NAPOPC_ST DA Server

User's Manual

[For Windows XP/7/10]

(Supports 7000, 8000, 87000 series modules and modbus controllers)

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)

Warning

ICP DAS Inc., LTD. assumes no liability for damages consequent to the use of this product. ICP DAS Inc., LTD. 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., LTD. for its use, or for any infringements of patents or other rights of third parties resulting from its use.

Copyright and Trademark Information

© Copyright 2002 by ICP DAS Inc., LTD. All rights reserved worldwide.

Trademark of Other Companies

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.

Technical Service:

Please contact local agent or email problem-report to service@icpdas.com .

FAQ : opc.icpdas.com > FAQ > NAPOPC_ST DA Server

Written by Wilson Chen; Last edited by Eva Li.
Copyright Jul. 2002, by ICP DAS CO., LTD. All Rights Reserved.

Table of Contents

Warning....2
Copyright and Trademark Information....2
Trademark of Other Companies....2
License....2
Technical Service: 2
1 NAPOPC_ST DA Server....5
1.1 Installing NAPOPC_ST DA Server....6
1.2 Uninstalling NAPOPC_ST DA Server....7
1.3 File....8
1.4 Searching Modules....12
1.5 Monitoring Devices....15
1.6 Adding A New Device....16
1.6.1 Adding A New I-7K/I-8K/I-87K/ZB-2K I/O Module....16
1.6.2 Adding A New RPC Controller....19
1.6.3 Adding A New FRnet I/O....21
1.6.4 Adding A New Modbus TCP Controller....23
1.6.5 Adding A New Modbus RTU Controller....25
1.6.6 Adding A New Modbus ASCII Controller....28
1.7 Adding A New Group....31
1.8 Adding A New Tag....31
1.8.1 Adding A New Tag For I/O Module....32
1.8.2 Adding A New Tag For Controller....33
1.8.3 Scaling Settings....35
1.9 Adding Multi Tags For Modbus Device....36
1.10 Expand/ Shrink Devices....37
1.11 Read/Write Tags....38
1.12 Editing A Device/Group/Tag Properties....39
1.13 Deleting A Device/Group/Tag....40
1.14 Generating Tags....41
1.15 Configurate Initial Status....41
1.16 License Manager....42
1.17 Help....43
1.18 About....44
2 Quick Start....45
3 Connect To OPC Server....46
3.1 Optimize Your Communication....46
3.2 VB5 Client Demo Program....49
3.3 .Net Client Demo Program....52
3.4 LabVIEW....54
3.5 NATIONAL INSTRUMENTS....57
3.6 WIZCON....62
3.7 iFix....68
3.8 InduSoft....73
3.9 Citect SCADA....80
4 Remote Accessing....100
4.1 Configuring DCOM....101
4.1.1 Configuring On the Server Site (XPAC)....102
4.1.2 Configuring On the Server Site (PC)....111

4.1.3 Configuring On the Server Site (WinPAC) 120
4.1.4 Configuring On the Client Site (PC) 122
4.1.5 Configuring On the Client Site (XPAC) 131
5 Writing Client Program with VB 141
5.1 Programming with VB5 141
5.1.1 Overview of OPC & VB 141
5.1.2 Tools You Will Need to Build Your VB Client 142
5.1.3 Building Your VB Client – Step By Step 142
5.2 Programming with .Net 145
5.2.1 Limitations about .Net client programming 145
5.2.2 Tools – You Need to Build Your .Net Client 146
5.2.3 Building Your VB.Net Client – Step By Step 146
5.2.4 Building Your VC#.Net Client – Step By Step 149
6 Changes List 153
6.1 New features of NAPOPC version 3.0 153
6.1.1 New IO Kernel 153
6.1.2 Customized Module/Device Polling Time 153
6.1.3 Multi-Thread Communication 153
6.1.4 Miscellaneous 154
6.2 New features of NAPOPC version 3.09 154
6.2.1 Support Remote Procedure Call with Quicker/UPC 154
6.2.2 Switch of Single-Thread and Multi-Thread 154
6.3 New features of NAPOPC_ST version 3.11 154
6.3.1 Rename NAPOPC to be NAPOPC_ST 154
6.3.2 Bug Fix 155
6.4 Features of NAPOPC_ST version 3.20 155
6.4.1 Support ET-7000 Search 155
6.4.2 Support ZB-2K I/O 155
6.4.3 Support FRnet Module 155
6.4.4 Support Account Selection 155
6.4.5 UI Modification 155
6.5 Features of NAPOPC_ST version 3.30 155
6.5.1 Support FRnet module via RPC communication 156
6.5.2 Support host watchdog setting for I-7000 module 156
6.5.3 Support usage of USB hardkey to enhance functionality 156
6.6 Features of NAPOPC_ST version 3.54 156
6.6.1 Support CSV file to create a NAPOPC_ST project based on Modbus TCP 156
7 Reference 157

1 NAPOPC_ST DA Server

The NAPOPC_ST DA Server 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 (see page 8). All tags belong to their module when they are scanned for perform I/O. (The "OPC" stands for "OLE for Process Control" and the "DA" stands for "Data Access".)

The following two figures show the difference between traditional mechanisms and the OPC mechanism.

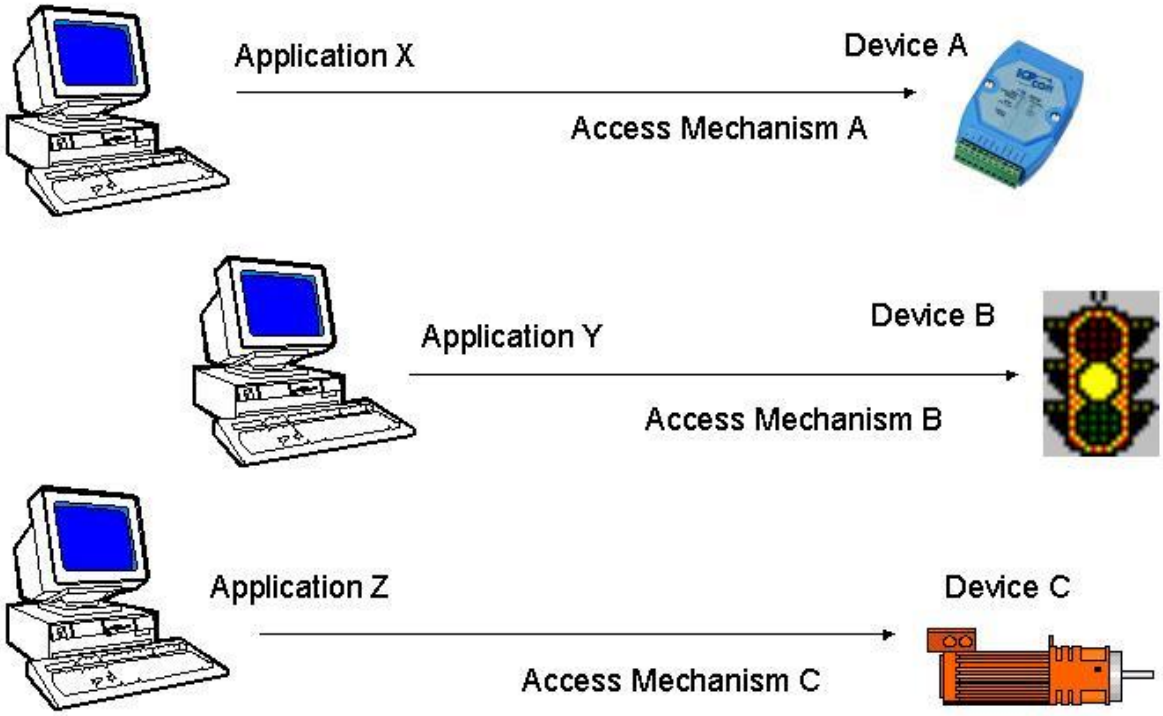


Figure 1-0-1 Traditional mechanisms used to access a device.

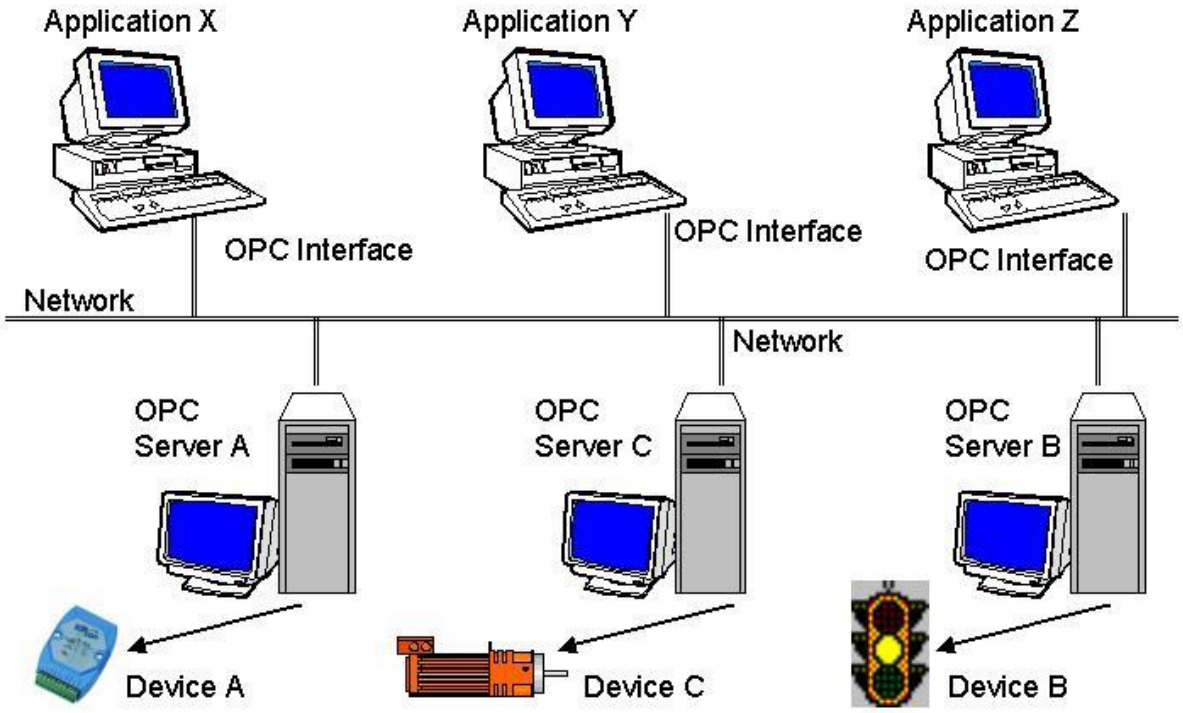


Figure 1-0-2 Using the OPC mechanism to access a device.

The main program of NAPOPC_ST DA Server is "NAPOPCSvr_ST.exe". It automatically calls the "DCON_PC.DLL", "IOCtrl.DLL" and "UART.DLL" functions on demand.

1.1 Installing NAPOPC_ST DA Server

You can get the software from the "CD: \Napdos\Napopcsvr\" or you can download it from http://opc.icpdas.com/download.htm .

Hardware Requirement:

- A personal computer with at least a Pentium, 133 MHz or faster processor
- 32 Mbytes ram (Preferably 64 Mbytes ram)
- 10 Mbytes hard disk free space

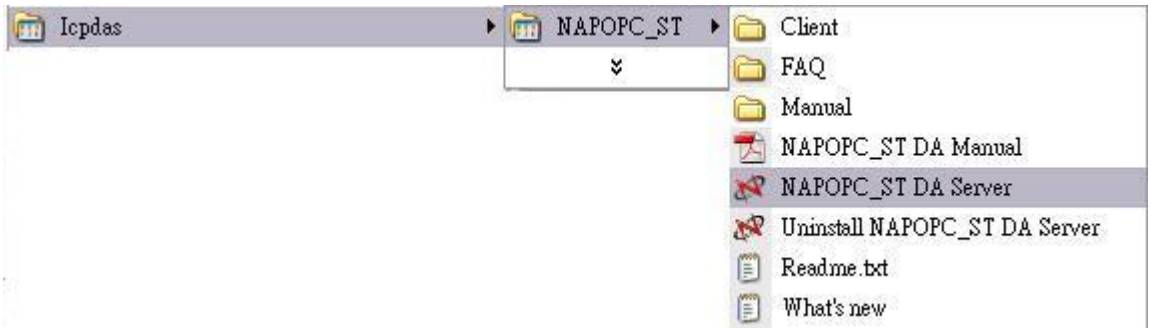
Software Requirement:

One of the following computer operating systems must be installed on your computer system.

- Windows XP
- Windows 7
- Windows 10

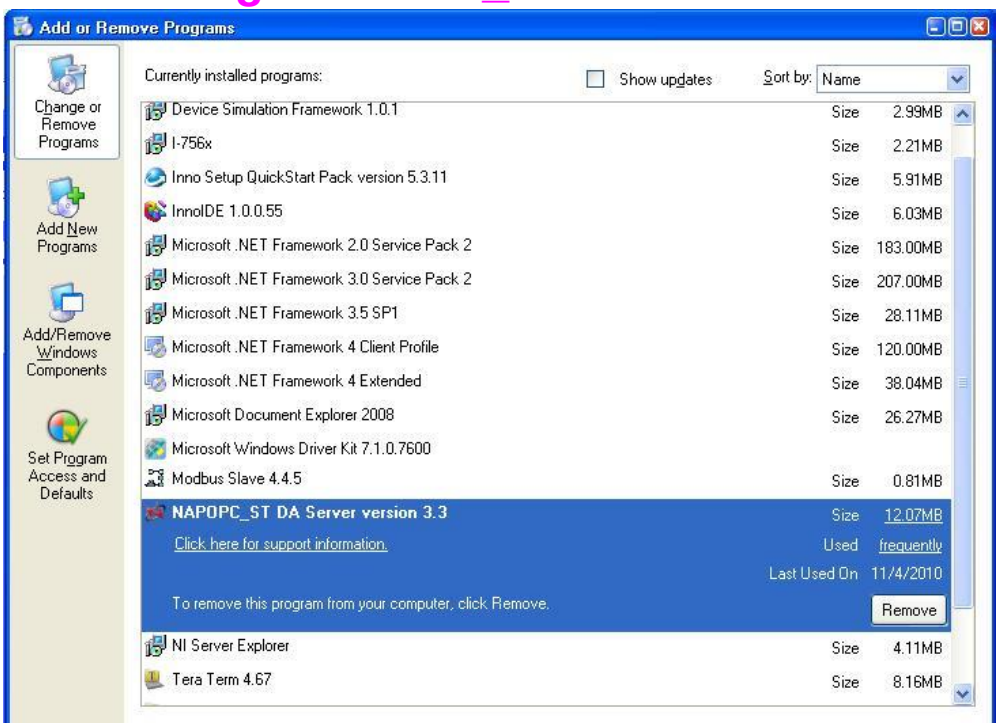
Double click the CD:\\Napdos\Napopcsvr\NAPOPC_ST DA Server.exe and follow the installing wizard to finish the installation.

After you complete the above steps, you can start the NAPOPC_ST Server by clicking the "NAPOPC_ST DA Server" as below.



Note: If you want to use FRnet modules in NAPOPC_ST, please download FRB driver and install it first. If you want to use FBR driver under windows 7/10, please make sure if it is available or not.
Download link: http://www.icpdas.com/download/frnet/index.htm
Note: Please uninstall old version NAPOPC_ST before install new NAPOPC_ST

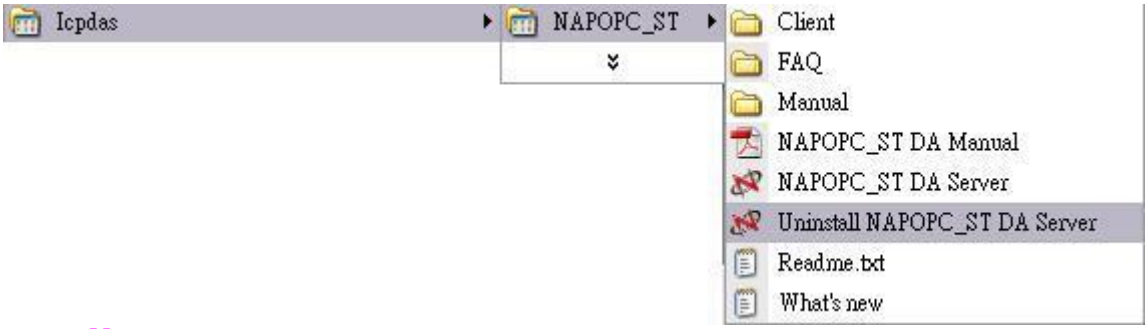
1.2 Uninstalling NAPOPC_ST DA Server



Step 1:
Go to "Add or Remove Programs" dialog and select NAPOPC_ST DA Server

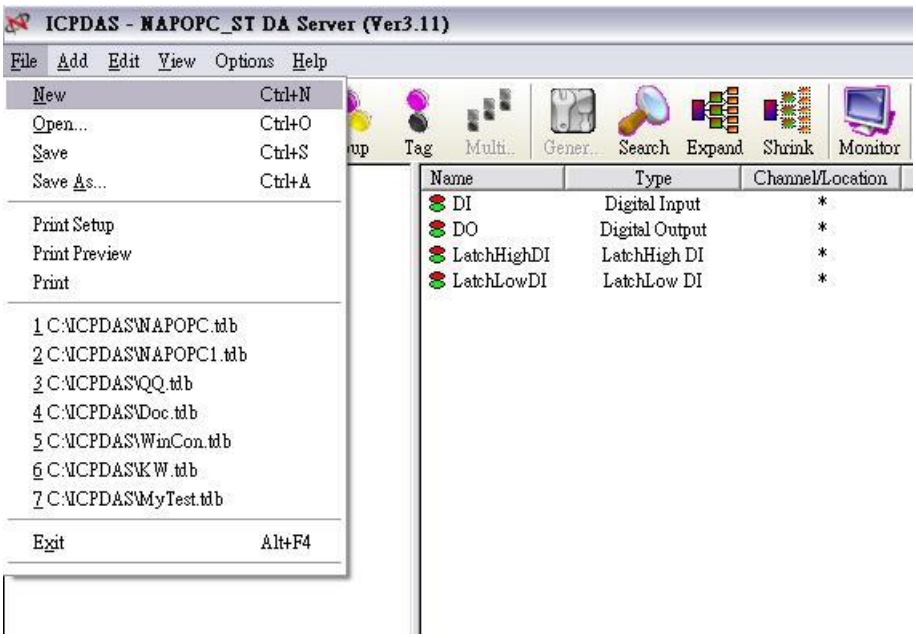
Step 2:
Press "Remove" button to remove NAPOPC_ST DA Server

NAPOPC_ST DA Server v3.30 or later version provides "Uninstall" function as below. You can use this function to uninstall NAPOPC_ST DA Server.



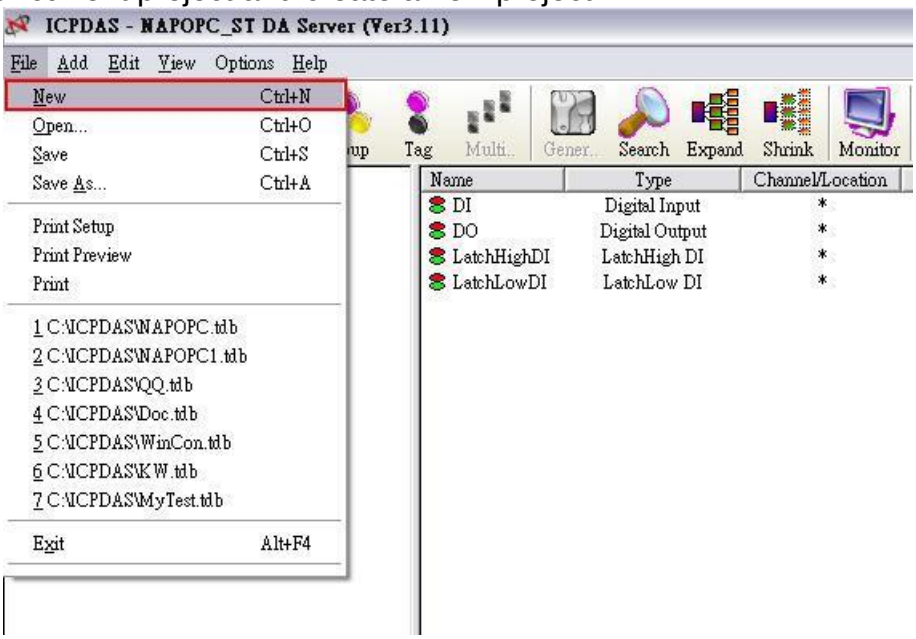
1.3 File

All configuration settings can be saved into configuration file by clicking the "File/ Save" and "File/ Save As …" menu item. The OPC server will automatically load the last configuration file with every launch.



