute)

Parameters

iMBAddr

[in] The modbus address of specific tag in the NAPOPC_CE5. The range is from 1 to 1000.

iRecv

[out] The analog value of specific tag.

iAttribute

[in] Assign which kind of analog value you want get.

Return Values

0 indicates success. WCA_ATT_ERROR means the iAttibute is neither 0 nor 1. Remarks

Requirements

Runs on	Versions	Defined in	Include	Link to
WinPAC 8000/5000	4.1.0.01 and later	Quicker.lib	WinConAgent.h	

Example

[eVC++]

//Get the analog I/O value //Get the analog input value from modbus address 1 short sRecvIn; GetAIO_Short(1,&sRecvIn,1); //Get the analog output value from modbus address 2 short sRecvOut; GetAIO_Short(2,&sRecvOut,0);

[VB.NET]

Dim m_GetAIOVal As short Quicker.QuickerIO.GetAIO_Short(7, m_GetAIOVal, 0)

[VC#.NET]

short m_GetAIOVal; Quicker.QuickerIO.GetAIO_Short(7,out m_GetAIOVal, 0);

GetAIO_Long

This function can get a single analog I/O value from a specific modbus address. **Syntax**

[eVC++]
unsigned char GetAIO_Long(unsigned short iMBAddr, long *iRecv,
unsigned char iAttribute)
[VB.NET/VC#.NET]
byte GetAIO_Long(ushort iMBAddr, out long fRecv, byte iAttribute)

Parameters

iMBAddr

[in] The modbus address of specific tag in the NAPOPC_CE5. The range is from 1 to 1000.

iRecv

[out] The analog value of specific tag.

iAttribute

[in] Assign which kind of analog value you want get.

Return Values

0 indicates success. WCA_ATT_ERROR means the iAttibute is neither 0 nor 1. Remarks

Requirements

Runs on	Versions	Defined in	Include	Link to
WinPAC 8000/5000	4.1.0.01 and later	Quicker.lib	WinConAgent.h	

Example

[eVC++]

//Get the analog I/O value //Get the analog input value from modbus address 1 long lRecvIn; GetAIO_Long(1,&fRecvIn,1); //Get the analog output value from modbus address 2 long lRecvOut; GetAIO_Long(2,&fRecvOut,0);

[VB.NET]

Dim m_GetAIOVal As long Quicker.QuickerIO.GetAIO_Long(7, m_GetAIOVal, 0)

[VC#.NET]

long m_GetAIOVal; Quicker.QuickerIO.GetAIO_Long(7,out m_GetAIOVal, 0);

GetAIO_Float

This function can get a single analog I/O value from a specific modbus address. **Syntax**

[eVC++] unsigned char GetAIO_Float(unsigned short iMBAddr, float *iRecv, unsigned char iAttribute)

[VB.NET/VC#.NET] byte GetAIO_Float(ushort iMBAddr, out float fRecv, byte iAttribute)

Parameters

iMBAddr

[in] The modbus address of specific tag in the NAPOPC_CE5. The range is from 1 to 1000.

iRecv

[out] The analog value of specific tag.

iAttribute

[in] Assign which kind of analog value you want get.

Return Values

0 indicates success. WCA_ATT_ERROR means the iAttibute is neither 0 nor 1. Remarks

Requirements

Runs on	Versions	Defined in	Include	Link to
WinPAC 8000/5000	4.1.0.01 and later	Quicker.lib	WinConAgent.h	

Example

[eVC++]

//Get the analog I/O value //Get the analog input value from modbus address 1 float fRecvIn; GetAIO_Float(1,&fRecvIn,1); //Get the analog output value from modbus address 2 float fRecvOut; GetAIO_Float(2,&fRecvOut,0);

[VB.NET]

Dim m_GetAIOVal As Single Quicker.QuickerIO.GetAIO_Float(7, m_GetAIOVal, 0)

[VC#.NET]

float m_GetAIOVal; Quicker.QuickerIO.GetAIO_Float(7,out m_GetAIOVal, 0);

GetAIO_Word

This function can get a single analog I/O value from a specific modbus address. **Syntax**

[eVC++] unsigned char GetAIO_Word(unsigned short iMBAddr, unsigned short *iRecv, unsigned char iAttribute)

[VB.NET/VC#.NET]

byte GetAIO_Word(ushort iMBAddr, out ushort fRecv, byte iAttribute)

Parameters

iMBAddr

[in] The modbus address of specific tag in the NAPOPC_CE5. The range is from 1 to 1000.

iRecv

[out] The analog value of specific tag.

iAttribute

[in] Assign which kind of analog value you want get.