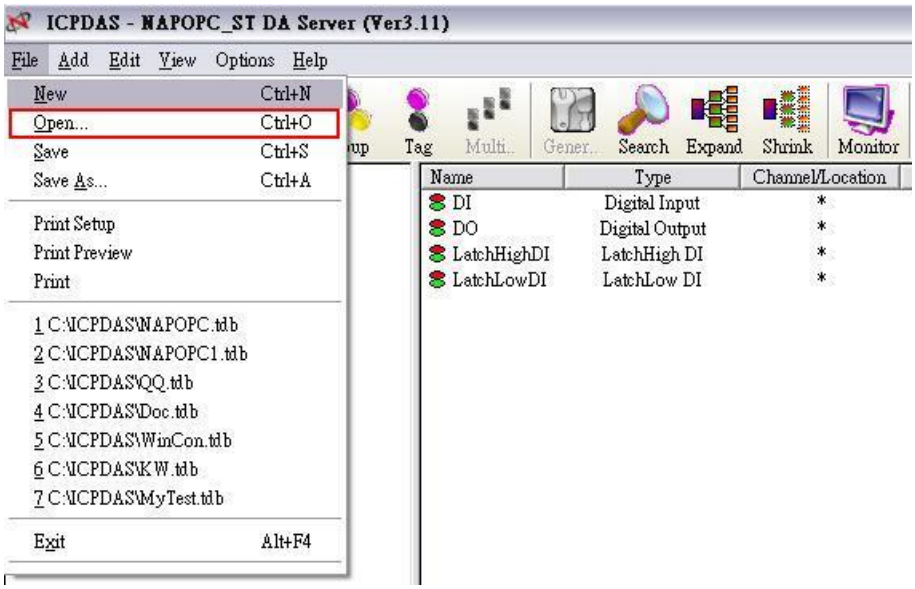
New:

Clean current project and create a new project



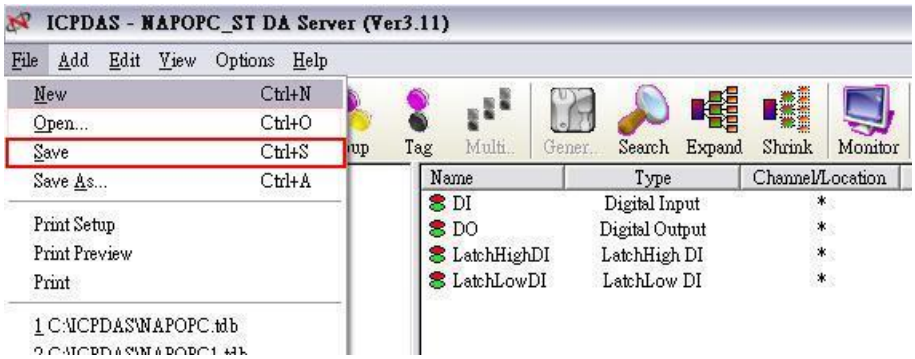
Open:

Load old NAPOPC_ST project



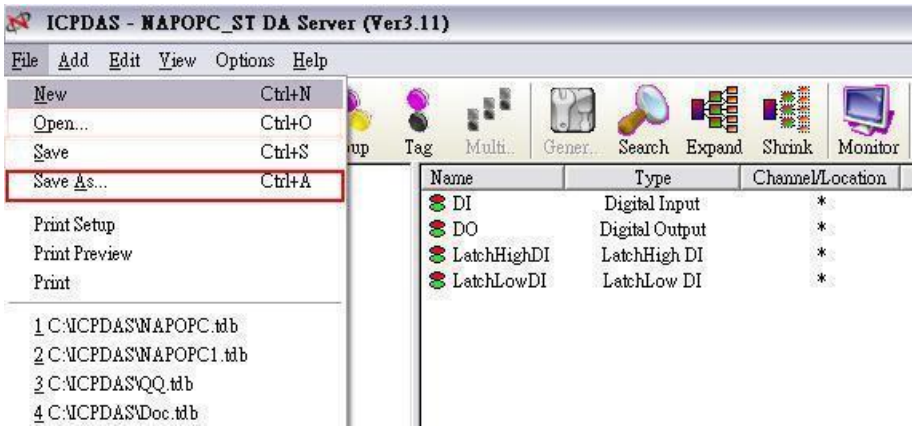
Save:

Save current NAPOPC_ST project



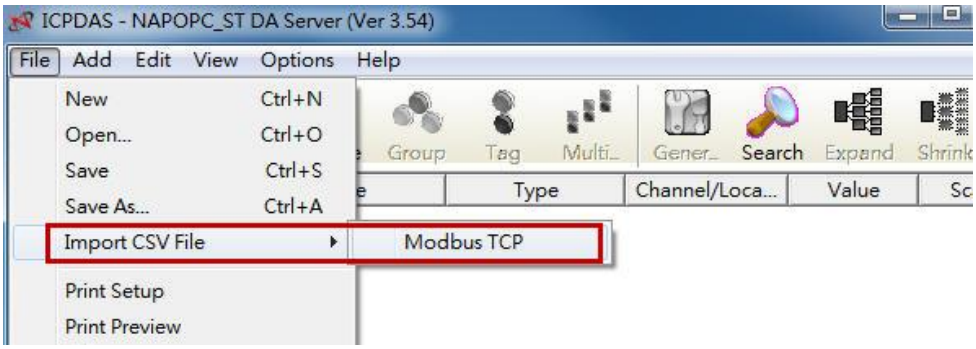
Save as…:

Save NAPOPC_ST project as a new one



Import CSV File > Modbus TCP…:

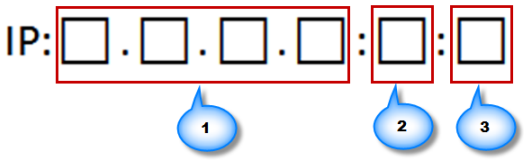
Import a CSV file to create a NAPOPC_ST project based on the Modbus TCP.



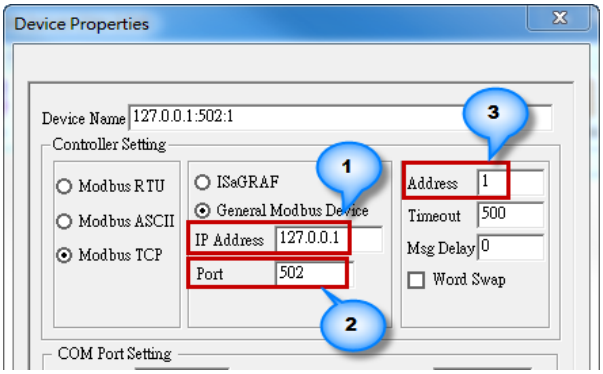
CSV file format: The file name is *.csv. The first row is for the title name and separated by commas. For example:

```
IP,GroupName,TagName,Address,Function,DataType
127.0.0.1:502:1,,G11R11_ACM_CURPOW_STS,1,1X,Bool
127.0.0.1:502:1,a1,G11R11_ACM_BAKPOW_STS,2,1X,Bool
192.168.1.6:502:1,,G12R11_B28V_CURPOW_STS,1,4X,Float
192.168.1.6:502:1,b1,G12R11_B28V_BAKPOW_STS,3,4X,Float
192.168.1.6:502:1,,G12R11_B120V_CURPOW_STS,5,4X,Float
192.168.1.6:502:1,b2,G12R11_B120V_BAKPOW_STS,7,4X,Float
192.168.1.6:502:2,,G12R11_B120V_BAKPOW_STS,9,4X,Float
```

Note 1. IP: IP Address, Port, and Address.



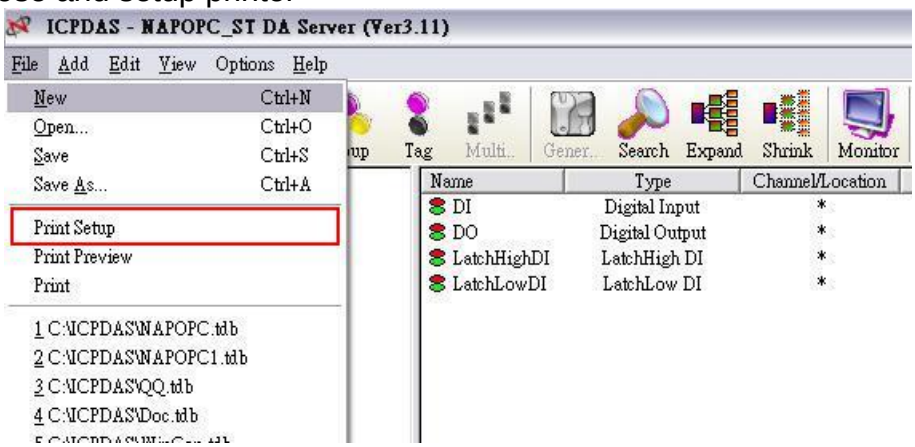
Note 2. GroupName: if blank, means the Tag belongs to the Device.



Title	Descriptions	
IP	Device IP Address, Port, Address; e.g., 127.0.0.1:502:1	
GroupName	Group name (If blank, means the Tag directly belongs to Device.)	
TagName	Tag name	
Address	Modbus address	
Function	**0X**	Coil Status, Modbus instructions 01, 05 and 15
	1X	Input Status, Modbus instruction 02
	3X	Input Register, Modbus instruction 04
	4X	Holding Register, Modbus instructions 03, 06 and 16
DataType	**bool**	Tag data type: Boolean
	short	Tag data type: 2 bytes Integer
	long	Tag data type: 4 bytes Integer
	float	Tag data type: 4 bytes Floating Point
	word	Tag data type: 2 bytes unsigned Integer
	dword	Tag data type: 4 bytes unsigned Integer

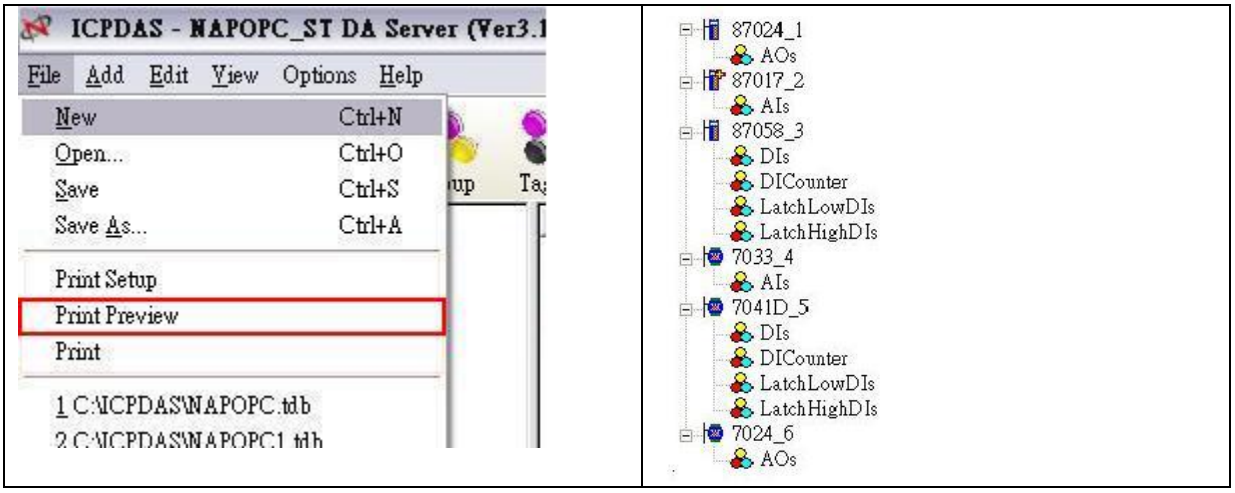
Print Setup:

Choose and setup printer



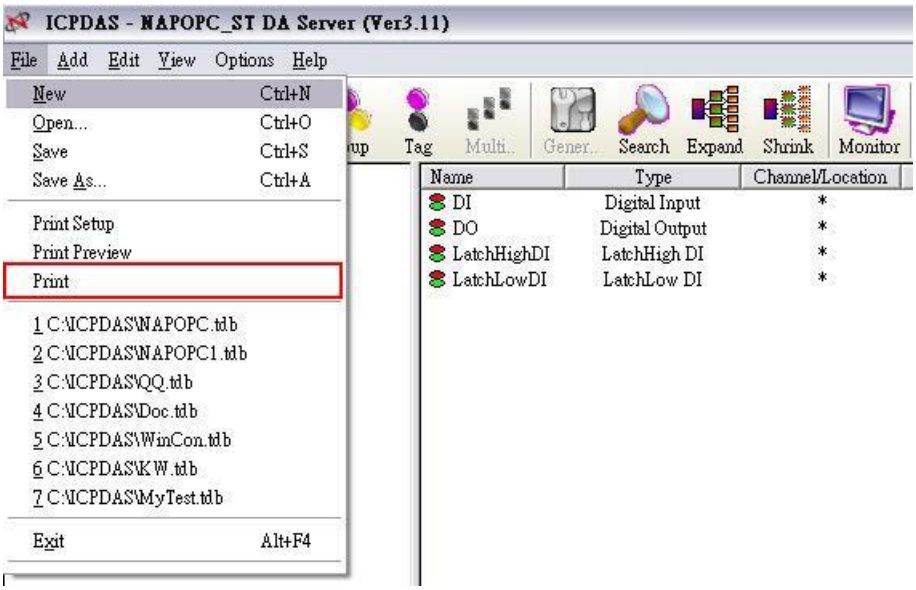
Print Preview:

Preview current modules list



Print

Print current modules list

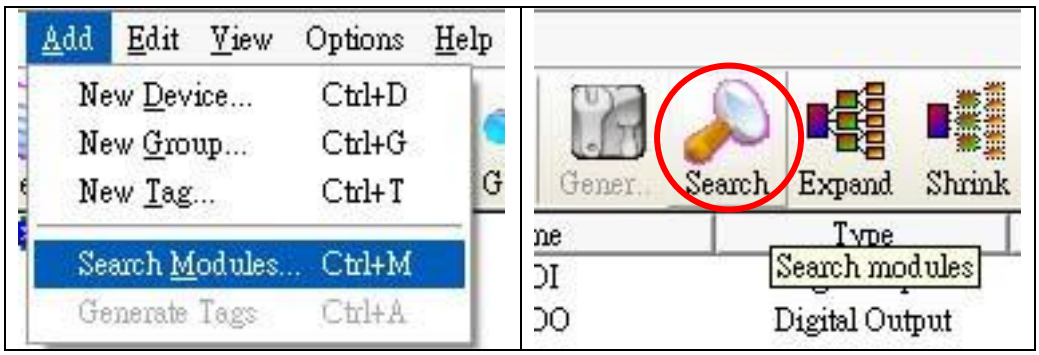


1.4 Searching Modules

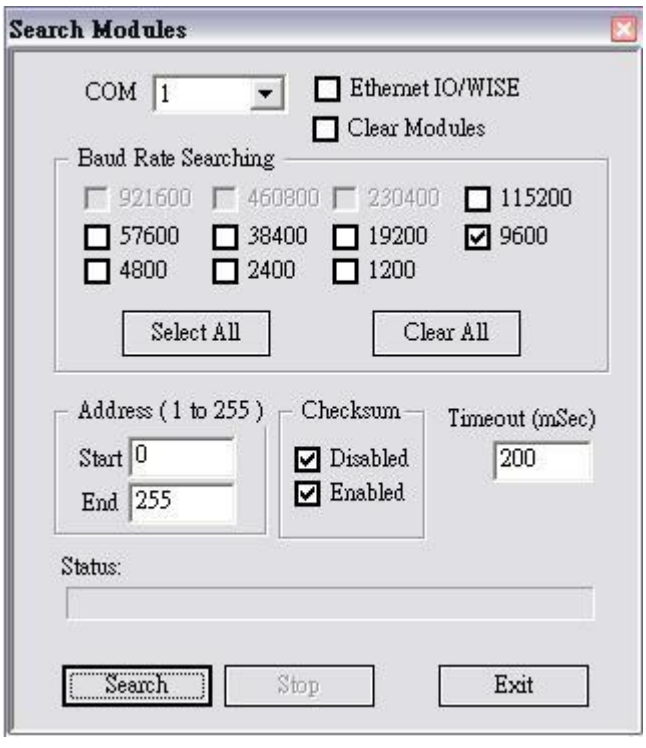
The "Search Modules…" function lets you configure the OPC server automatically. It searches the RS-232 and RS-485 network to find modules and then generates tags automatically. This function generates AI/AO, DI/DO, Latched DI and Counter tags.

NOTE:
For complete module support, please update "\ICPDAS\NAPOPC_ST\module.ini", "\ICPDAS\NAPOPC_ST\module_ET.ini", "\ICPDAS\NAPOPC_ST\module_FRnet.ini" ,and "\ICPDAS\NAPOPC_ST\module_WISE.ini"from http://opc.icpdas.com/download.htm frequently.

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



Step 2: The "Search Modules" dialog box pops up.



COM Port:

Specifies which "COM Port" number is search. The default value is 1 and the valid range is from 1 to 255. Please verify the "COM Port" number that the RS-232 or RS-485 network is connected.

Ethernet IO/WISE:

If this field is checked, NAPOPC can search not only the modules communicating via COM port but also "Ethernet IO" and "WISE" 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".

Baud Rate Searching:

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

Naturally, if multiple baud rates are checked, the search will be longer. The computer system must 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 of the "Baud Rate" be checked. Please refer to the above "Baud Rate Searching" section.

Clear All:

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

Address/Start:

Specifies the starting address. The default value is 1 and the valid range is from 1 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 1 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 communicate timeout value for each module. The default value is 500 (equal to 0.5 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.

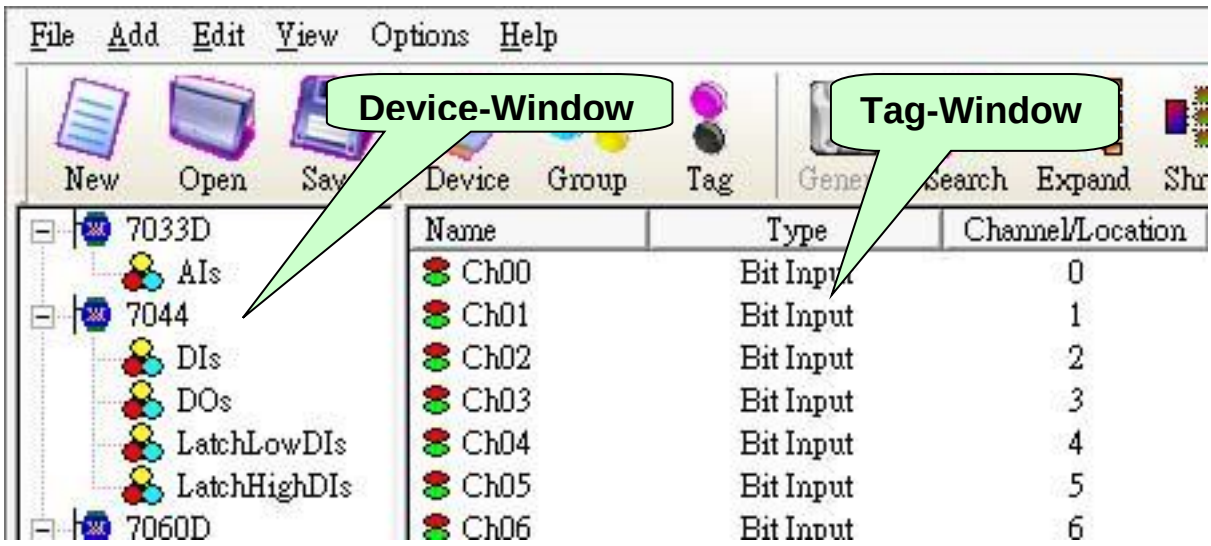
Stop:

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.