Return Values

0 indicates success. WCA_ATT_ERROR means the iAttibute is neither 0 nor 1. Remarks

Requirements

Runs on	Versions	Defined in	Include	Link to
WinPAC 8000/5000	4.1.0.01 and later	Quicker.lib	WinConAgent.h	

Example

[eVC++]

//Get the analog I/O value //Get the analog input value from modbus address 1 unsigned short usRecvIn; GetAIO_Word(1,&fRecvIn,1); //Get the analog output value from modbus address 2 unsigned short usRecvOut; GetAIO_Word(2,&usRecvOut,0);

[VB.NET]

Dim m_GetAIOVal As UInt16 Quicker.QuickerIO.GetAIO_Word(7, m_GetAIOVal, 0)

[VC#.NET]

ushort m_GetAIOVal; Quicker.QuickerIO.GetAIO_Word(7,out m_GetAIOVal, 0);

GetAIO_DWord

This function can get a single analog I/O value from a specific modbus address. **Syntax**

[eVC++] unsigned char GetAIO_DWord(unsigned short iMBAddr, unsigned long *iRecv, unsigned char iAttribute)

[VB.NET/VC#.NET]

byte GetAIO_DWord(ushort iMBAddr, out ulong fRecv, byte iAttribute)

Parameters

iMBAddr

[in] The modbus address of specific tag in the NAPOPC_CE5. The range is from 1 to 1000.

iRecv

[out] The analog value of specific tag.

iAttribute

[in] Assign which kind of analog value you want get.

Return Values

0 indicates success. **WCA_ATT_ERROR** means the iAttibute is neither 0 nor 1. **Remarks**

Requirements

Runs on	Versions	Defined in	Include	Link to
WinPAC 8000/5000	4.1.0.01 and later	Quicker.lib	WinConAgent.h	

Example

[eVC++]

//Get the analog I/O value //Get the analog input value from modbus address 1 unsigned long ulRecvIn; GetAIO_DWord(1,&ulRecvIn,1); //Get the analog output value from modbus address 2 unsigned long ulRecvOut; GetAIO_DWord(2,&ulRecvOut,0);

[VB.NET]

Dim m_GetAIOVal As UInt64 Quicker.QuickerIO.GetAIO_DWord(7, m_GetAIOVal, 0)

[VC#.NET]

ulong m_GetAIOVal; Quicker.QuickerIO.GetAIO_DWord(7,out m_GetAIOVal, 0);

SetDO

This function can set a single digital output status to a specific modbus address **Syntax**

[eVC++] unsigned char SetDO(unsigned short iMBAddr, unsigned char iSend)

[VB.NET/VC#.NET]

byte SetDO(ushort iMBAddr, byte iSend)

Parameters

iMBAddr

[in] The modbus address of specific tag in the NAPOPC_CE5. The range is from 1 to 1000.

iSend

[in] The digital status of specific tag. 1 means ON. 0 means OFF.

Return Values

0 indicates success.

Remarks

Requirements

Runs on	Versions	Defined in	Include	Link to
WinPAC 8000/5000	4.1.0.01 and later	Quicker.lib	WinConAgent.h	

Example

[eVC++]

//Set the digital output ON to modbus address 1
SetDO(1,1);

[VB.NET]

Dim m_SetDOVal As Byte Quicker.QuickerIO.SetDO(1, m_SetDOVal)

[VC#.NET]

byte m_SetDOVal; Quicker.QuickerIO.SetDO(1, m_SetDOVal);

SetAO_Short

This function can set a single analog output value to a specific modbus address **Syntax**

[eVC++]

unsigned char SetAO_Short(unsigned short iMBAddr, short *iSend)

[VB.NET/VC#.NET] byte SetAO_Short(ushort iMBAddr, out short iSend)

Parameters

iMBAddr

[in] The modbus address of specific tag in the NAPOPC_CE5. The range is from 1 to 1000.

iSend

[out] The analog value of specific tag.

Return Values

0 indicates success.

Remarks

Requirements

Runs on	Versions	Defined in	Include	Link to
WinPAC 8000/5000	4.1.0.01 and later	Quicker.lib	WinConAgent.h	

Example

[eVC++]

//Set the analog output value as 42 to modbus address 1
SetAO_Short(1,42);

[VB.NET]

Quicker.QuickerIO.SetAO_Short(1, 42)

[VC#.NET]

Quicker.QuickerIO.SetAO_Short(1, 42);

SetAO_Long

This function can set a single analog output value to a specific modbus address **Syntax**

[eVC++]

unsigned char SetAO_Long(unsigned short iMBAddr, long *iSend)

[VB.NET/VC#.NET]

byte SetAO_Long(ushort iMBAddr, out long iSend)

Parameters

iMBAddr

[in] The modbus address of specific tag in the NAPOPC_CE5. The range is from 1 to 1000.

iSend

[out] The analog value of specific tag.

Return Values

0 indicates success.

Remarks

Requirements

Runs on	Versions	Defined in	Include	Link to
WinPAC 8000/5000	4.1.0.01 and later	Quicker.lib	WinConAgent.h	

Example

[eVC++]

//Set the analog output value as 2323 to modbus address 1
SetAO_Long(1,2323);

[VB.NET]

Quicker.QuickerIO.SetAO_Long(1, 2323)

[VC#.NET]

Quicker.QuickerIO.SetAO_Long(1, 2323);

SetAO_Float

This function can set a single analog output value to a specific modbus address **Syntax**

[eVC++]

unsigned char SetAO_Float(unsigned short iMBAddr, float *iSend)

[VB.NET/VC#.NET]

byte SetAO_Float(ushort iMBAddr, out float iSend)

Parameters

iMBAddr

[in] The modbus address of specific tag in the NAPOPC_CE5. The range is from 1 to 1000.

iSend

[out] The analog value of specific tag.