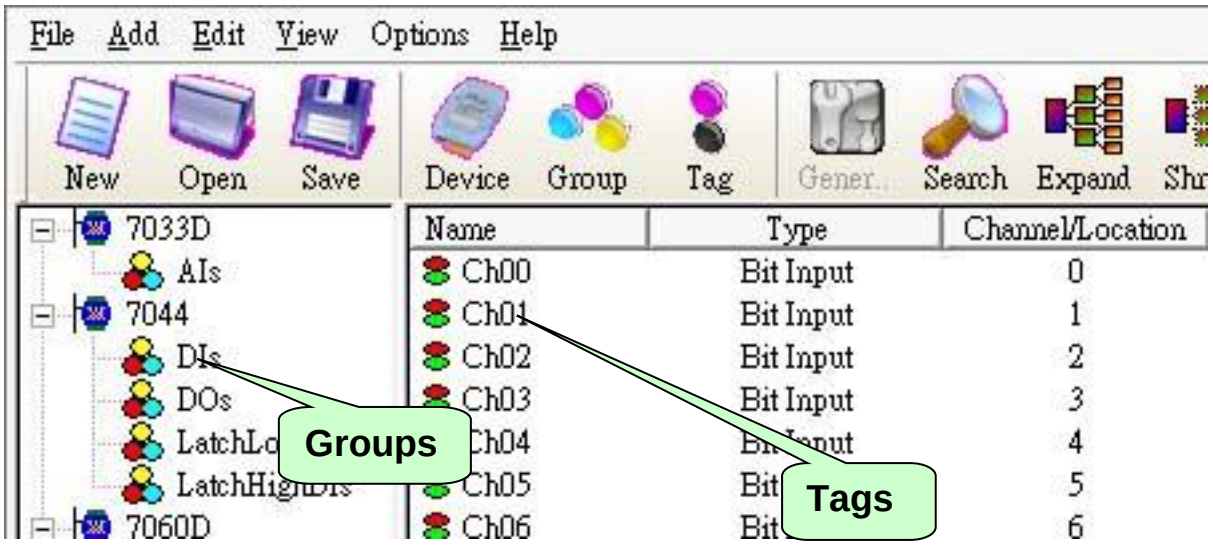
Step 3: After the search, the discovered modules will be listed on the Device-Window (left side). Users can also see the tags on the Tag-Window (right side) generated by the "Search Modules…" function automatically.



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.

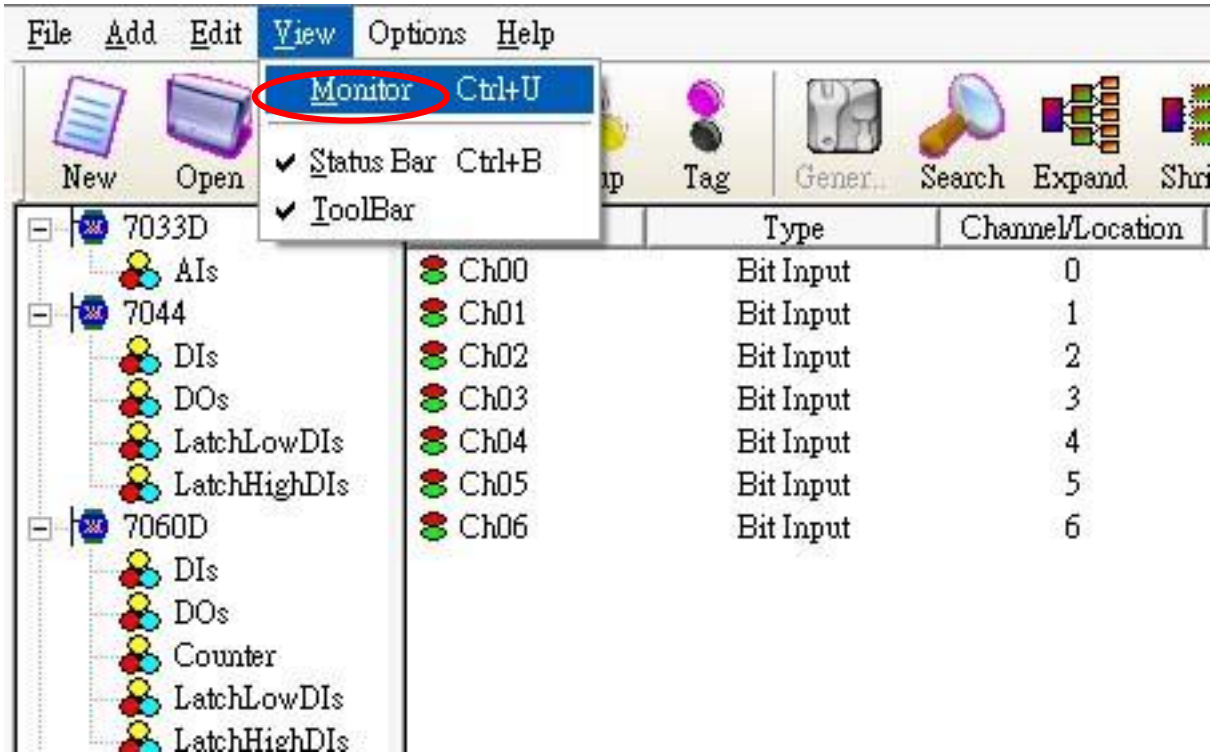
Using the "Digital Input" and "Digital Output" tags to access modules is highly recommended.



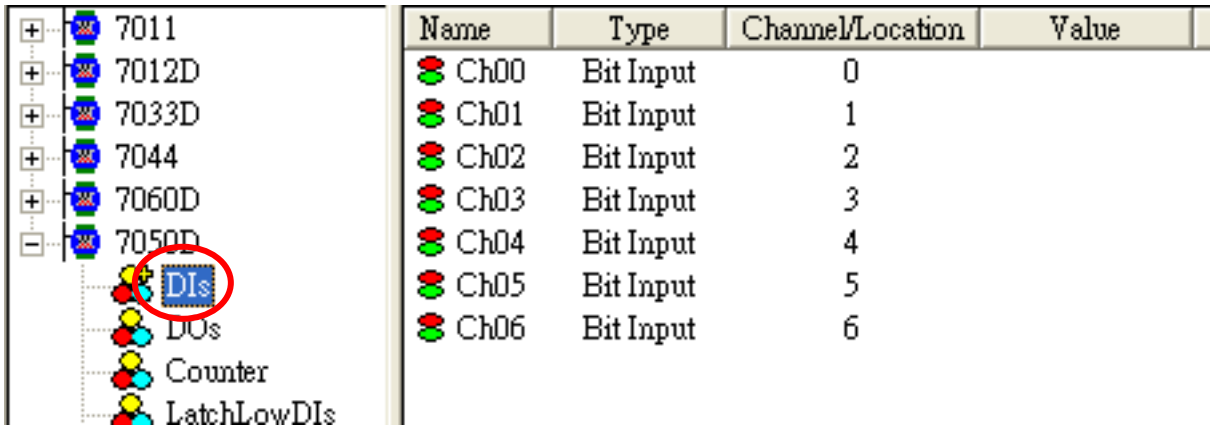
1.5 Monitoring Devices

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

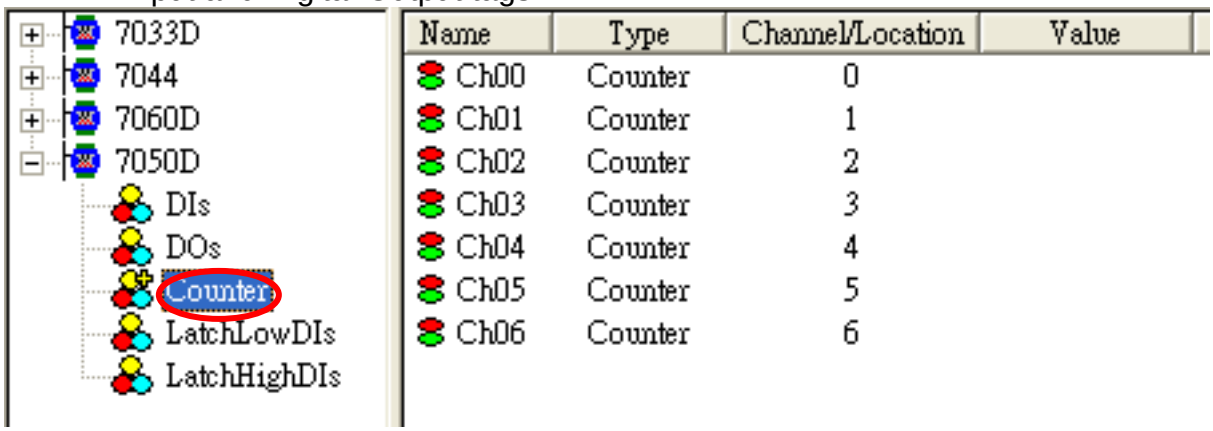
Step 1: Click the "View/ Monitor" menu item or the icon to enable monitor.



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



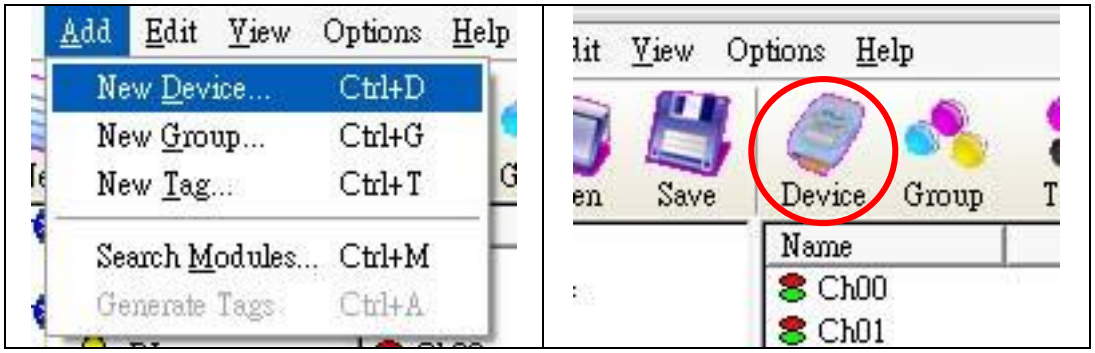
Step 3: Select the "7050D" module on the Device-Window to monitor its own Digital-Input and Digital-Output tags.



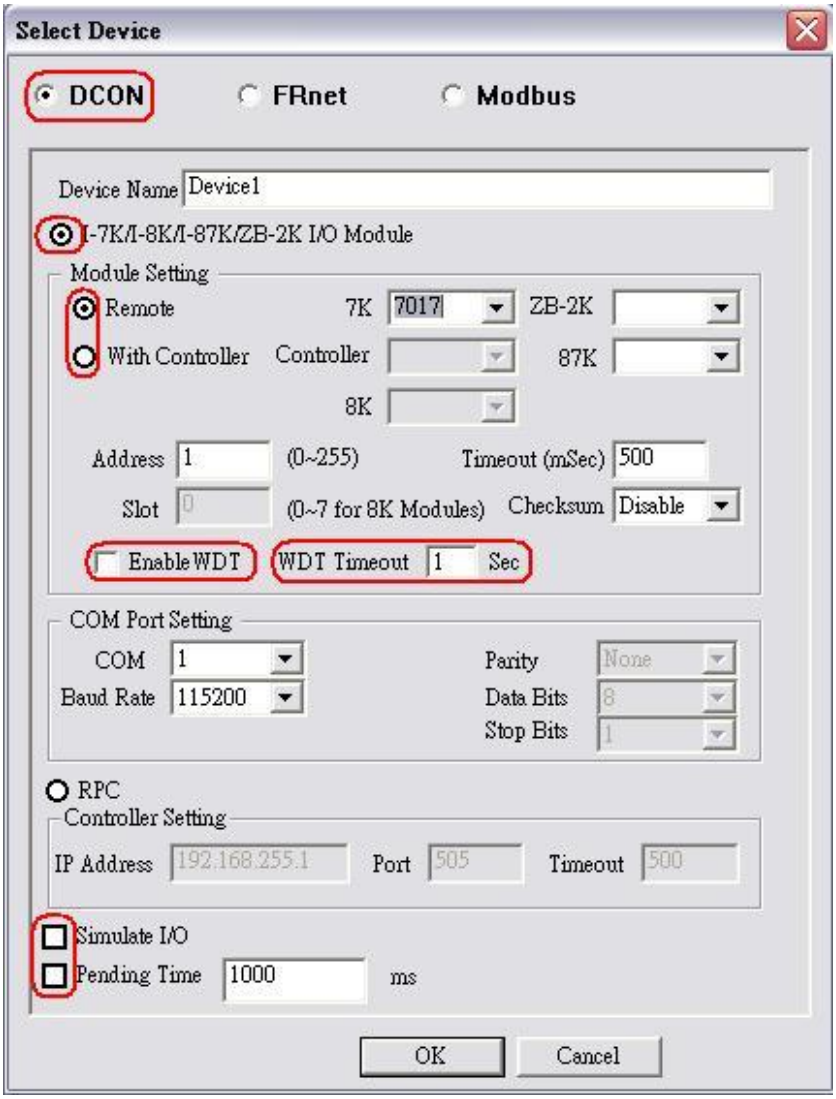
1.6 Adding A New Device

1.6.1 Adding A New I-7K/I-8K/I-87K/ZB-2K I/O Module

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



Step 2: The "Select Device" dialog box pops up.
Step 3: Click on the "DCON" radio button.
Step 4: Click on the "I-7K/I-8K/I-87K/ZB-2K I/O Modules" radio button.
Step 5: Click the "Remote" or "With Controller" radio button.



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.

7K/ 87K/ 8K/ZB-2K Controller Module ID:

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

Address:

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

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

Timeout:

Specifies timeout (Response time) value for this module. The default value is 200 ms. 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.

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.

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.

Enable WDT:

The "Enable WDT" checkbox switches if the module enables watchdog function or not. If the module wants to use watchdog, it should be configured by DCON Utility to enable WDT and setup timeout value.

WDT Timeout:

The "WDT Timeout" range is 1 to 25.5 sec. Please give an appropriate value for each module. And the "Communication Mechanism" has to be "Multi-Thread".

COM Port:

Specifies the COM port to be used. Please verfiy which COM port number that the RS-232 / 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 controller unit 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 failure for this module.

This field is disabled for the 8000 sub-devices. It will use the 8000 main-controller unit 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 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 main controller unit.

Pending Time:

Minimum interval time between two access. To activate this function, NAPOPC_ST 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_ST 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.

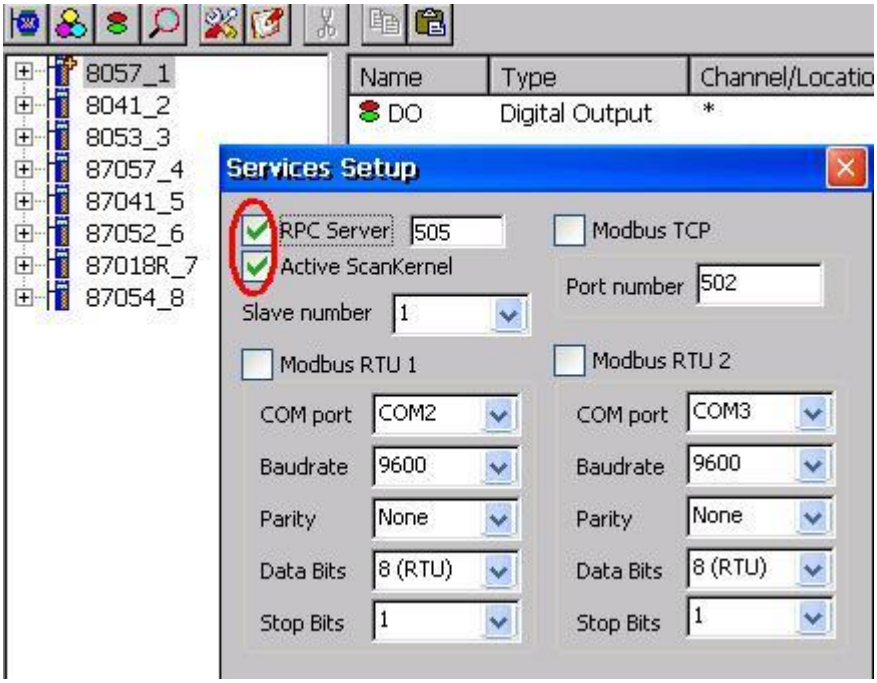
Cancel:

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

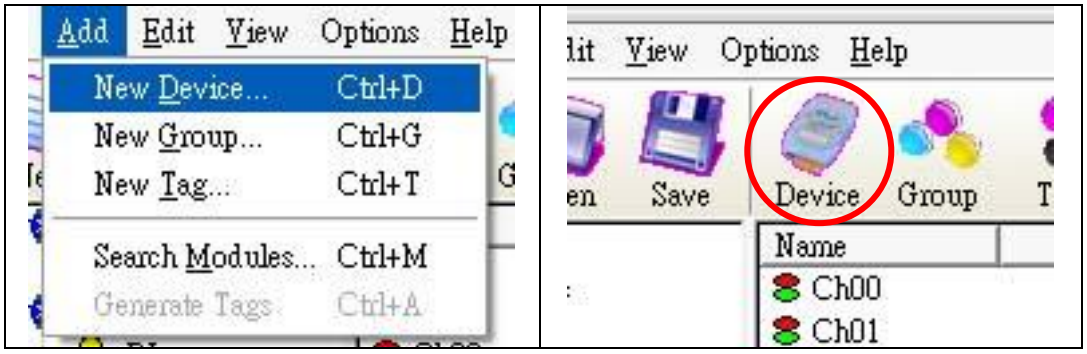
Step 6: Click on the "OK" button to add this new module.

1.6.2 Adding A New RPC Controller

Before adding a new "RPC" controller, please check "RPC Server" and "Active ScanKernel" at "Services Setup" dialog of "NAPOPC_CE5/CE6" in WinCon/ WinPAC/ ViewPAC.