Return Values

0 indicates success.

Remarks

Requirements

Runs on	Versions	Defined in	Include	Link to
WinPAC 8000/5000	4.1.0.01 and later	Quicker.lib	WinConAgent.h	

Example

[eVC++]

//Set the analog output value as 5.5 to modbus address 1
SetAO_Float(1,5.5);

[VB.NET]

Quicker.QuickerIO.SetAO_Float(1, 5.5)

[VC#.NET]

Quicker.QuickerIO.SetAO_Float(1, 5.5);

SetAO_Word

This function can set a single analog output value to a specific modbus address **Syntax**

[eVC++]

unsigned char SetAO_Word(unsigned short iMBAddr, unsigned short *iSend)

[VB.NET/VC#.NET] byte SetAO_Word(ushort iMBAddr, out ushort iSend)

Parameters

iMBAddr

[in] The modbus address of specific tag in the NAPOPC_CE5. The range is from 1 to 1000.

iSend

[out] The analog value of specific tag.

Return Values

0 indicates success.

Remarks

Requirements

Runs on	Versions	Defined in	Include	Link to
WinPAC 8000/5000	4.1.0.01 and later	Quicker.lib	WinConAgent.h	

Example

[eVC++]

//Set the analog output value as 222 to modbus address 1
SetAO_Word(1,222);

[VB.NET]

Quicker.QuickerIO.SetAO_Word(1, 222)

[VC#.NET]

Quicker.QuickerIO.SetAO_Word(1, 222);

SetAO_DWord

This function can set a single analog output value to a specific modbus address **Syntax**

[eVC++]

unsigned char SetAO_DWord(unsigned short iMBAddr, unsigned long *iSend)

[VB.NET/VC#.NET] byte SetAO_DWord(ushort iMBAddr, out ulong iSend)

Parameters

iMBAddr

[in] The modbus address of specific tag in the NAPOPC_CE5. The range is from 1 to 1000.

iSend

[out] The analog value of specific tag.

Return Values

0 indicates success.

Remarks

Requirements

Runs on	Versions	Defined in	Include	Link to
WinPAC 8000/5000	4.1.0.01 and later	Quicker.lib	WinConAgent.h	

Example

[eVC++]

//Set the analog output value as 2323 to modbus address 1
SetAO_DWord(1,2323);

[VB.NET]

Quicker.QuickerIO.SetAO_DWord(1, 2323)

[VC#.NET]

Quicker.QuickerIO.SetAO_DWord(1, 2323);

4.4.1.3 Modbus Function

This group provides 8 functions to user to add their own variables into NAPOPC_CE5 for sharing the values to modbus client via modbus service of NAPOPC_CE5. If user create internal device and create internal tag, this data can not only be accessed by modbus client but also OPC client via NAPOPC_CE5.

MBSetCoil

The function can set a coil value into NAPOPC_CE5. **Syntax**

[eVC++]
unsigned char MBSetCoil(unsigned short iMBAddress, unsigned char iStatus,
unsigned char iAttr)

[VB.NET/VC#.NET] byte MBSetCoil(ushort iMBAddress, byte iStatus, byte iAttr)

Parameters

iMBAddress

[in] The modbus address which you want to set into. The range of modbus address is from 1001 to 20999.

iStatus

[in] The coil status of specific modbus address. 1 means ON. 0 means OFF.

iAttr

[in] Assign which kind of coil you want set. 1 means input coil which will be requested by modbus function number 2. 0 means output coil which will be requested by modbus function number 1/5/15.

Return Values

0 indicates success. WCA_MBADDR_OVER means the iMBAddress over the range. The legal range is from number 1001 to number 20999. WCA_MBATTR_ERROR means the iAttr is neither 1 nor 0.

Remarks

Requirements

Runs on	Versions	Defined in	Include	Link to
WinPAC 8000/5000	4.1.0.01 and later	Quicker.lib	WinConAgent.h	

Example

//Set input coil status ON at address 1
[eVC++]
MBSetCoil(1,1,1);

[VB.NET]

Quicker.Modbus.MBSetCoil(1, 1, 1)

[VC#.NET]

Quicker.Modbus.MBSetCoil(1, 1, 1);

MBGetCoil

The function can get a coil value from a specific modbus address. **Syntax**

[eVC++] unsigned char MBGetCoil(unsigned short iMBAddress, unsigned char *iStatus, unsigned char iAttr)

[VB.NET/VC#.NET] byte MBGetCoil(ushort iMBAddress, out byte iStatus, byte iAttr)

Parameters

iMBAddress

[in] The modbus address which you want to get from. The range of modbus address is from 1001 to 20999.

iStatus

[out] The coil status of specific modbus address. 1 means ON. 0 means OFF.

iAttr

[in] Assign which kind of coil you want get. 1 means input coil which will be requested by modbus function number 2. 0 means output coil which will be requested by modbus function number 1/5/15.