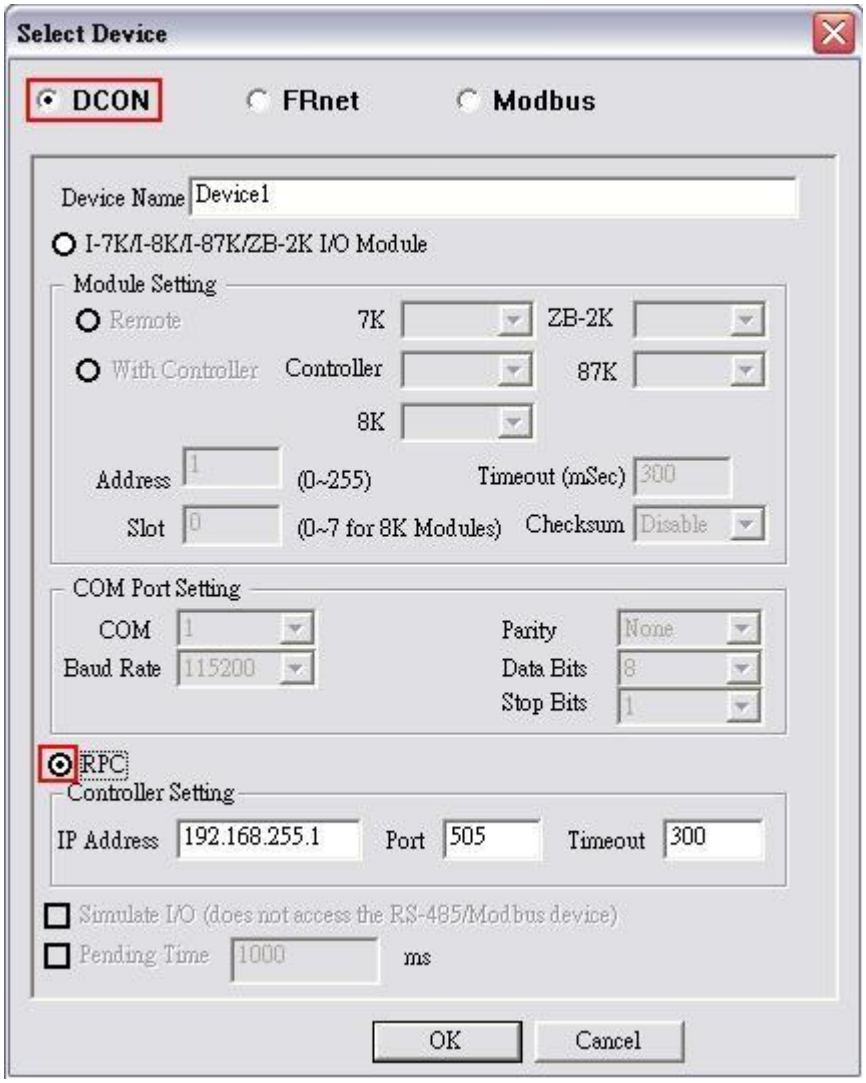
Step 1: Click on the "Add/ New Device…" menu item or the icon to add a new module.



Step 2: The "Select Device" dialog box pops up.

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

Step 4: Click on the "RPC" radio button.



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.

Timeout:

Specifies timeout (Response time) value for this controller. The default value is 200 ms. A smaller timeout value may cause communication failure and a larger timeout value may reduce the performance of the client program.

Port:

You have to set up the value with "505" for communicating with NAPOPC_CE5 or NAPOPC_CE6.

IP Address:

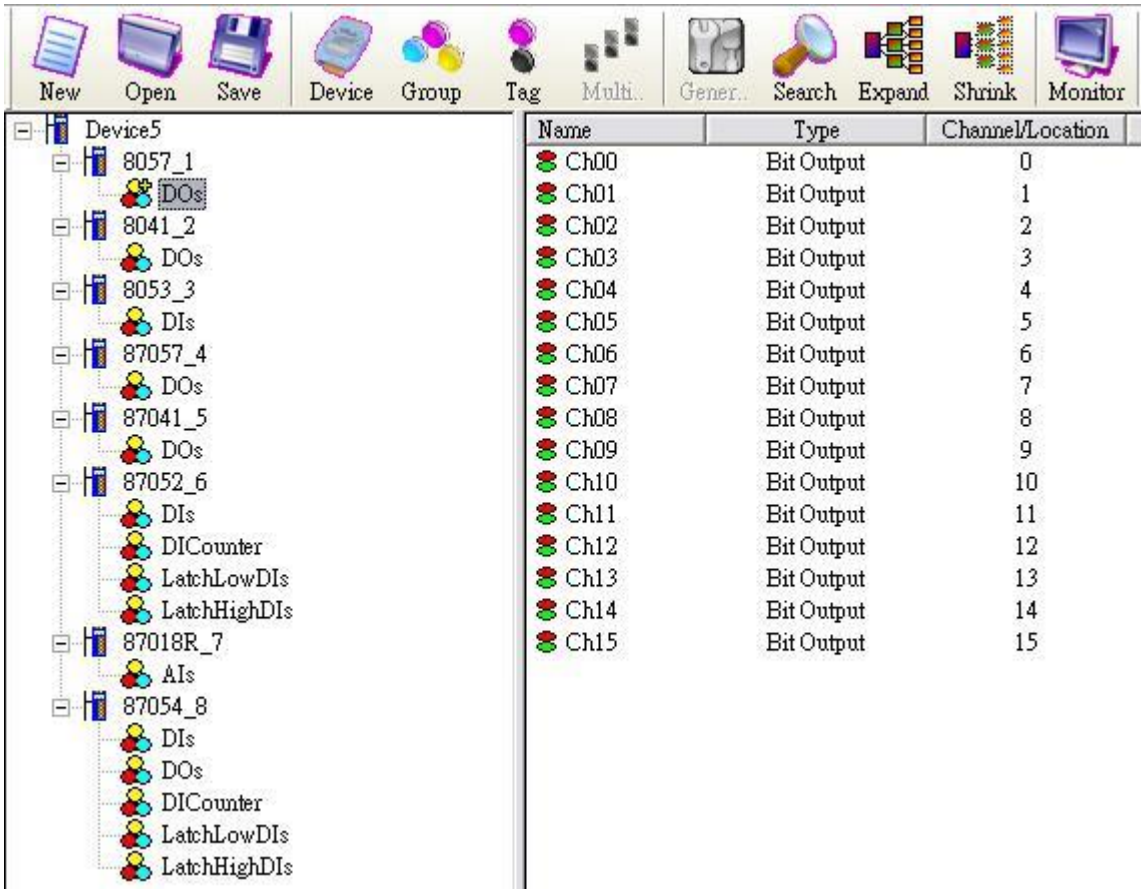
The uniqe IP address of your NAPOPC_CE5 or NAPOPC_CE6.

OK:

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

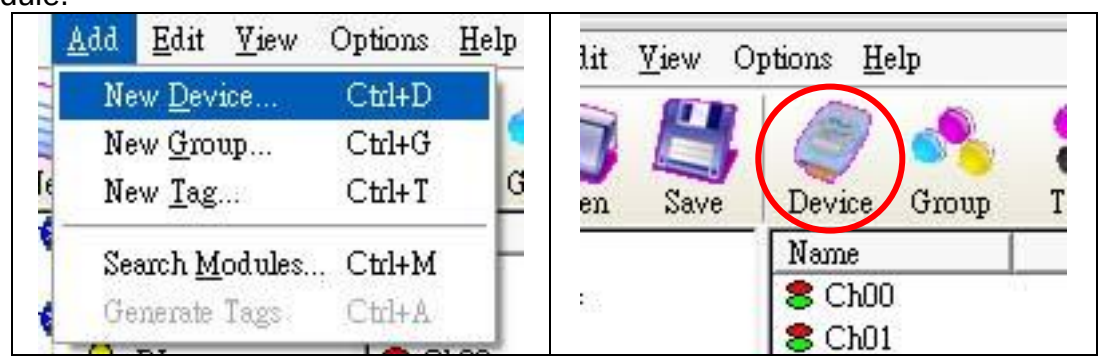
Cancel:
Click on the "Cancel" button to avoid any changes.
Step 5: Click on the "OK" button to add this new device.

Step 6: After clicking on the "OK" button, NAPOPC_XPE will automatically synchronize and generate the modules of "NAPOPC_CE5/CE6".

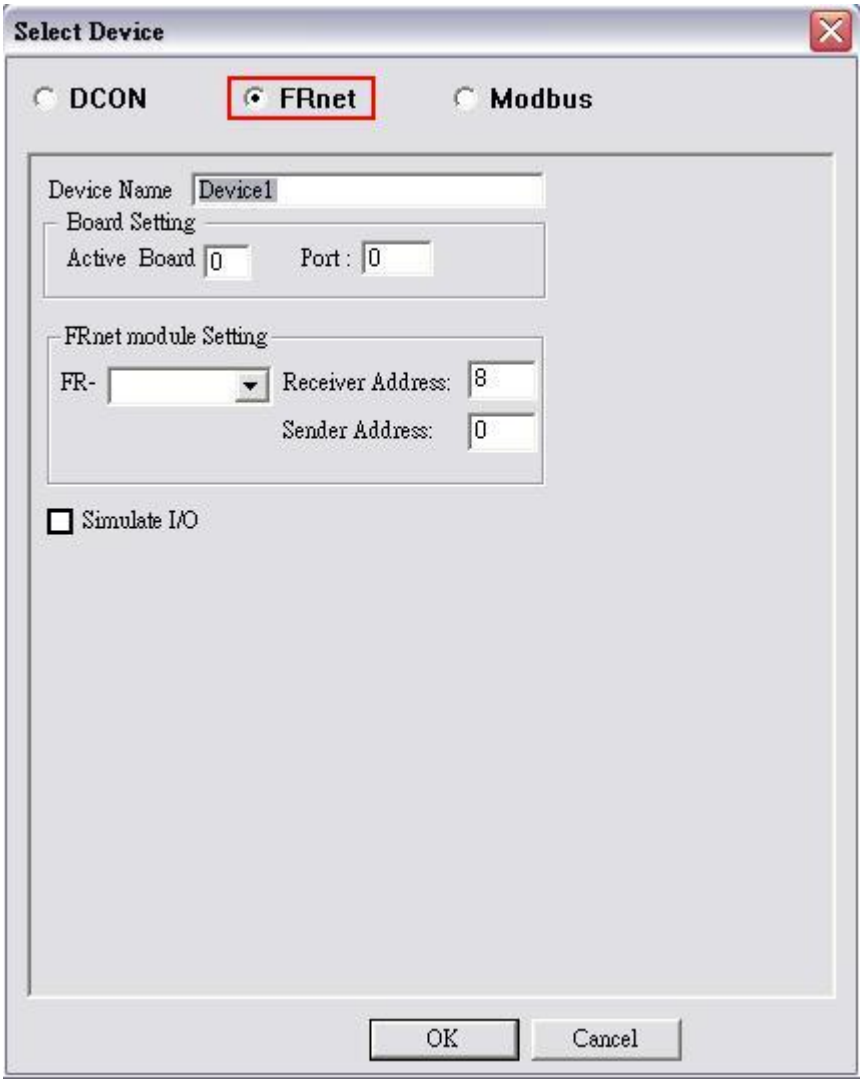


1.6.3 Adding A New FRnet I/O

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



Step 2: The "Select Device" dialog box pops up.
Step 3: Click on the "FRnet" radio button.



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.

Active Board:

Board number that you want to active, start from number 0(0 the First FRB Board, 1 the Second FRB Board). Please refer to the FRB Board manual for more information.

Port:

The "Port" indicates the port number(0 or 1) of FRB Board. Please refer to the FRB Board manual for more information.

FRnet Module ID:

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:

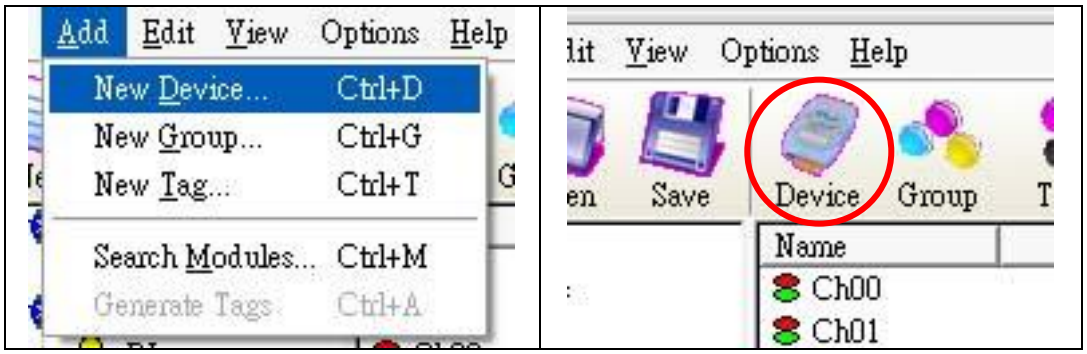
FRnet communication needs correct hardware configurations for the sender address (SA) and receiver address (RA) on the host controller and the remote module in the network. Please refer to the FRnet manual for more information..

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.

1.6.4 Adding A New Modbus TCP Controller

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



Step 2: The "Select Device" dialog box pops up.

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

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

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 247.

Timeout:

Specifies timeout (Response time) value for this controller. The default value is 200 ms. A smaller timeout value may cause communication failure.

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.

Request Tag Number:

The "Request Tag Number" sets tag value numbers that each command will get from device. For ISaGRAF, it should less than 124 for coil and register. For 7188MTCP, it should less than 498 for coil and 127 for register. The default numbers are both 122. (For Modbus standard, it can't greater than 2000 for coil and 127 for register.

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 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_ST 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_ST 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.

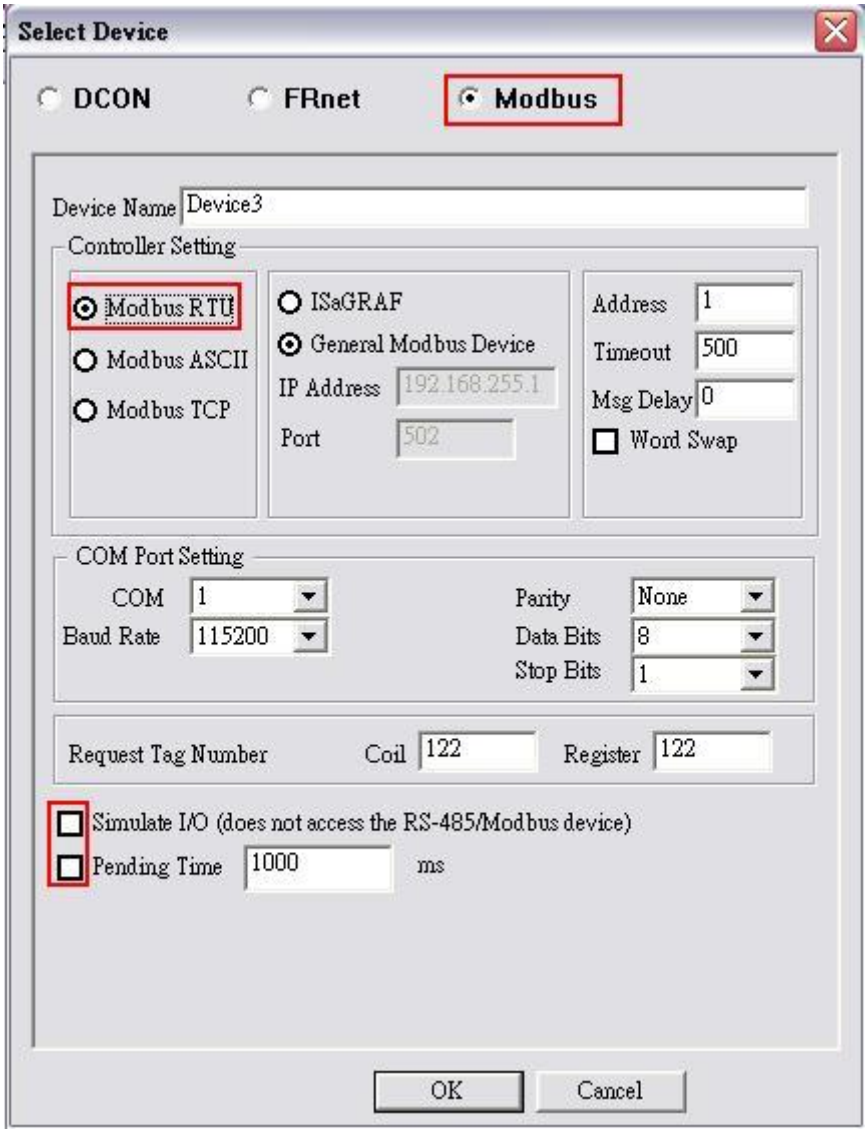
1.6.5 Adding A New Modbus RTU Controller

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

Step 2: The "Select Device" dialog box pops up.

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

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



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 247.

Timeout:

Specifies timeout (Response time) value for this controller. The default value is 200 ms. 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-232 / 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

Request Tag Number:

The "Request Tag Number" sets tag value numbers that each command will get from device. For ISaGRAF, it should less than 124 for coil and register. For 7188MTCP, it should less than 498 for coil and 127 for register. The default numbers are both 122. (For Modbus standard, it can't greater than 2000 for coil and 127 for register.

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 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_ST 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_ST 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.

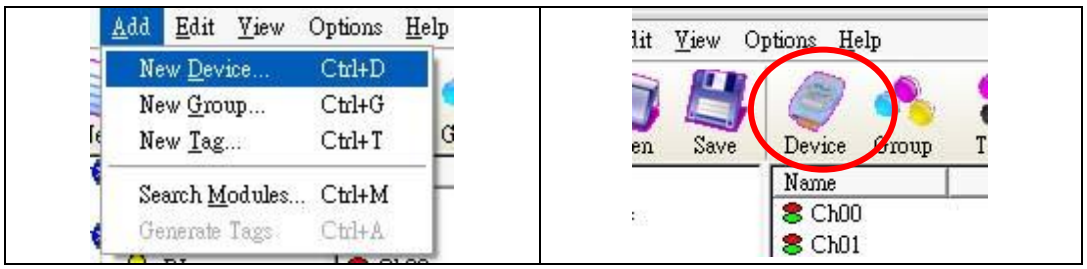
Cancel:

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

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

1.6.6 Adding A New Modbus ASCII Controller

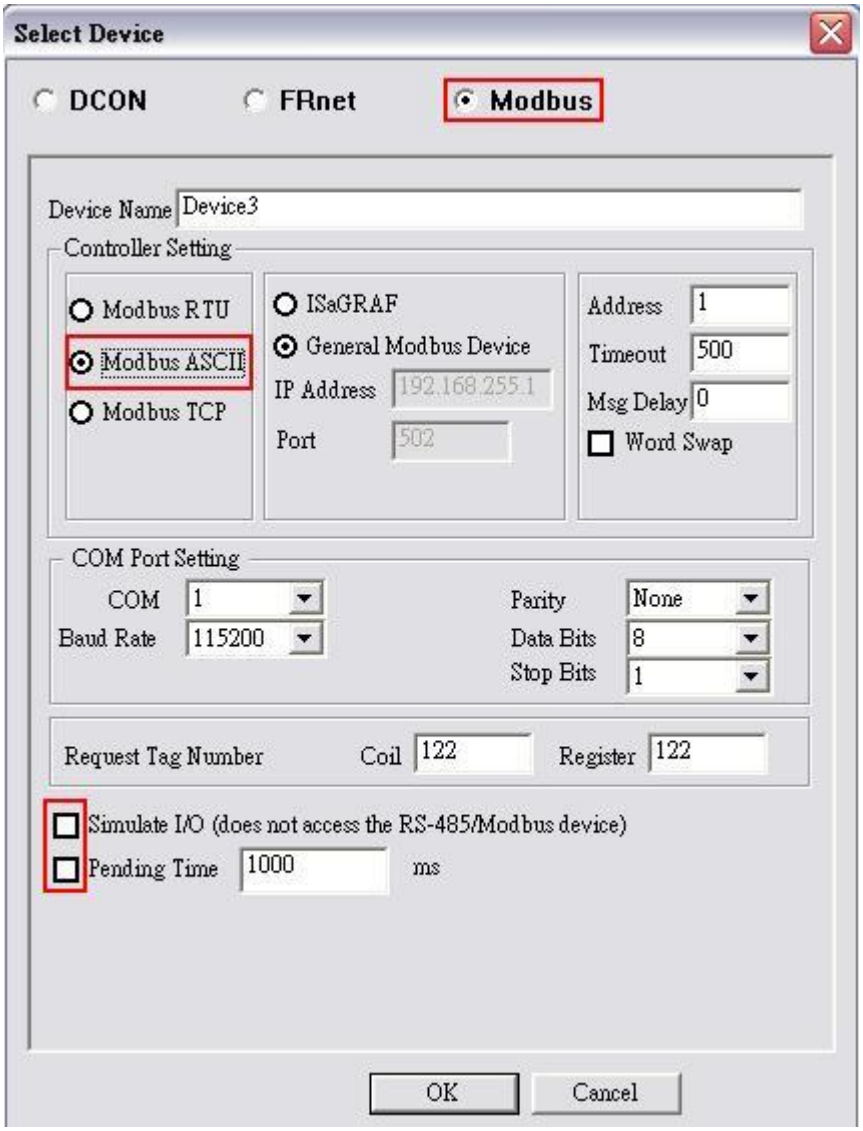
Step 1: Click on the "Add/ New Device…" menu item or the icon to add a new module.