Return Values

0 indicates success. WCA_MBADDR_OVER means the iMBAddress over the range. The legal range is from number 1001 to number 20999. WCA_MBATTR_ERROR means the iAttr is neither 1 nor 0.

Remarks

Requirements

Runs on	Versions	Defined in	Include	Link to
WinPAC 8000/5000	4.1.0.01 and later	Quicker.lib	WinConAgent.h	

Example

[eVC++]

//Get input coil status from address 1
unsigned char iStatus;
MBGetCoil(1,&iSatus,1);

[VB.NET]

Dim m_MBGetCoilVal As Byte Quicker.Modbus.MBGetCoil(1, m_MBGetCoilVal, 1)

[VC#.NET]

byte m_MBGetCoilVal; Quicker.Modbus.MBGetCoil(1,out m_MBGetCoilVal, 1);

MBSetReg

The function can set a register value into NAPOPC_CE5. Syntax

[eVC++] unsigned char MBSetReg(unsigned short iMBAddress, short iStatus, unsigned char iAttr)

[VB.NET/VC#.NET] byte MBSetReg(ushort iMBAddress, short iStatus, byte iAttr)

Parameters

iMBAddress

[in] The modbus address which you want to set into. The range of modbus address is from 1001 to 20999.

iStatus

[in] The register value of specific modbus address.

iAttr

[in] Assign which kind of register you want set. 1 means input register which will be requested by modbus function number 4. 0 means output register which will be requested by modbus function number 3/6/16.

Return Values

0 indicates success. WCA_MBADDR_OVER means the iMBAddress over the range. The legal range is from number 1001 to number 20999. WCA_MBATTR_ERROR means the iAttr is neither 1 nor 0.

Remarks

Requirements

Runs on	Versions	Defined in	Include	Link to
WinPAC 8000/5000	4.1.0.01 and later	Quicker.lib	WinConAgent.h	

Example

[eVC++]

//Set input register value 123 at address 1
MBSetReg(1,123,1);

[VB.NET]

Quicker.Modbus.MBSetReg(1, 123, 1)

[VC#.NET]

Quicker.Modbus.MBSetReg(1, 123, 1);

MBGetReg

The function can get a register value from a specific modbus address. **Syntax**

[eVC++]	
unsigned char MBGetReg(unsigned short iMBAddress, short *iStatus,	
unsigned char iAttr)	

[VB.NET/VC#.NET] byte MBGetReg(ushort iMBAddress, out short iStatus, byte iAttr)

Parameters

iMBAddress

[in] The modbus address which you want to get from. The range of modbus address is from 1001 to 20999.

iStatus

[out] The register value of specific modbus address.

iAttr

[in] Assign which kind of register you want get. 1 means input register which will be requested by modbus function number 4. 0 means output register which will be requested by modbus function number 3/6/16.

Return Values

0 indicates success. WCA_MBADDR_OVER means the iMBAddress over the range. The legal range is from number 1001 to number 20999. WCA_MBATTR_ERROR means the iAttr is neither 1 nor 0.

Remarks

Requirements

Runs on	Versions	Defined in	Include	Link to
WinPAC 8000/5000	4.1.0.01 and later	Quicker.lib	WinConAgent.h	

Example [eVC++]

//Get input register value from address 1 short iSataus; MBGetReg(1,&iSatus,1);

[VB.NET]

Dim m_MBGetRegVal As short Quicker.Modbus.MBGetReg(1, m_MBGetRegVal, 1)

[VC#.NET]

short m_MBGetRegVal; Quicker.Modbus.MBGeReg(1,out m_MBGetRegVal, 1);

MBSetReg_Long

The function can set a register value into NAPOPC_CE5. **Syntax**

[eVC++]

unsigned char MBSetReg_Long(unsigned short iMBAddress, long iStatus, unsigned char iAttr)

[VB.NET/VC#.NET] byte MBSetReg(ushort iMBAddress, int iStatus, byte iAttr)

Parameters

iMBAddress

[in] The modbus address which you want to set into. The range of modbus address is from 1001 to 20999.

iStatus

[in] The register value of specific modbus address.

iAttr

[in] Assign which kind of register you want set. 1 means input register which will be requested by modbus function number 4. 0 means output register which will be requested by modbus function number 3/6/16.

Return Values

0 indicates success. WCA_MBADDR_OVER means the iMBAddress over the range. The legal range is from number 1001 to number 20999. WCA_MBATTR_ERROR means the iAttr is neither 1 nor 0.

Remarks

Requirements

Runs on	Versions	Defined in	Include	Link to
WinPAC 8000/5000	4.1.0.01 and later	Quicker.lib	WinConAgent.h	

Example

[eVC++]

//Set input register value 123 at address 1
MBSetReg_Long(1,123,1);

[VB.NET]

Quicker.Modbus.MBSetReg_Long(1, 123, 1)

[VC#.NET]

Quicker.Modbus.MBSetReg_Long(1, 123, 1);

MBGetReg_Long

The function can get a register value from a specific modbus address. **Syntax**

[eVC++]

unsigned char MBGetReg_Long(unsigned short iMBAddress, long *iStatus, unsigned char iAttr)

[VB.NET/VC#.NET]

byte MBGetReg_Long(ushort iMBAddress, out int iStatus, byte iAttr)

Parameters *iMBAddress*

[in] The modbus address which you want to get from. The range of modbus address is from 1001 to 20999.

iStatus

[out] The register value of specific modbus address.

iAttr

[in] Assign which kind of register you want get. 1 means input register which will be requested by modbus function number 4. 0 means output register which will be requested by modbus function number 3/6/16.

Return Values

0 indicates success. WCA_MBADDR_OVER means the iMBAddress over the range. The legal range is from number 1001 to number 20999. WCA_MBATTR_ERROR means the iAttr is neither 1 nor 0.

Remarks

Requirements

Runs on	Versions	Defined in	Include	Link to
WinPAC 8000/5000	4.1.0.01 and later	Quicker.lib	WinConAgent.h	

Example [eVC++]

//Get input register value from address 1 long iSataus; MBGetReg_Long(1,&iSatus,1);

[VB.NET]

Dim m_MBGetRegVal As Integer Quicker.Modbus.MBGetReg_Long(1, m_MBGetRegVal, 1)

[VC#.NET]

int m_MBGetRegVal; Quicker.Modbus.MBGeReg_Long(1,out m_MBGetRegVal, 1);

MBSetReg_DWord

The function can set a register value into NAPOPC_CE5. **Syntax**

[eVC++]

unsigned char MBSetReg_DWord(unsigned short iMBAddress, unsigned long iStatus, unsigned char iAttr)

[VB.NET/VC#.NET]

byte MBSetReg(ushort iMBAddress, uint iStatus, byte iAttr)

Parameters

iMBAddress

[in] The modbus address which you want to set into. The range of modbus address is from 1001 to 20999.

iStatus

[in] The register value of specific modbus address.

iAttr

[in] Assign which kind of register you want set. 1 means input register which will be requested by modbus function number 4. 0 means output register which will be requested by modbus function number 3/6/16.

Return Values

0 indicates success. WCA_MBADDR_OVER means the iMBAddress over the range. The legal range is from number 1001 to number 20999. WCA_MBATTR_ERROR means the iAttr is neither 1 nor 0.

Remarks

Requirements

Runs on	Versions	Defined in	Include	Link to
WinPAC 8000/5000	4.1.0.01 and later	Quicker.lib	WinConAgent.h	

Example

[eVC++]

//Set input register value 123 at address 1
MBSetReg_DWord(1,123,1);

[VB.NET]

Quicker.Modbus.MBSetReg_DWord(1, 123, 1)

[VC#.NET]

Quicker.Modbus.MBSetReg_DWord(1, 123, 1);

MBGetReg_DWord

The function can get a register value from a specific modbus address. **Syntax**

[eVC++]

unsigned char MBGetReg_DWord(unsigned short iMBAddress, unsigned long *iStatus, unsigned char iAttr)

[VB.NET/VC#.NET]

byte MBGetReg_DWord(ushort iMBAddress, out uint iStatus, byte iAttr)

Parameters

iMBAddress

[in] The modbus address which you want to get from. The range of modbus address is from 1001 to 20999.

iStatus

[out] The register value of specific modbus address.

iAttr

[in] Assign which kind of register you want get. 1 means input register which will be requested by modbus function number 4. 0 means output register which will be requested by modbus function number 3/6/16.

Return Values

0 indicates success. WCA_MBADDR_OVER means the iMBAddress over the range. The legal range is from number 1001 to number 20999. WCA_MBATTR_ERROR means the iAttr is neither 1 nor 0.

Remarks

Requirements

Runs on	Versions	Defined in	Include	Link to
WinPAC 8000/5000	4.1.0.01 and later	Quicker.lib	WinConAgent.h	

Example [eVC++]

++J
//Get input register value from address 1
unsigned long iSataus;

MBGetReg_DWord(1,&iSatus,1);

[VB.NET]

Dim m_MBGetRegVal As UInt32 Quicker.Modbus.MBGetReg_DWord(1, m_MBGetRegVal, 1)

[VC#.NET]

uint m_MBGetRegVal; Quicker.Modbus.MBGeReg_DWord(1,out m_MBGetRegVal, 1);

4.4.1.4 UserShare Function

These functions allow users to add their own variables into share memory block for sharing the values with different application program. The data using these functions can not be accessed by modbus client and OPC client.

UerSetCoil

The function can set an unsigned char variable into share memory block. **Syntax**

[eVC++] unsigned char UserSetCoil(unsigned short iUserAddress, unsigned char iStatus)

[VB.NET/VC#.NET] byte UserSetCoil(ushort iUserAddress, byte iStatus)

Parameters

iUserAddress

[in] The address which you want to set into. The range of address is from 1 to 19999. *iStatus*

[in] unsigned char variable.