Step 2: The "Select Device" dialog box pops up.

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

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



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 247.

Timeout:

Specifies timeout (Response time) value for this controller. The default value is 200 ms. 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-232 / 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

Request Tag Number:

The "Request Tag Number" sets tag value numbers that each command will get from device. For ISaGRAF, it should less than 124 for coil and register. For 7188MTCP, it should less than 498 for coil and 127 for register. The default numbers are both 122. (For Modbus standard, it can't greater than 2000 for coil and 127 for register.

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 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_ST 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_ST 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:

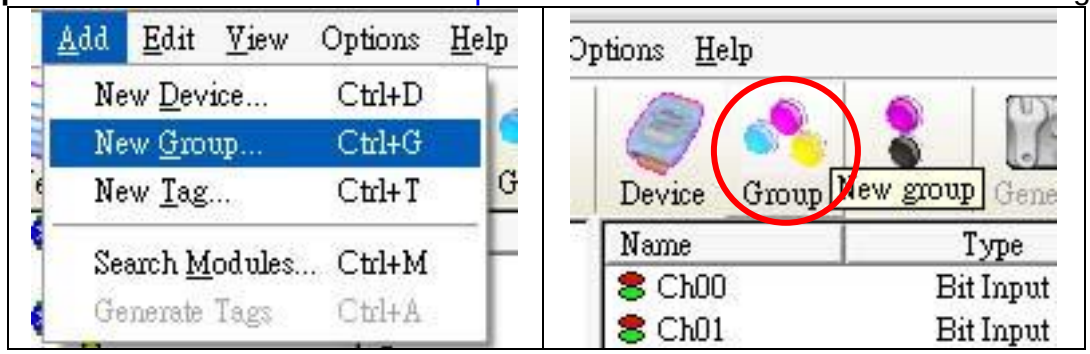
Click on the "Cancel" button to avoid any changes.

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

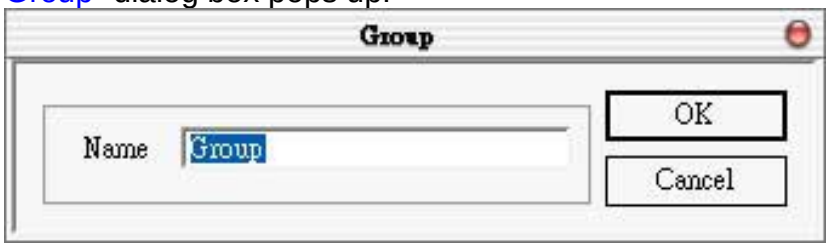
1.7 Adding A New Group

If the device you add is "RPC", you do not need to add groups manually. NAPOPC_ST will automatically synchronize and generate the modules of "RPC".

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



Step 2: The "Group" dialog box pops up.



Name:

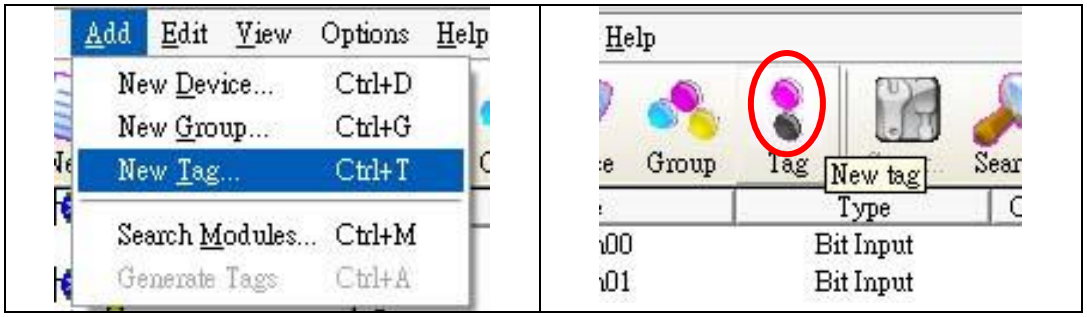
A "Group Name" may have any name, but avoid names with spaces or punctuation such as "|!.,". The "Group Name" must not be used twice. A group can be defined as a subdirectory containing one or more tags. A device may have many subgroups of tags. All tags belong to their module when they are scanned to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

1.8 Adding A New Tag

If the device you add is "RPC", you do not need to add tags manually. NAPOPC_ST will automatically synchronize and generate the modules of "RPC".

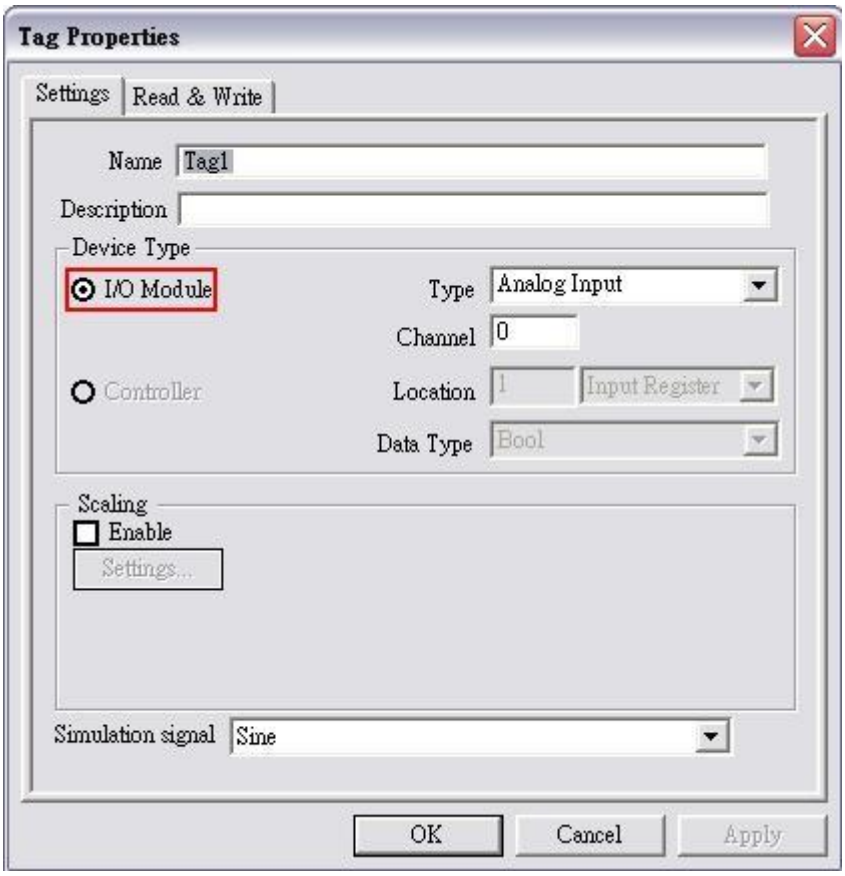
1.8.1 Adding A New Tag For I/O Module

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



Step 2: The "Tag Properties" dialog box pops up.

Step 3: Choose the "Settings" page. Because the tag belongs to the module-type device, the "I/O Modules" radio button is active.



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.

Description:

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

Type:

Specifies the command to be used for this tag. Different modules support different commands. For commands, please refer to a "MODULES.HTM" file in \\ICPDAS\NAPOPC_ST folder

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 signal:

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.

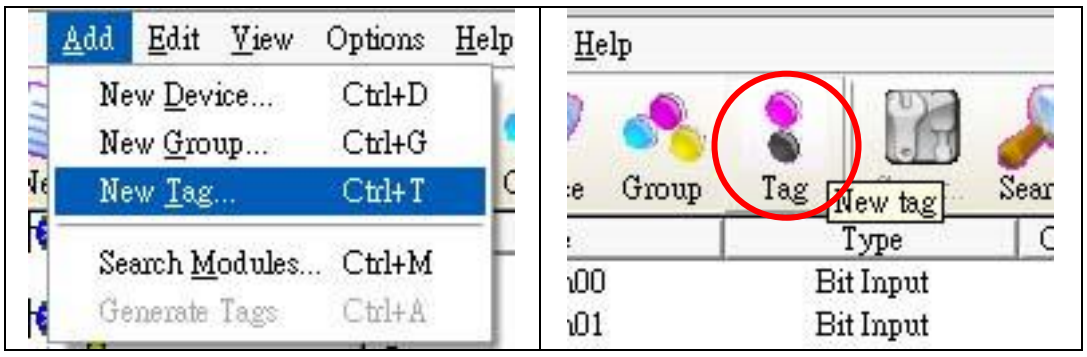
Settings:

Click on this button to set the scaling feature.

For more information, please refer to the section "1.7.3 Scaling Settings".

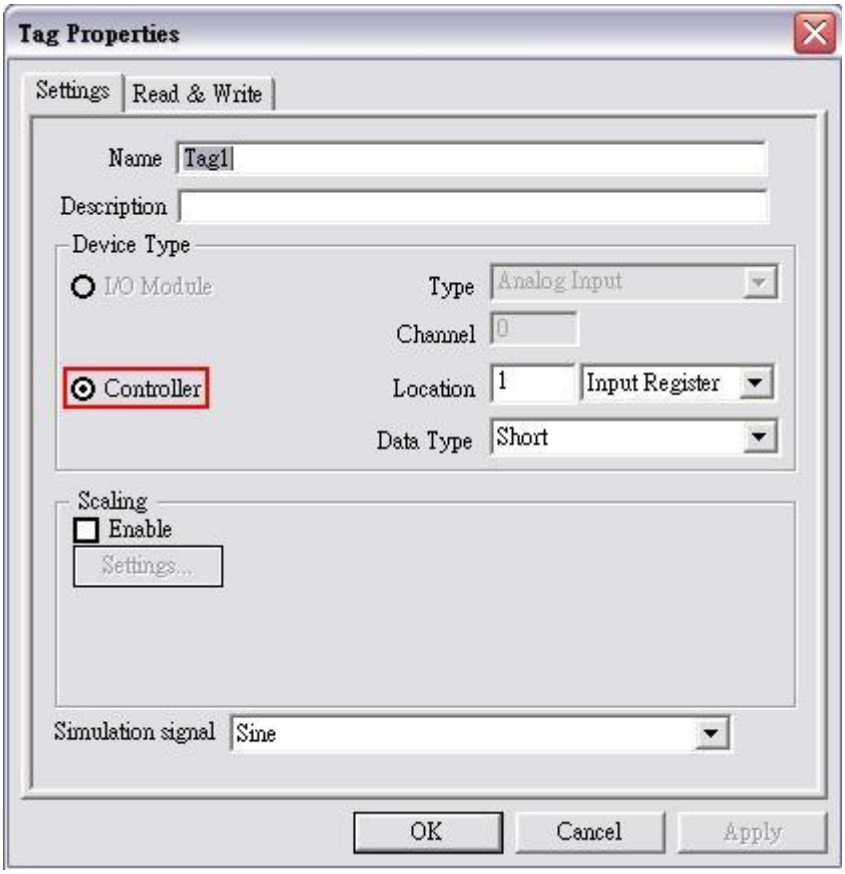
1.8.2 Adding A New Tag For Controller

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



Step 2: The "Tag Properties" dialog box pops up.

Step 3: Choose the "Settings" page. Because the tag belongs to the controller-type device, the "Controller" radio button is active.



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.

Description:

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

Data:

Specifies the data type of this tag which's location type is "Input Register" or "Output Register". NAPOPC_ST Server support 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".

Location:

Specifies the tag address. It must be the same with the the variable address in the controller. Besides, you have to choose the location type. After you choose the location number, there are four location types you can

choose.They are "Input Coil", "Output Coil", "Input Register", and "Output Register". When you monitor controller device(see 1.3 Monitoring Device), the "Channel/Location" field will show a value according to the location and location type as below.

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

Simulation signal:

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.

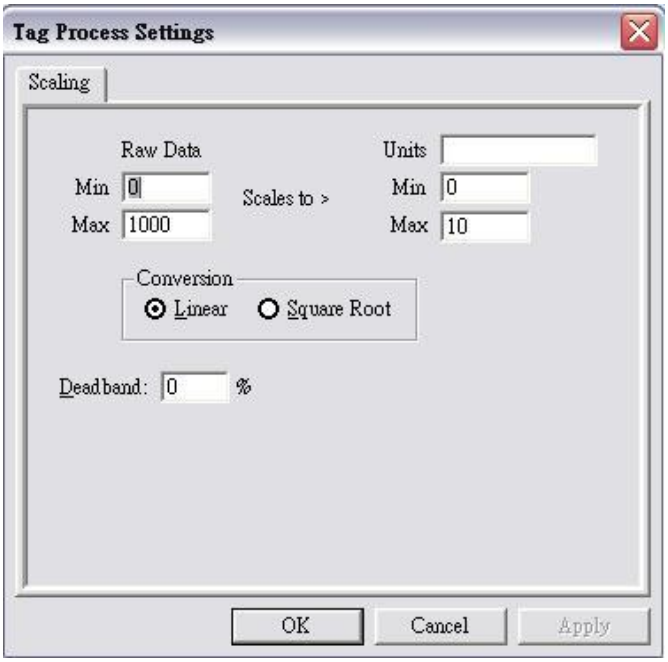
Settings:

Click on this button to set the scaling feature.

For more information, please refer to the section "1.7.3 Scaling Settings".

1.8.3 Scaling Settings

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



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.
For more information, please refer to the "4.5.1.6 Percent Deadband" section in the "OPCDA20_Cust.PDF" manual, page 68.

OK:

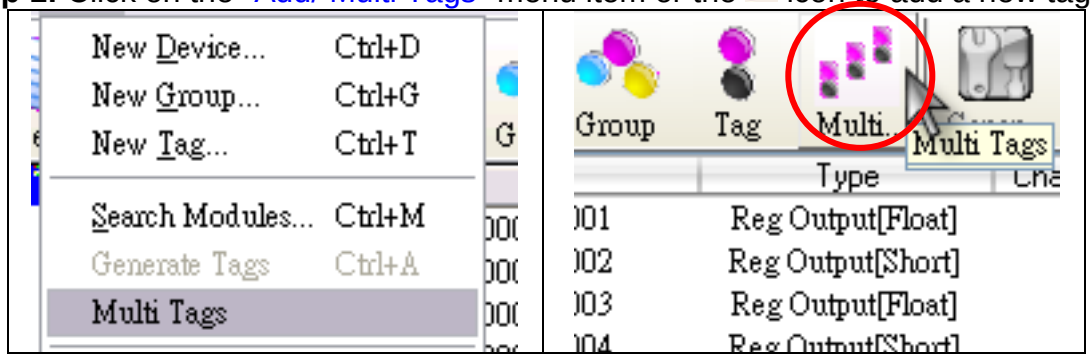
Click the "OK" button to save these settings.

Cancel:

Click the "Cancel" button to avoid any changes.

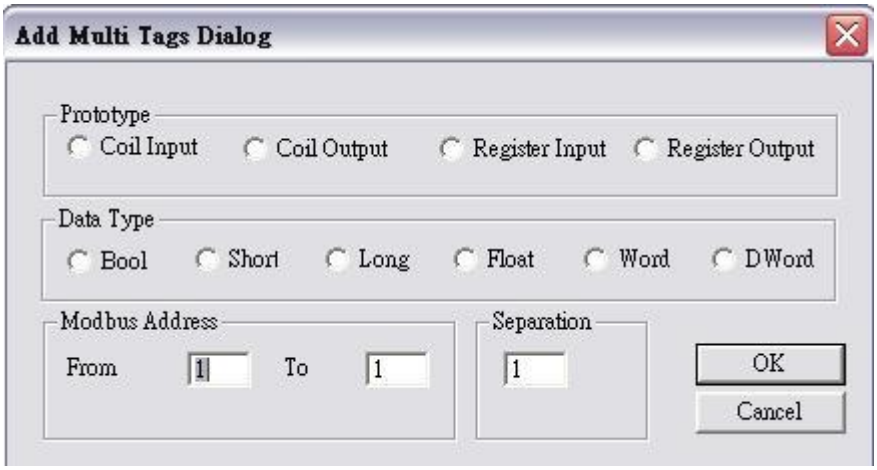
1.9 Adding Multi Tags For Modbus Device

This function only work when the device's protocol is Modbus.
Step 1: Click on the "Add/ Multi Tags" menu item or the icon to add a new tag.



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

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



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:

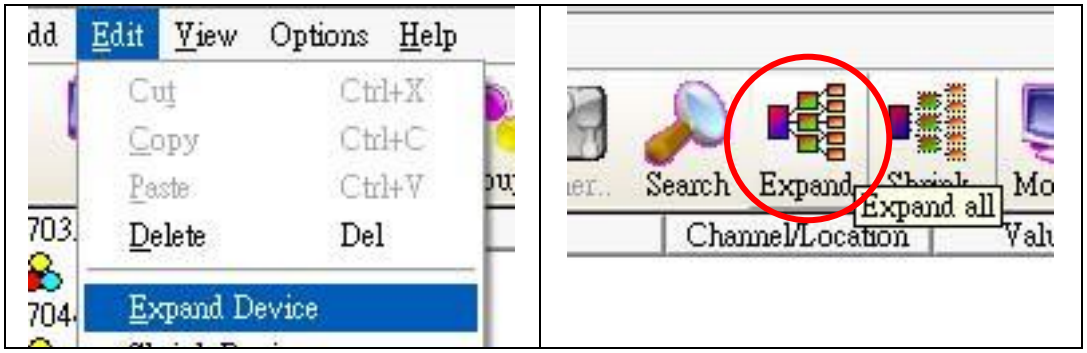
Click on the "OK" button to add the new tag setting.

Cancel:

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

1.10 Expand/ Shrink Devices

Click on the "Edit/ Expand device (Shrink device)" menu item or the icon to expand(shrink) all devices..



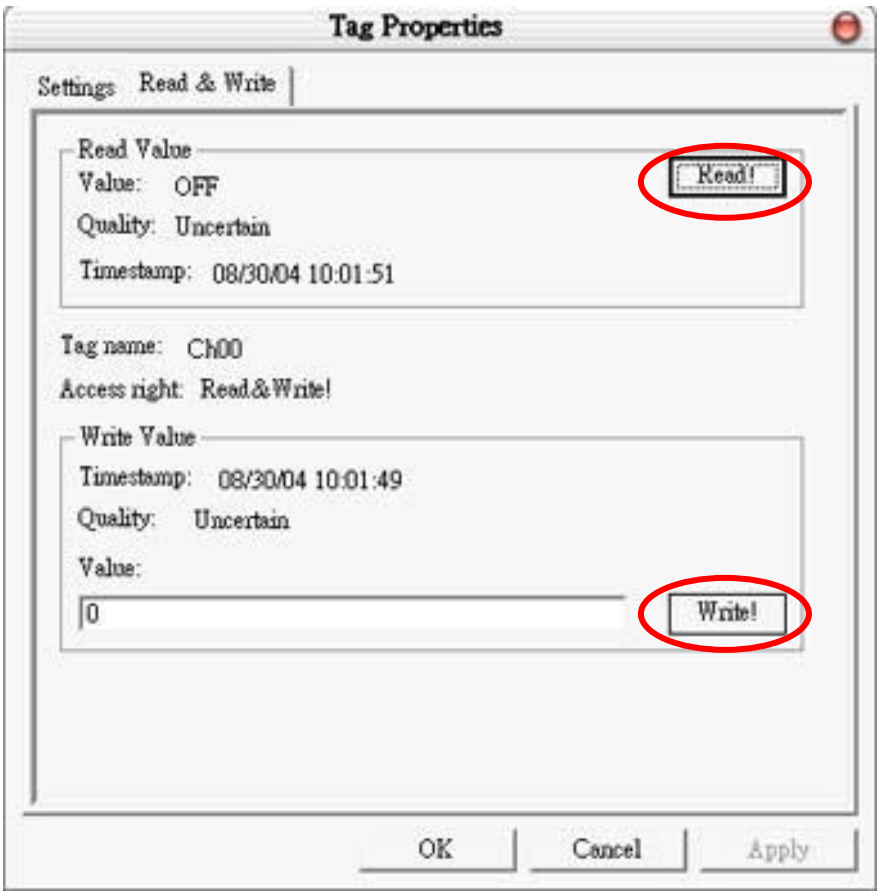
1.11 Read/Write Tags

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.



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 quality shows "Bad". And the other situation is "Uncertain". Maybe you do not click the "View/ Monitor" menu item to enable monitor etc.

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:

Two kinds of aceess right, "Read Only!" and "Read&Write!", would be shown. The access right depends on what kind of tag property it is. Please refer to the "1.7 Adding A New Tag"

Write Value/Timestamp:

It shows the time, when you write the tag.

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". Maybe you do not click the "View/ Monitor" menu item to enable monitor etc.

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".

1.12 Editing A Device/Group/Tag Properties

To edit a existing Device(/Group/Tag), just select the Device(/Group/Tag) and right click the mouse button. Then select the "Properties…" option.

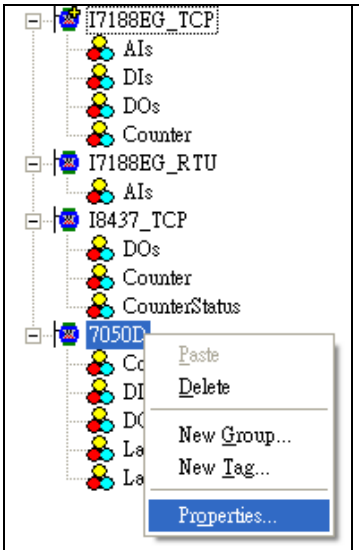


Figure 1-11-1.

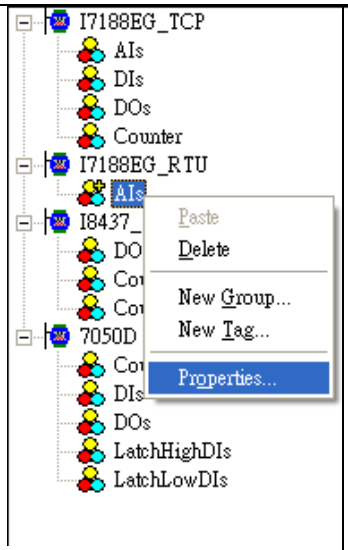


Figure 1-11-2.

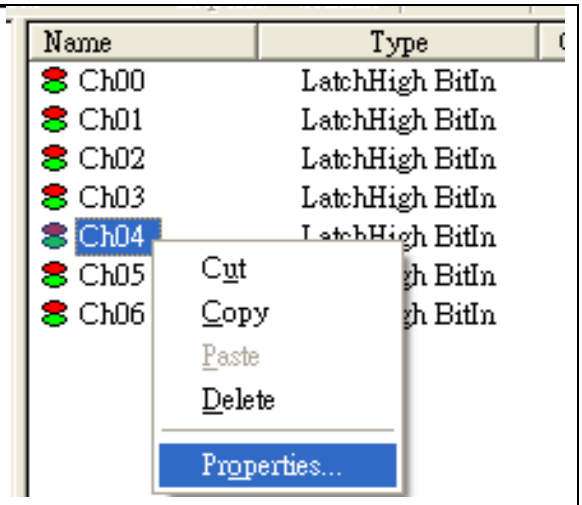


Figure 1-11-3.

Figure 1-11-1. Select a device and right click the mouse button. Then select the "Properties.." option.

Figure 1-11-2. Select a group and right click the mouse button. Then select the "Properties.." option.

Figure 1-11-3. Select a tag and right click the mouse button. Then select the "Properties.." option.

1.13 Deleting A Device/Group/Tag

To delete a existing Device/Group/Tag, just select the Device(/Group/Tag) and right click the mouse button. Then select the "Delete…" option or the icon.

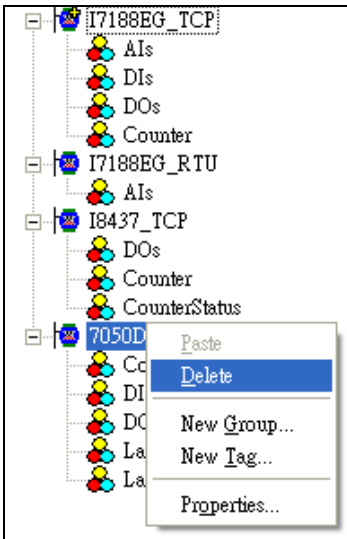


Figure 1-12-1.

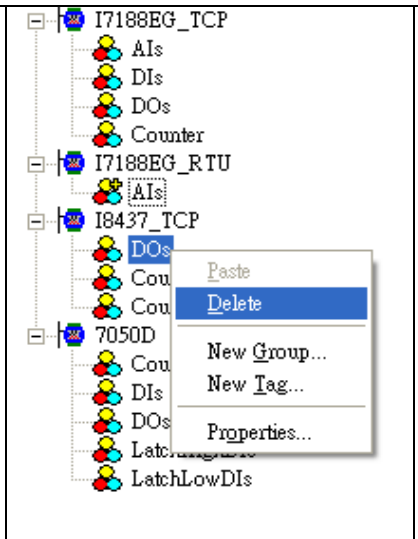


Figure 1-12-2.

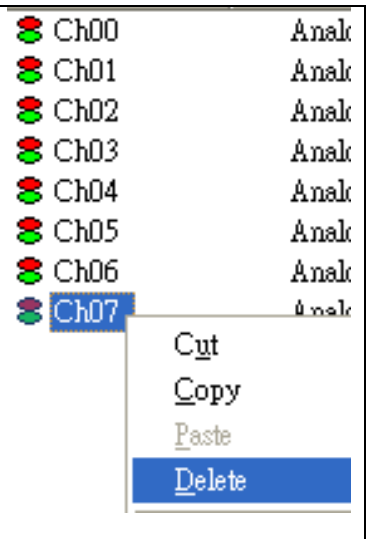


Figure 1-12-3.

Figure 1-12-1. Select a device and right click the mouse button. Then select the "Delete" option.

Figure 1-12-2. Select a group and right click the mouse button. Then select the "Delete" option.

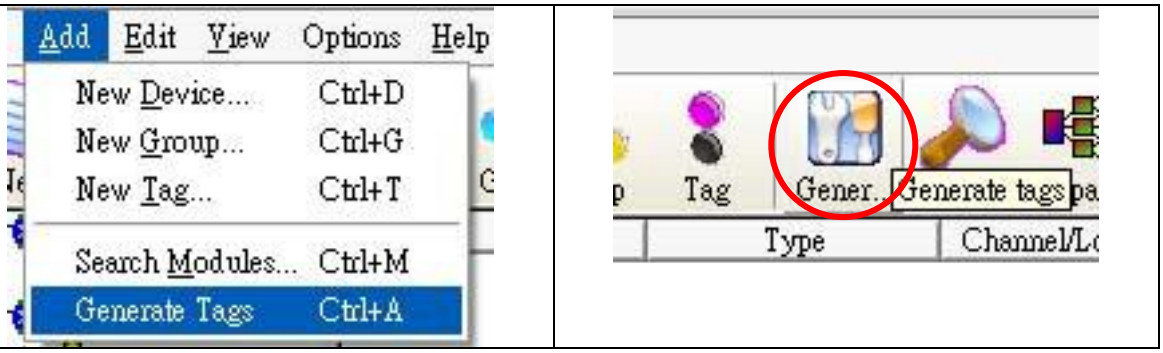
Figure 1-12-3. Select a tag and right click the mouse button. Then select the "Delete" option.

1.14 Generating Tags

This function lets you easily test the OPC server 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.

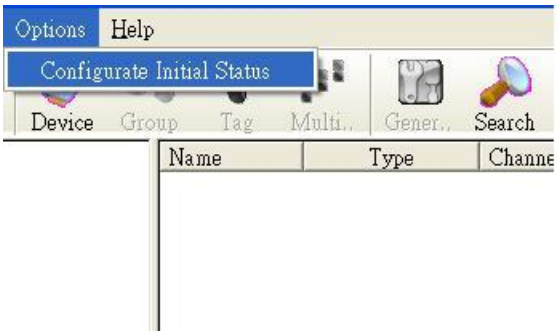
Step 2: Click on the "Add/ Generate Tags" menu item or the icon to generate tags.

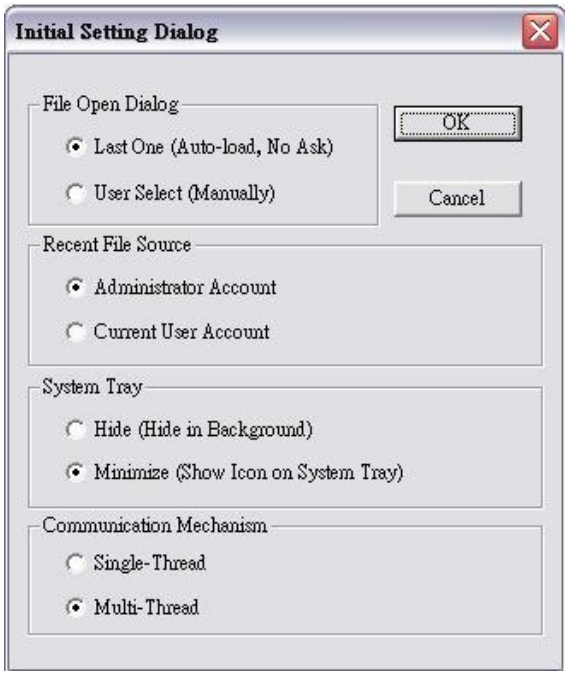


1.15 Configurate Initial Status

This function lets you configurate initial status of NAPOPC_ST server when client softwares connect to it. The ""Recent File Source" options lets user save related information in registry under "Administrator account" or "Current user account". Some DCOM application needs to adjust this option for specific scenario. The "File Open Dialog" option lets user load a file automatically or manually. The "System Tray" option can hide or minimize program of NAPOPC_ST server. And "Communication Mechanism" option lets user define the communication behavior of NAPOPC_ST. Generally, "Multi-Thread" is the best choice for high performace. However, for some particular OPC clients which can not work smoothly under "Multi-Thread" communication, user can choose "Single-Thread" instead. This function, Configurate Initial Status, can be selected only when "Monitor" function isn't running.

Step 1: Click on the "Options/ Configurate Initial Status" menu item to open "Initial Setting Dialog".





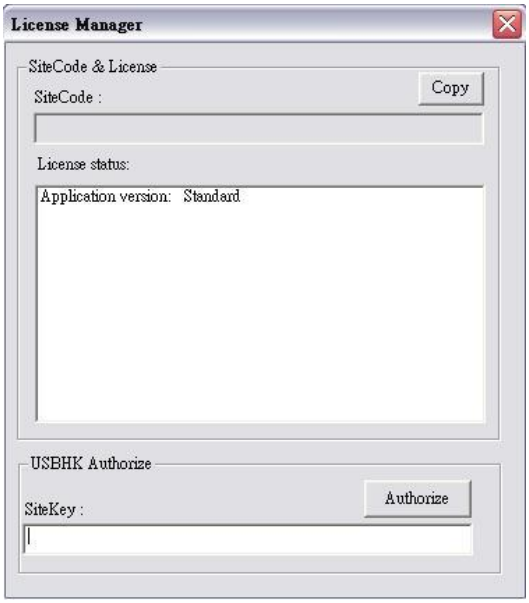
Note: "Communication Mechanism" has to be "Multi-Thread" if DCON device enables WDT function.

1.16 License Manager

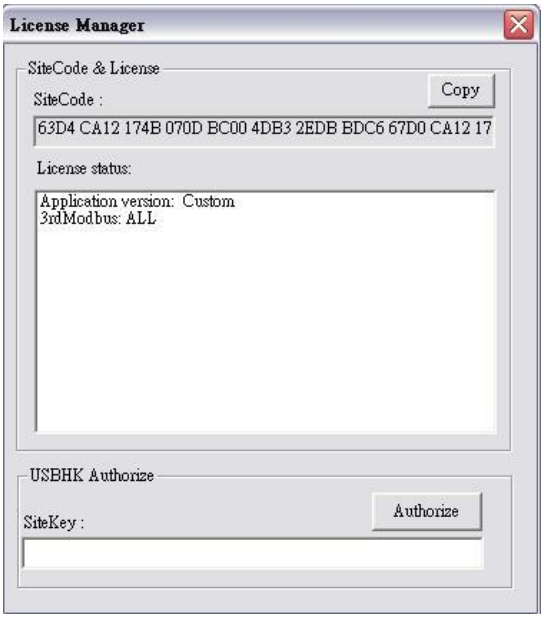
NAPOPC_ST DA Server v3.30 or later version provides "License Manager" function to manage the NAPOPC_ST license. You can purchase USB hardkey to enhance the functionality of NAPOPC_ST DA Server.

Step 1: Click on the "Options/ License Manager" menu item to open "License Manager Dialog".

Step 2: If there is no USB hardkey plugged in, license status shows "Application version: Standard".



Step 3: If there is a USB hardkey plugged in, license status shows how many licenses it has. For this example, it shows "3rdModbus: ALL" which means NAPOPC_ST DA Server allows using third party modbus devices.



Step 4: If you have already had USB hardkey and you want to upgrade it, you need to press "Copy" to copy SiteCode to your distributor. Your distributor will give you a SiteKey to fill it in SiteKey field and press "Authorize" to upgrade USB hardkey.

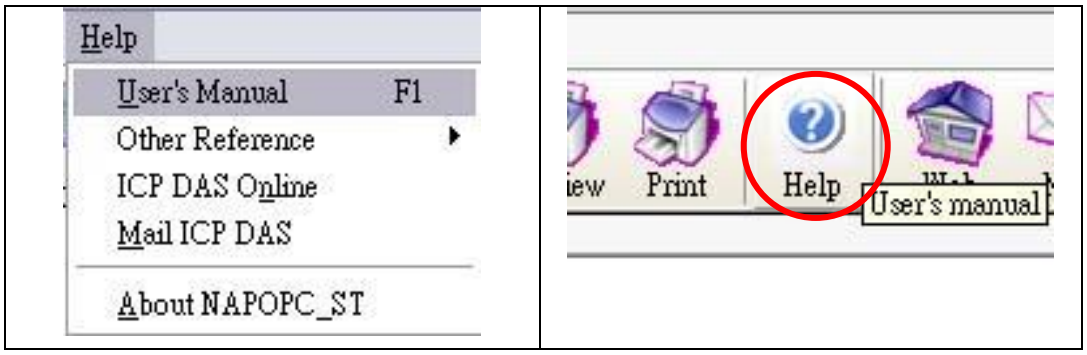
NOTE:

You can plug two identical license Hardkeys in one PC. NAPOPC will execute redundancy when one Hardkey takes place failure unexpectedly.

1.17 Help

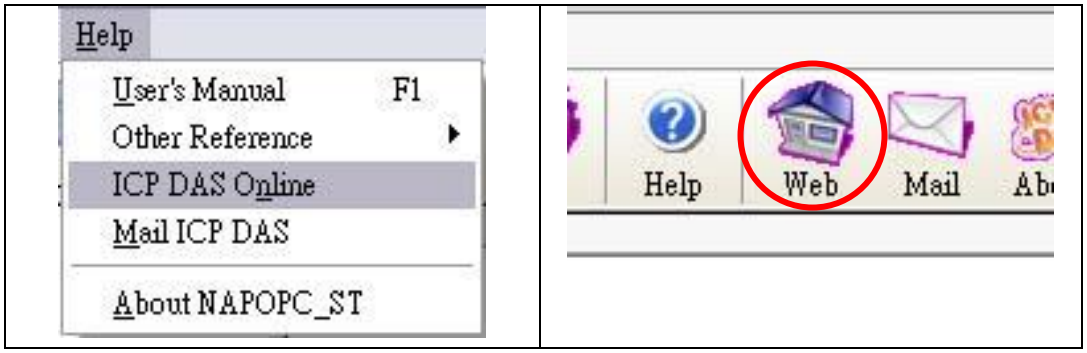
Refer to the user's manual by Checking the "Help/ User's Manual" menu item. All PDF formatted files are best view using Acrobat Reader 5 or newer. You can install it from our CD or download a free copy from Adobe's Web Site.

Clicking on the "Help/ User's Manual" menu item or the icon refer to the user's manual.

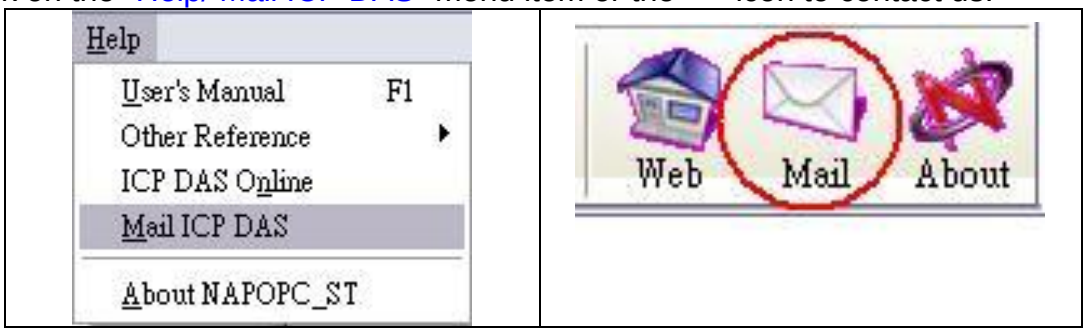


Visit our web by checking the "Help/ ICP DAS Online" menu item or contact us by checking the "Help/ Mail ICP DAS" menu item.

Click on the "Help/ ICP DAS Online" menu item or the icon to browse our web.



Click on the "Help/ Mail ICP DAS" menu item or the icon to contact us.



1.18 About

Click on the "Help/ About NAPOPC_ST" menu item or the icon to see the "About NAPOPC_ST DA Server" dialog box. It shows the version number.

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

Step 2: The "About NAPOPC_ST DA Server" window pops up.

2 Quick Start

Please follow these steps:

[Configure NAPOPC_ST Server]

1. Wire Modules or Controllers.
   Wire modules in the RS-232 / RS-485 network. (Refer to "\CD \Napdos\7000\manual \GetStart.PDF" manual.)
   Wire controllers to your PC.

2. Configure Modules or Controllers.
   Use DCON Utility to set modules. (Refer to "\CD \Napdos\7000\manual \GetStart.PDF" manual.)
   Use ISaGRAF to configure the I-7188EG/XG or I-8xx7.

3. Install the OPC server.
   Install the NAPOPC_ST on your computer.

4. Run the OPC Server.
   Launch the OPC server by executing "
   C:\ICPDAS\NAPOPC_ST\NAPOPCSvr_ST.exe"

5. Search Modules.
   Refer to the "1.3 Search Modules…" section to search modules in the RS-485 network.

6. Add a new controller
   Refer to the "1.5 Adding A New Device" section to add a new modbus RTU or modbus TCP controller.

7. Save Configuration.
   Save the configuration by clicking "File/Save" menu item.

8. Close OPC server.
   Close OPC Server by clicking "File/Exit" menu item.

[Connect to NAPOPC_ST Server]

1. Connect to OPC server.
   Run your client program and connect to our OPC server by linking "NAPOPC.Svr" or " NAPOPC.Svr.1".
   (Please refer to user's manual of your client software provided by your vendor.)
   This forces the system to run the OPC server automatically in background.