Return Values

0 indicates success. **WCA_USERADDR_OVER** means the iUserAddress over the range. The legal range is from number 1 to number 19999. **Remarks**

Requirements

Runs on	Versions	Defined in	Include	Link to
WinPAC 8000/5000	4.1.0.01 and later	Quicker.lib	WinConAgent.h	

Example [eVC++]

//Set coil value into address 1 UserSetCoil(1,1);

[VB.NET]

Quicker.UserShare.UserSetCoil(1, 1)

[VC#.NET]

Quicker.UserShare.UserSetCoil(1, 1);

UserGetCoil

The function can get an unsigned char variable from share memory block. **Syntax**

[eVC++]

unsigned char UserGetCoil(unsigned short iUserAddress, unsigned char *iStatus)

[VB.NET/VC#.NET]

byte UserGetCoil(ushort iUserAddress, out byte iStatus)

Parameters

iUserAddress

[in] The address which you want to get from. The range of address is from 1 to 19999. *iStatus*

[out] The pointer to an unsigned char variable.

Return Values

0 indicates success. WCA_USERADDR_OVER means the iUserAddress over the range. The legal range is from number 1 to number 19999. Remarks

Requirements

Runs on	Versions	Defined in	Include	Link to
WinPAC 8000/5000	4.1.0.01 and later	Quicker.lib	WinConAgent.h	

Example [eVC++]

//Get coil value from address 1
unsigned char iStatus;
UserGetCoil(1,&iSatus);

[VB.NET]

Dim m_UserGetCoilVal As Byte Quicker.UserShare.UserGetCoil(1, m_UserGetCoilVal)

[VC#.NET]

byte m_UserGetCoilVal; Quicker.UserShare.UserGetCoil(1,out m_UserGetCoilVal);

UserSetReg_Str

The function can set a string variable into share memory block. **Syntax**

[eVC++] <u>unsigned char UserSetReg_Str(unsigned short iUserAddress, char *iStatus)</u> [VB.NET/VC#.NET] byte UserSetReg_Str(ushort iUserAddress, char[] cSetStr)

Parameters

iUserAddress

[in] The address which you want to set into. The range of address is from 1 to 1024. *iStatus*

[out] char variable.

Return Values

0 indicates success. WCA_USERADDR_OVER means the iUserAddress over the range. The legal range is from number 1 to number 1024. **Remarks**

Requirements

Runs on	Versions	Defined in	Include	Link to
WinPAC 8000/5000	4.1.0.01 and later	Quicker.lib	WinConAgent.h	

Example

[eVC++]
 //Set string KKK into address 1
 char *SetString;
 CString m_USAValStr;
 m_USAValStr = _T("KKK");
 SetString = (LPSTR)(LPCTSTR)m_USAValStr;
 UserSetReg_Str(1,SetString);

[VB.NET]

Dim Rtn As Byte Dim UserSetRegStrVal As String

 $Rtn = Quicker.UserShare.UserSetReg_Str(1, UserSetRegStrVal.ToCharArray())$

[VC#.NET]

byte Rtn; string UserSetRegStrVal; Rtn = Quicker.UserShare.UserSetReg_Str(1, UserSetRegStrVal.ToCharArray());

UserGetReg_Str

The function can get a string variable from share memory block. **Syntax**

[eVC++]

unsigned char UserGetReg_Str(unsigned short iUserAddress, char *iStatus)

[VB.NET/VC#.NET]

byte UserGetReg_Str(ushort iUserAddress, byte[] cGetStr)

Parameters

iUserAddress

[in] The address which you want to get from. The range of address is from 1 to 1024. *iStatus*

[out] The pointer to a long variable.

Return Values

0 indicates success. WCA_USERADDR_OVER means the iUserAddress over the range. The legal range is from number 1 to number 1024. **Remarks**

Requirements

Runs on	Versions	Defined in	Include	Link to
WinPAC 8000/5000	4.1.0.01 and later	Quicker.lib	WinConAgent.h	

Example [eVC++]

-+] //Get string from modbus address 1 char iStatus[256]; UserGetReg_Str(1,iStatus);

[VB.NET]

Dim UserGetStr(256) As Byte Dim Rtn As Byte Rtn = Quicker.UserShare.UserGetReg_Str(1, UserGetStr)

[VC#.NET]

byte Rtn; byte[] UserGetStr = new byte[256]; Rtn = Quicker.UserShare.UserGetReg_Str(1, UserGetStr);

UserSetReg_Float

The function can set a float variable into share memory block. **Syntax**

[eVC++]

unsigned char UserSetReg_Float(unsigned short iUserAddress, float *iStatus)

[VB.NET/VC#.NET] byte UserSetReg_Float(ushort iUserAddress, out float iStatus)

Parameters

iUserAddress

[in] The address which you want to set into. The range of address is from 1 to 19999. *iStatus*

[out] float variable.

Return Values

0 indicates success. **WCA_USERADDR_OVER** means the iUserAddress over the range. The legal range is from number 1 to number 19999.