3 Connect To OPC Server

The OPC is defined by the OPC Foundation, and any client program supporting OPC can connect to OPC server (for example: Lab VIEW v5.0 and WIZCON 7.51). Any computer language supporting the COM mechanism can also connect to the OPC server directly through the COM interface.

The first section shows you how to optimize your communication. And the following sections show you how to connect to OPC server by using client program provided by Factory Soft, Inc, Lab VIEW, Server Explorer provided by National Instruments, WIZCON, iFix, InduSoft and CitectSCADA. To connect to OPC server by other OPC client, please refer to your OPC client user's manual.

3.1 Optimize Your Communication

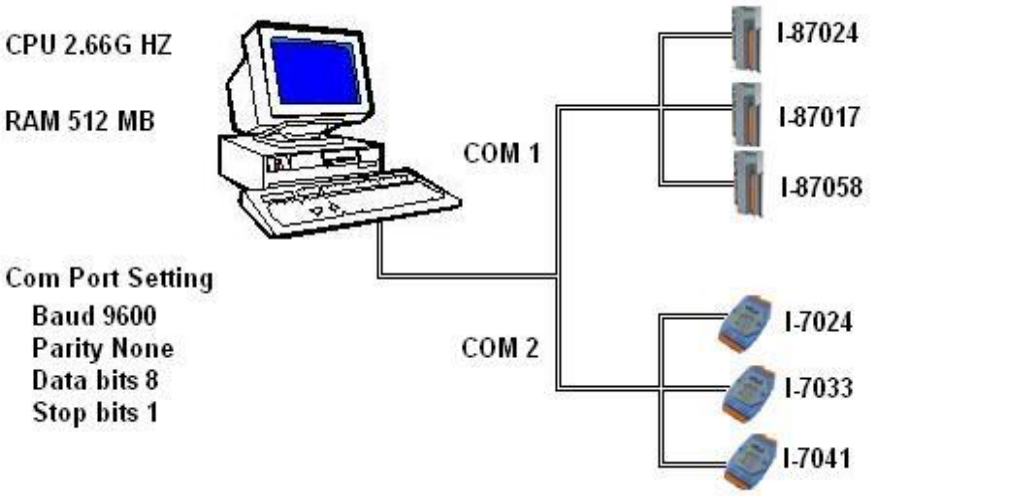
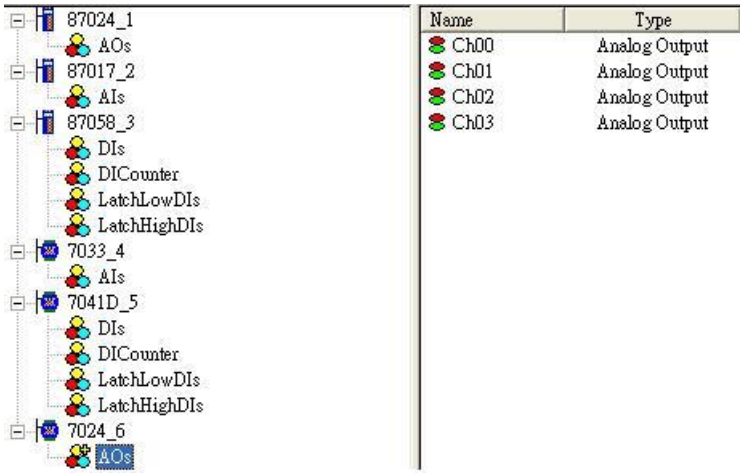
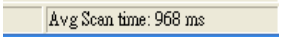


Figure 3.1.1 Communication architecture of I/O modules

Figure 3.1.1 is a figure of communication architecture of I/O modules. NAPOPC_ST server accesses to I-87024/I-87017/I-87058/I-7024/I-7033/I-7041 via serial COM port. The assumed situation, we only need the interval time of accessing I-87024 and I-7024 is 1 sec. The interval time of I-7041 and I-87058 is 3 sec. However, we want to update I-7033 and I-87017 every 100 ms. For this purpose, we can achieve it by seven steps as below.

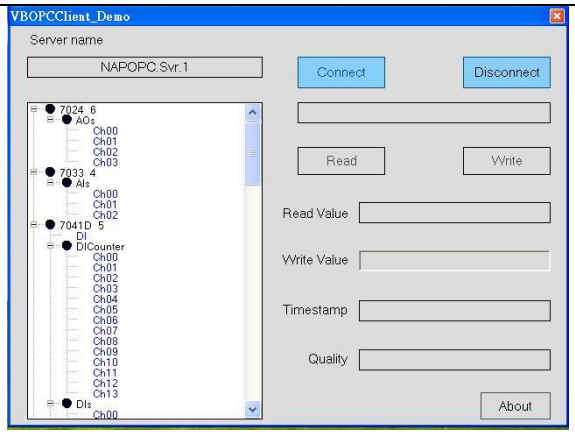
Step 1: First of all, we try to connect all modules on COM1 and to auto search these modules.



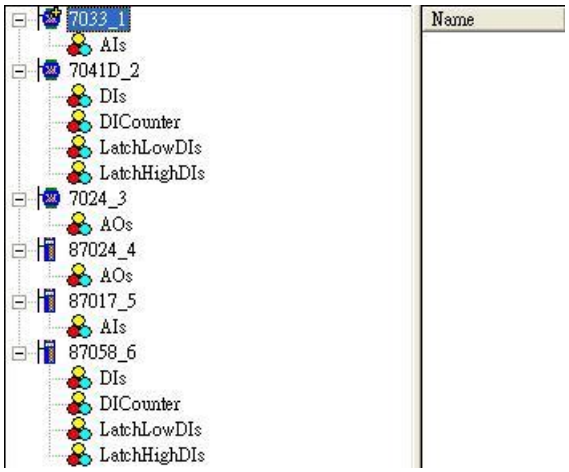
Step 2: Connect OPC client to NAPOPC_ST Server. At the status bar of NAPOPC_ST window, it shows average scan time 

when clicking any module.
We will find the average scan time is 968 ms.

	COM1					
Module	I-87024	I-87017	I-87058	I-7033	I-7041	I-7024
Avg Scan Time	**968 ms**					



Step 3: We divide these modules into two groups. I-87024/I-87017/I-87058 connects to COM1. I-7033/I-7041/I-7024 connects to COM2. And we search again.



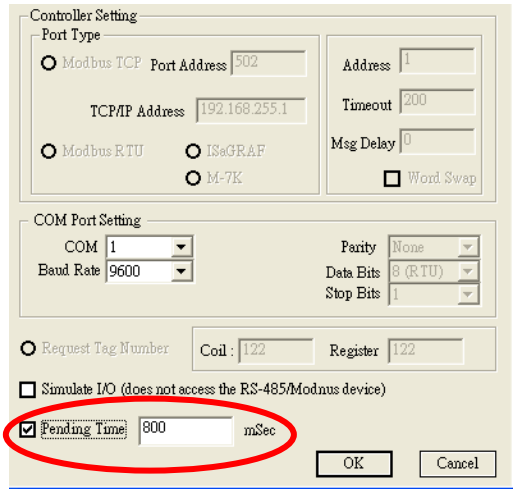
Step 4: Connect OPC client to NAPOPC_ST server again. We can discover the average scan time separately when clicking each module. We will find the average scan time of COM1 is 391 ms, and of COM2 is 516 ms.

	COM 1			COM2		
Module	I-87024	I-87017	I-87058	I-7033	I-7041	I-7024
Avg Scan Time	**391 ms**			**516 ms**		

Step 5: Now we can set pending time to each module as below.

	I-7033	I-7041	I-7024
Pending Time	—	**2000**	**800**

	I-87024	I-87017	I-87058
Pending Time	**800**	—	**2000**



Step 7: Discover the average scan time of COM1 and COM2. We can find it is at our target. The average scan time of COM1 is 62 ms. The average scan time of COM2 is 31 ms.

	COM 1			COM2		
Module	I-87024	I-87017	I-87058	I-7033	I-7041	I-7024
Avg Scan Time	**62 ms**			**31 ms**		

3.2 VB5 Client Demo Program

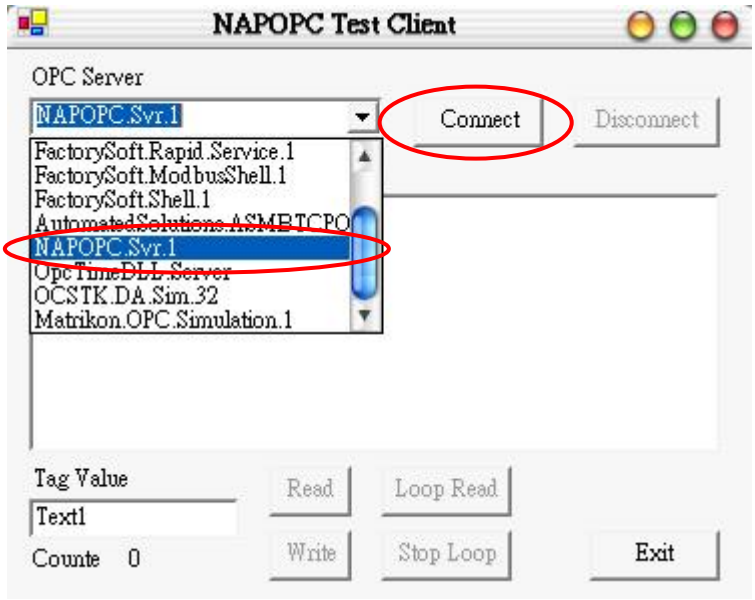
We provide three OPC client demo programs for Visual Basic 5.0, Visual Basic .Net and Visual C# .Net. It is placed under the "\\ICPDAS\NAPOPC_ST\Client\" folder after installation of our NAPOPC_ST server.

Note: The .Net demo programs could compatibility for Visual Studio .Net 2003 or later.

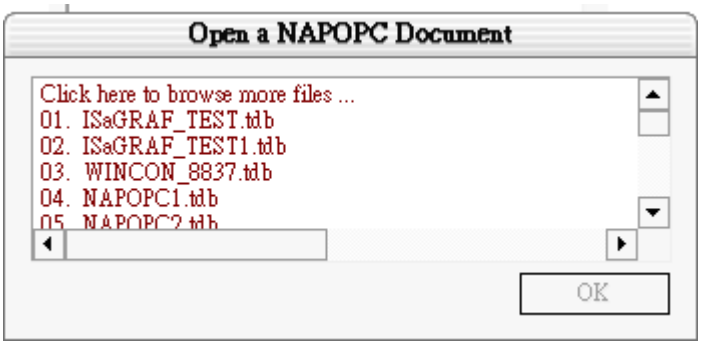
Step 1: Launch the client demo program.
(The client program will search the system registry to find OPC servers. The new servers will be added to the list.)

Step 2: Select the "NAPOPC.Svr.1" OPC Server.

Step 3: Click on the "Connect" button.



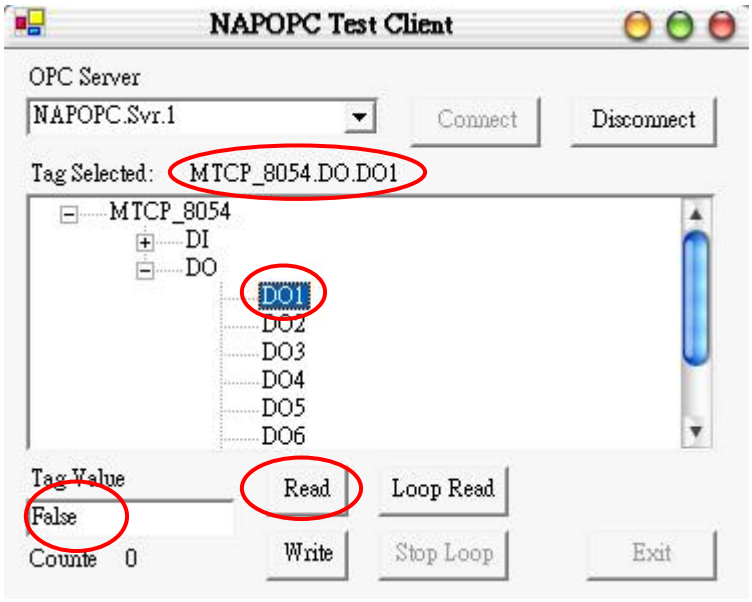
Step 4: Select a file which you want to use and click on the "OK" button.



The following steps 5 ~ 6 are read operation of Modbus TCP controllers.

Step 5: Select a tag matching on your configuration.
(For example: Select the "i-8437" controller, "DO" group and "DO1" tag in the tree-view window.)

Step 6: Click on the "Read" button to read the ""MTCP_8054.DO.DO1" value.

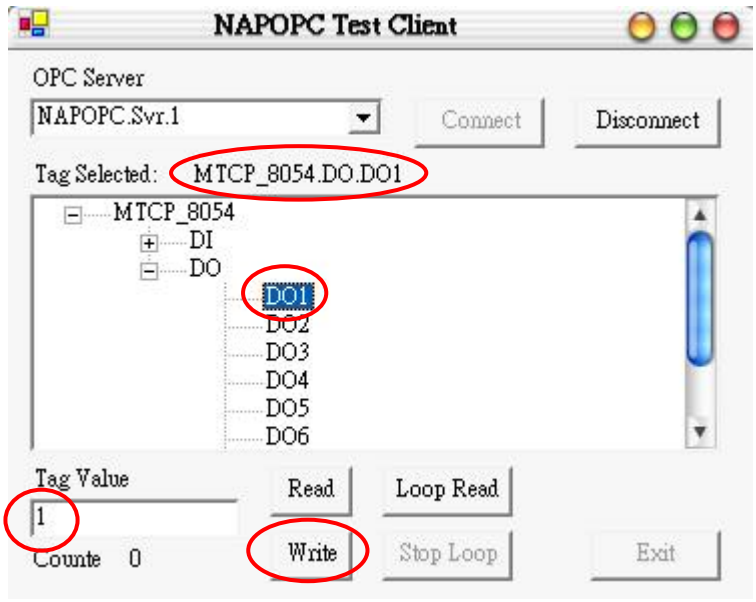


The following steps 7 ~ 9 are write operation of Modbus TCP controllers.

Step 7: Select a tag matching on your configuration.
(For example: Select the "i-8437" controller, "DO" group and "DO1" tag in the tree-view window.)

Step 8: Fill in the "Tag-Value" field with 1.

Step 9: Click the "Write" button to write the "MTCP_8054.DO.DO1" value.

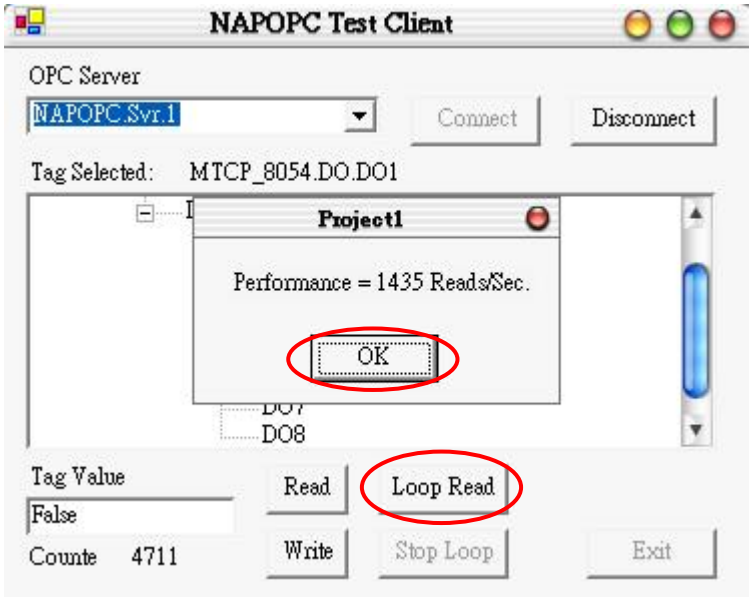


The following steps 10 ~ 12 are read operation, which loops unless you stop it.

Step 10: Click on the "Loop Read" to continuously read data.

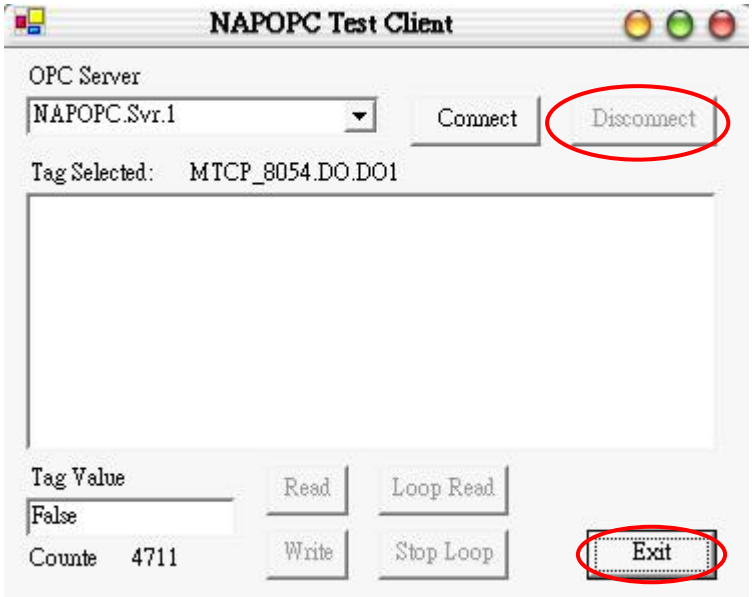
Step 11: After about 5 seconds (or more), click on the "Stop Loop" button to stop reading.

Step 12: A window pops up to show the performance.
Close it by clicking the "OK" button.



Step 13: Click on the "Disconnect" button to disconnect from the OPC server.

Step 14: Click on the "Exit" button to end the client demo program.

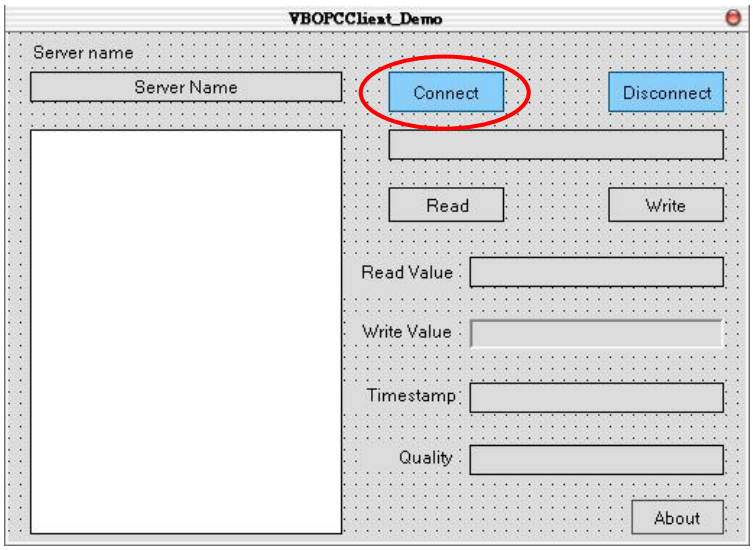


3.3 .Net Client Demo Program

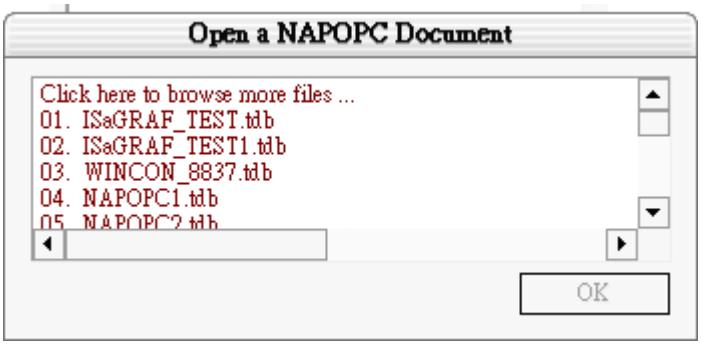
We provide another two OPC client demo programs for Visual Basic .Net 2003 and Visual C# .Net 2003. It's placed under the following folder "\\ICPDAS\NAPOPC_ST\Client\OPC_NetClientDemo\VBOPCClient_Demo" and "\\ICPDAS\NAPOPC_ST\Client\OPC_NetClientDemo\VCSOPCClient_Demo"after installation of our NAPOPC_ST server.