Remarks

Requirements

Runs on	Versions	Defined in	Include	Link to
WinPAC 8000/5000	4.1.0.01 and later	Quicker.lib	WinConAgent.h	

Example

[eVC++]

//Set register value 2.5 into address 1
UserSetReg_Float(1,2.5);

[VB.NET]

Dim Rtn As Byte Dim UserSetRegFloatVal As Single Rtn = Quicker.UserShare.UserSetReg_Float(1, UserSetRegFloatVal)

[VC#.NET]

byte Rtn; float RegFloat; Rtn = Quicker.UserShare.UserSetReg_Float(1,out RegFloat);
UserGetReg_Float

The function can get a float variable from share memory block. **Syntax**

[eVC++]

unsigned char UserGetReg_Float(unsigned short iUserAddress, float *iStatus)

[VB.NET/VC#.NET]

byte UserGetReg_Float(ushort iUserAddress, out float iStatus)

Parameters

iUserAddress

[in] The address which you want to get from. The range of address is from 1 to 19999. *iStatus*

[out] The pointer to a float variable.

Return Values

0 indicates success. WCA_USERADDR_OVER means the iUserAddress over the range. The legal range is from number 1 to number 19999. Remarks

Requirements

Runs on	Versions	Defined in	Include	Link to
WinPAC 8000/5000	4.1.0.01 and later	Quicker.lib	WinConAgent.h	

Example [eVC++]

//Get register value from address 1
float iStatus;
UserGetReg_Float(1,&iSatus);

[VB.NET]

Dim Rtn As Byte Dim m_UserGetRegFloatVal As Single Rtn = Quicker.UserShare.UserGetReg_Float(1, m_UserGetRegFloatVal)

[VC#.NET]

byte Rtn; float m_UserGetRegFloatVal; Rtn = Quicker.UserShare.UserGetReg_Float(1,out m_UserGetRegFloatVal);

UserSetReg_Short

The function can set a short variable into share memory block. **Syntax**

[eVC++]

unsigned char UserSetReg_Short(unsigned short iUserAddress, short *iStatus)

[VB.NET/VC#.NET] byte UserSetReg_short(ushort iUserAddress, out int iStatus)

Parameters

iUserAddress

[in] The address which you want to set into. The range of address is from 1 to 19999. *iStatus*

[out] short variable.

Return Values

0 indicates success. **WCA_USERADDR_OVER** means the iUserAddress over the range. The legal range is from number 1 to number 19999. **Remarks**

Requirements

Runs on	Versions	Defined in	Include	Link to
WinPAC 8000/5000	4.1.0.01 and later	Quicker.lib	WinConAgent.h	

Example

[eVC++]

//Set register value 222 into address 1
UserSetReg_Short(1,222);

[VB.NET]

Dim Rtn As Byte Dim UserSetRegShortVal As Integer Rtn = Quicker.UserShare.UserSetReg_Short(1, UserSetRegShortVal)

[VC#.NET]

byte Rtn; int RegShort; Rtn = Quicker.UserShare.UserSetReg_Short(1,out RegShort);

UserGetReg_Short

The function can get a short variable from share memory block. **Syntax**

[eVC++]

unsigned char UserGetReg_Short(unsigned short iUserAddress, short *iStatus)

[VB.NET/VC#.NET] byte UserGetReg_Float(ushort iUserAddress, out short iStatus)

Parameters

iUserAddress

[in] The address which you want to get from. The range of address is from 1 to 19999. *iStatus*

[out] The pointer to a short variable.

Return Values

0 indicates success. WCA_USERADDR_OVER means the iUserAddress over the range. The legal range is from number 1 to number 19999. Remarks

Requirements

Runs on	Versions	Defined in	Include	Link to
WinPAC 8000/5000	4.1.0.01 and later	Quicker.lib	WinConAgent.h	

Example [eVC++]

//Get register value from address 1
short iStatus;
UserGetReg_Short(1,&iSatus);

[VB.NET]

Dim Rtn As Byte Dim m_UserGetRegShortVal As Integer Rtn = Quicker.UserShare.UserGetReg_Short(1, m_UserGetRegShortVal)

[VC#.NET]

byte Rtn; short m_UserGetRegShortVal; Rtn = Quicker.UserShare.UserGetReg_Short(1,out m_UserGetRegShortVal);

UserSetReg_Long

The function can set a long variable into share memory block. **Syntax**

[eVC++]

unsigned char UserSetReg_Long(unsigned short iUserAddress, long *iStatus)

[VB.NET/VC#.NET] byte UserSetReg_Long(ushort iUserAddress, out long iStatus)

Parameters

iUserAddress

[in] The address which you want to set into. The range of address is from 1 to 19999. *iStatus*

[out] long variable.

Return Values

0 indicates success. **WCA_USERADDR_OVER** means the iUserAddress over the range. The legal range is from number 1 to number 19999. **Remarks**

Requirements

Runs on	Versions	Defined in	Include	Link to
WinPAC 8000/5000	4.1.0.01 and later	Quicker.lib	WinConAgent.h	

Example

[eVC++]

//Set register value 112233 into address 1
UserSetReg_Long(1,112233);

[VB.NET]

Dim Rtn As Byte Dim UserSetRegLongVal As Integer Rtn = Quicker.UserShare.UserSetReg_Long(1, UserSetRegLongVal)

[VC#.NET]

byte Rtn; int RegLong; Rtn = Quicker.UserShare.UserSetReg_Long(1,out RegLong);

UserGetReg_Long

The function can get a long variable from share memory block. **Syntax**

[eVC++]

unsigned char UserGetReg_Long(unsigned short iUserAddress, long *iStatus)

[VB.NET/VC#.NET] byte UserGetReg_Long(ushort iUserAddress, out long iStatus)

Parameters

iUserAddress

[in] The address which you want to get from. The range of address is from 1 to 19999. *iStatus*

[out] The pointer to a long variable.

Return Values

0 indicates success. **WCA_USERADDR_OVER** means the iUserAddress over the range. The legal range is from number 1 to number 19999.

Remarks

Requirements

Runs on	Versions	Defined in	Include	Link to
WinPAC 8000/5000	4.1.0.01 and later	Quicker.lib	WinConAgent.h	

Example

[eVC++]

//Get register value from address 1
long iStatus;
UserGetReg_Long(1,&iSatus);

[VB.NET]

Dim Rtn As Byte Dim m_UserGetRegLongVal As Integer Rtn = Quicker.UserShare.UserGetReg_Long(1, m_UserGetRegLongVal)

[VC#.NET]

byte Rtn; int m_UserGetRegLongVal; Rtn = Quicker.UserShare.UserGetReg_Long(1,out m_UserGetRegLongVal);

4.4.2 Quicker API for VB.NET/VC#.NET Developer

Step 1:

Create a smart device project

Step 2:

[Add Reference] ->QuickerNet.dll

Step 3:

Refer to the function prototype of QuickerNet.dll by Object Browser **Step 4:**

Call the functions in the QuickerNet.dll (Please refer to the Quicker_VB.NET_Demo /Quicker_VC#.NET_Demo)

Step 5:

Build your project and copy it and relative library into WinPAC-8000/5000

Note: Quicker.dll, QuickerNet.dll, and VB.NET/VC#.NET application program must be copied to the same folder in the WinPAC-8000/5000

4.5 NAPOPC_CE5 with Rule Script

NAPOPC_CE5 provides "Rule Script Editor" to user for editing the rules. This function is based on the instinctive design style to develop rule list. The program designers can easily implement their logic via "IF...THEN..." syntax into rule list to achieve the purpose of chain reaction control. The "Rule Script" is suitable within the non-critical situation. Using this function can not only avoid typing error but also save developing time.

4.5.1 Rule Script Syntax

Rule script syntax is very instinctive as well. In the "IF" area, the relation between timer and other variables is "AND". The triggered frequency of the rule is decided by the timer of each rule. If the rule has timer and the "THEN" area has "0xxxxx" variable, the "0xxxxx" variable will frequently "ON/OFF" switch like blinking function.

E x 1:

```
IF THEN ('000001' = 0.0) [Timer = '300']
```

Variable) ms Judge Status	: Logic AND/C	Variable	Assign O		VD (Delete
None 🔽			None	- ,- - ,-			Edit
None 🔽			None		AND		
None 🔽			None	<u> </u>			
Active Rule	Script (Hint: DO:()xxxxx DI:1xx	XXX AI:3XXXXX	AO:4xxxx	0		Faire
IF THEN ('	000001' = '0.0000)00') [Timer = '	300']			(Cance
IF THEN ("	000001' = '0.0000	000') [Timer = '	300']			(Car

Which means the variable "000001" will do "ON/OFF" switch every 300ms.

E x 2:

```
IF ('100001' '==' '0.000000') and ('400001' '==' '3.000000') THEN ('000001' '=' '1.000000')
```

Timer 300 ms Variable Judge Status Logic AND/OR	Variable Assign Output Logic AND
400001 V >= V 3 OR V None V <= V 0 AND V	None 💌 = 🛛 AND 💌
None 🔽 > 🔽 🛛	None 💌 = 🔽
Active Rule Script (Hint: DO:0xxxxx DI:1xxxxx	AI:3xxxxx AO:4xxxxx)
IF ('100001' '==' '0,000000') 'and' ('400001' '>-	' '3.000000') THEN ('000001' = '1.000000')

Which means the variable "000001" will do "ON" when variable "100001" is "0" and variable "400001" is "3". For more advanced application, user can use the variable in the "Internal device" to be a temporary buffer to chain each rule.

Appendix A – Error list and description

Code Description I/O Unit Min Max				
Code	Define	Description		
0	WCA_OK	OK		
102	WCA_Stop	ScanKernel has been stopped		
103	WCA_SLOTNO_OVER	Slot number must be 1 - 8		
104	WCA_ATT_ERROR	Attribute number error. It should be 1 or 0		
105	WCA_COMNO_OVER	COM port No. must be 2 or 3		
106	WCA_SLAVENO_OVER	Slave number must be 1 - 256		
107	WCA_NOT_MASTER	Not the main AP which calls ScanKernel		
108	WCA_MBADDR_OVER	Modbus DIO address must be 449 – 2048, AIO address must be		
		225 - 2048		
109	WCA_MBATTR_ERROR	Modbus attribute must be 1 or 0		
110	WCA_USERADDR_OVER	User defined address must be 1 - 8192		
111	WCA_USERRATTR_ERROR	User defined register value must be -32768 to 32767		