Step 1: Launch the client demo program "VBOPCClient_Demo.exe" or "VCSOPCClient_Demo.exe". (The client program set "NAPOPC.Svr.1" as default OPC Server)

Step 2: Click on the "Connect" button.



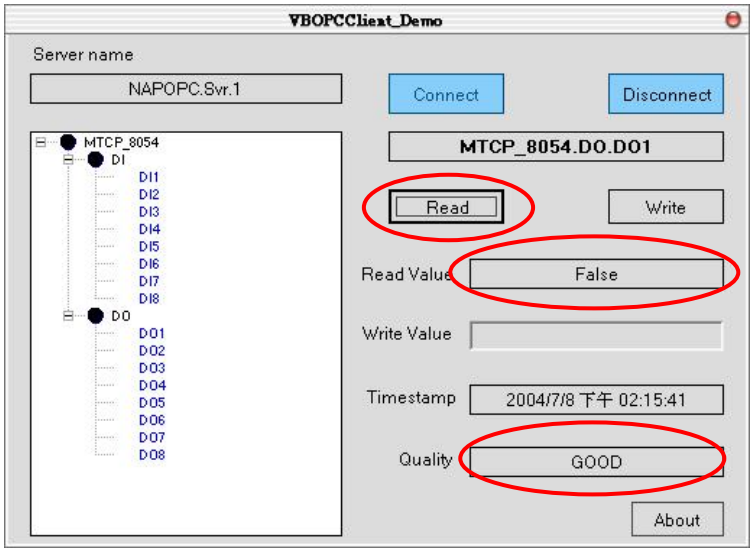
Step 3: Select a file which you want to use and click on the "OK" button.



The following steps 4 ~ 5 are read operation of Modbus TCP controllers.

Step 4: Select a tag matching on your configuration.
(For example: Select the "i-8437" controller, "DO" group and "DO1" tag in the tree-view window.)

Step 5: Click on the "Read" button to read the "MTCP_8054.DO.DO1" value.

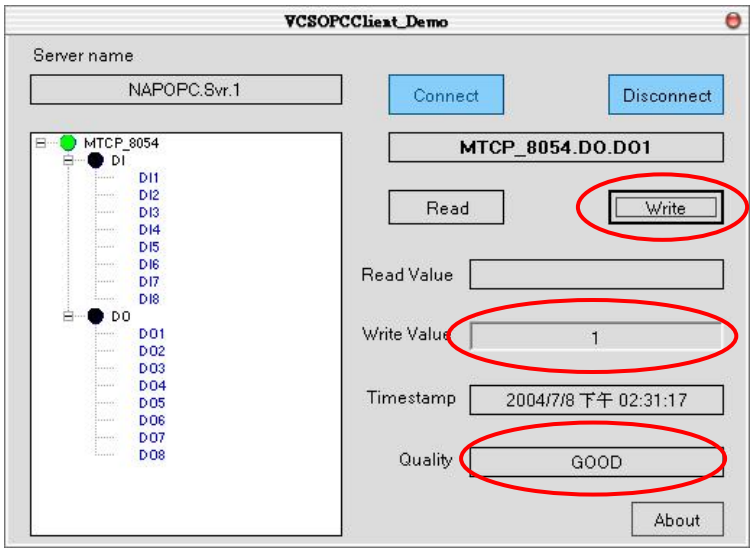


The following steps 6 ~ 8 are write operation of Modbus TCP controllers.

Step 6: Select a tag matching on your configuration.
(For example: Select the "i-8437" controller, "DO" group and "DO1" tag in the tree-view window.)

Step 7: Fill in the "Write Value" field with 1.

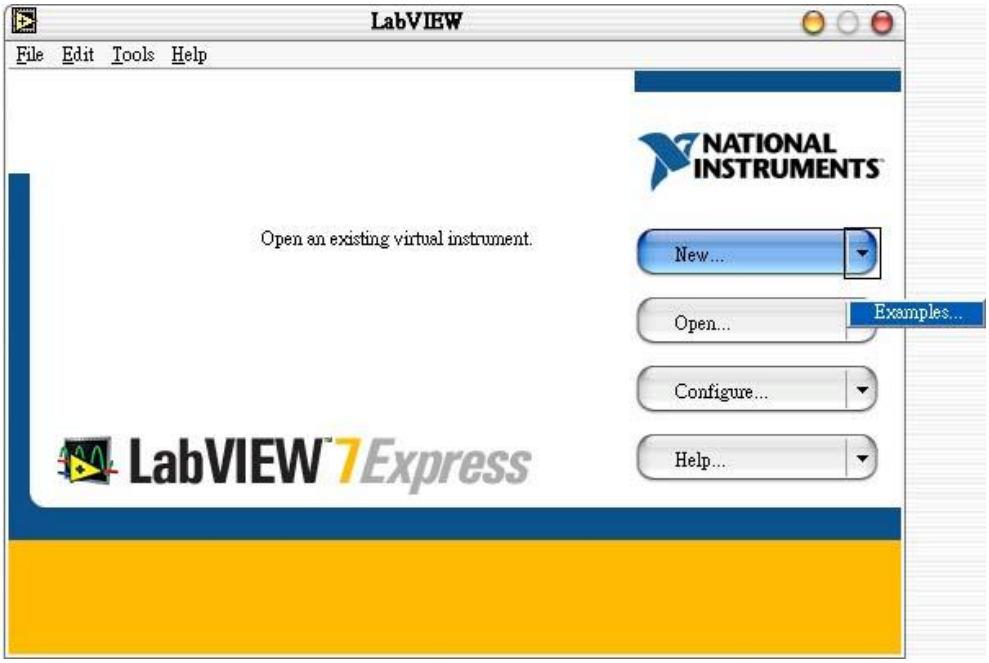
Step 8: Click the "Write" button to write the "MTCP_8054.DO.DO1" value.



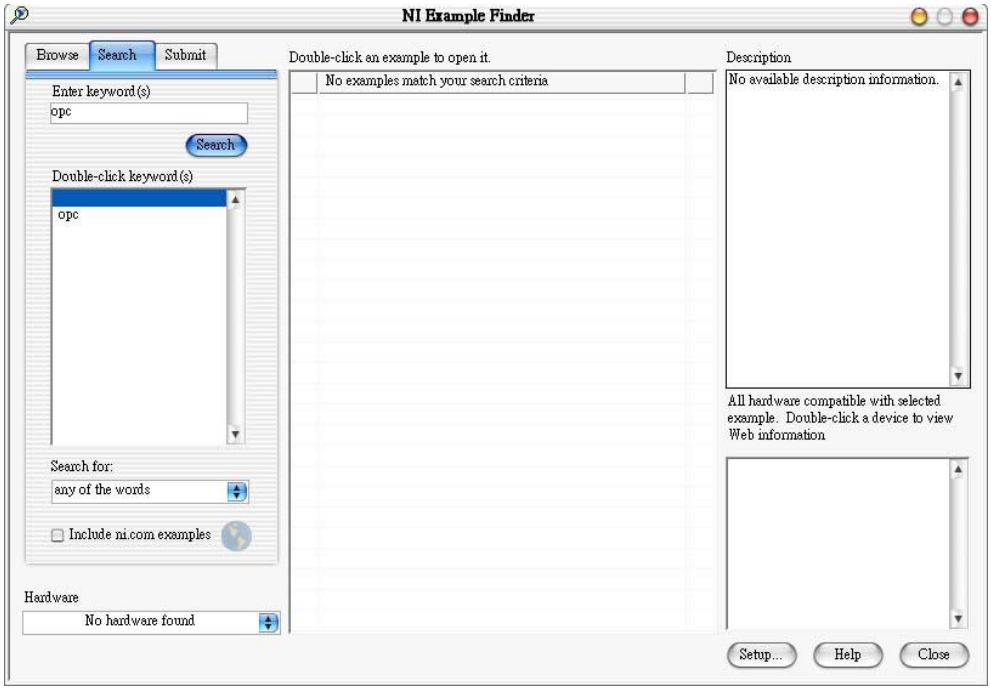
Step 13: Click on the "Disconnect" button to disconnect from the OPC server.

3.4 LabVIEW

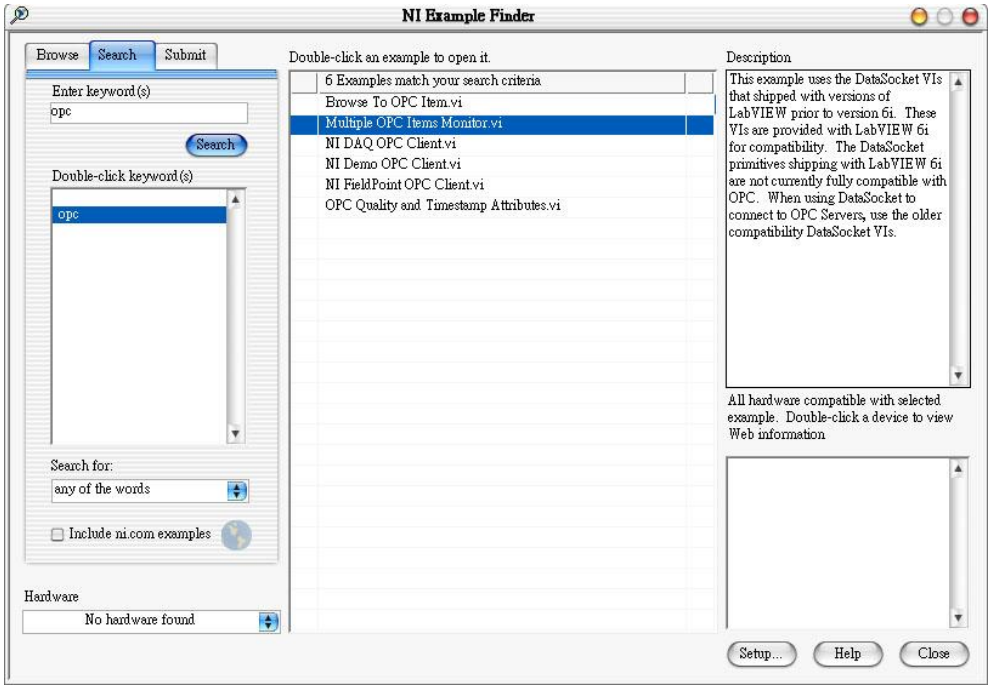
Step 1: Run the LabVIEW program and select "Open…" -> Example



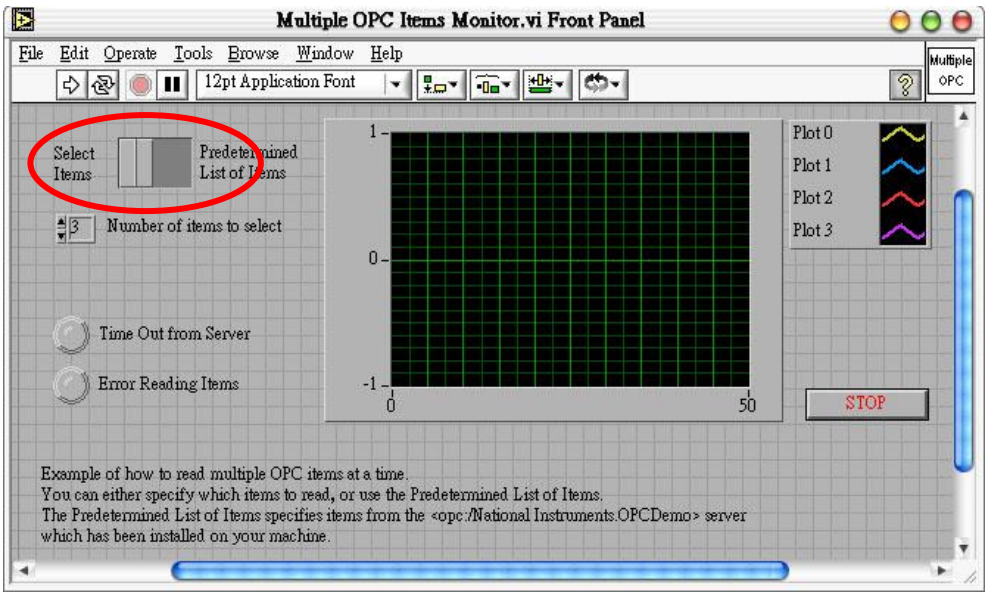
Step 2: Click on the "Search Examples" button to get information on using OPC.



Step 3: Double-click on the "Multiple OPC Items Monitor.vi" item in the middle window of NI Example Finder dialog..



Step 4: Click on the "Select Items" item in the "Multiple OPC Items Monitor.vi" demo.



Step 5: Run this demo.

Step 6: Click on a machine name in the "Network" tree-view.

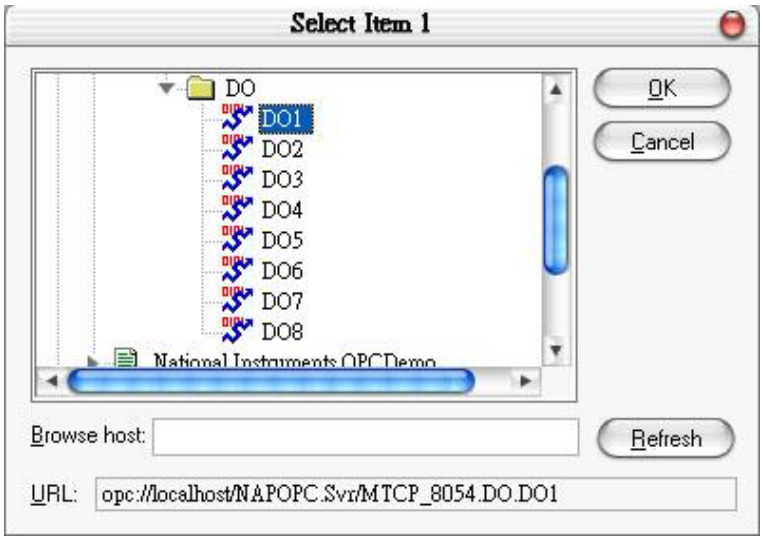
Step 7: Select the "NAPOPC.Svr" OPC server.

Step 8: Click on the "OK" button to close it.

Step 9: Select a file which you want to use and click on the "OK" button.

Step 10: Select an item (tag) in the tree-view.

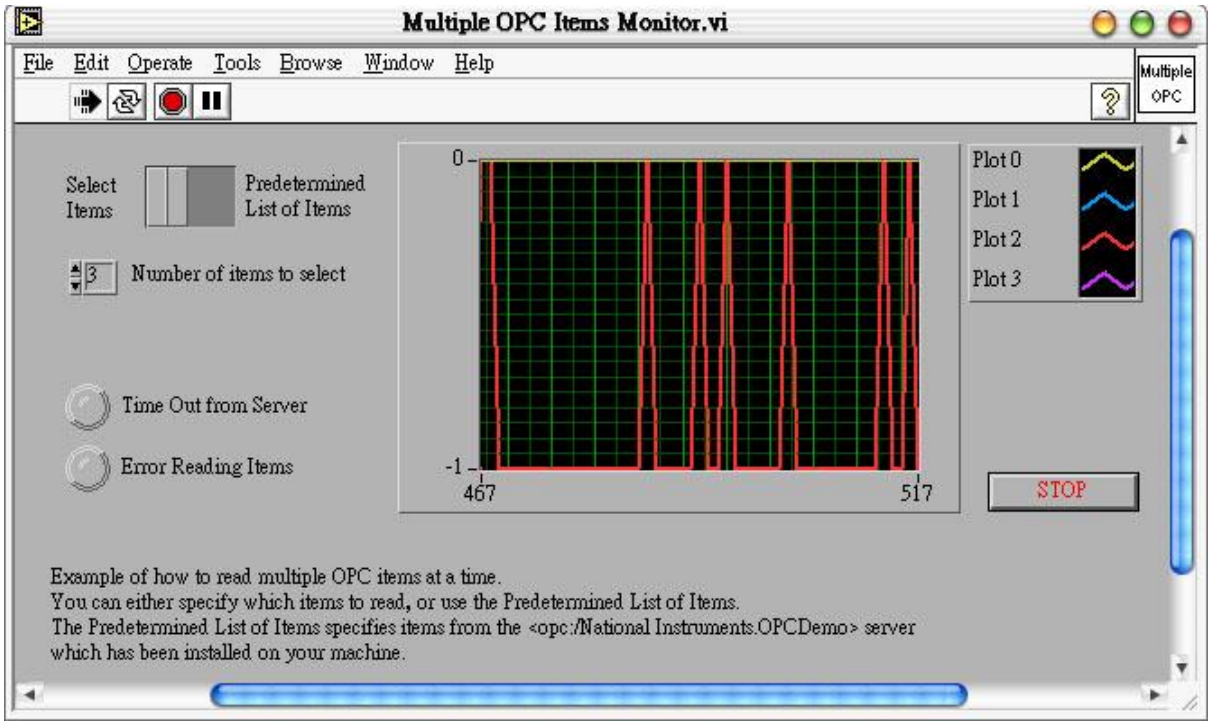
Step 11: Click on the "OK" button to add this one.



Step 12: Repeat the steps 6 ~ 11 to add more items(tags).

Step 13: Click on the "Cancel" button to finish adding items(tags).

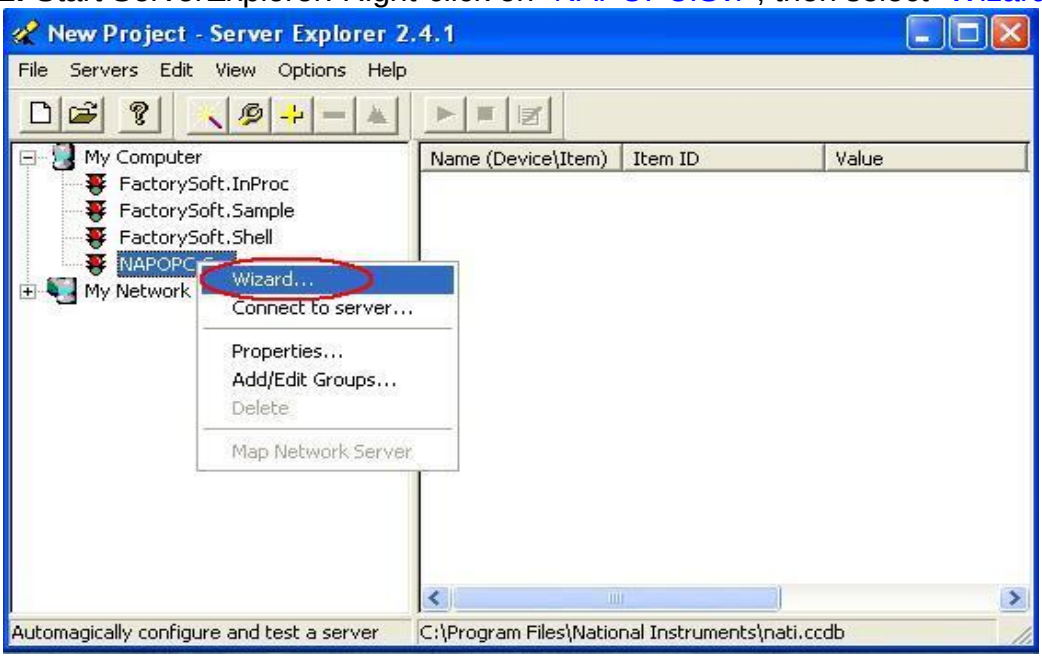
Step 14: The grid window graphs a line(s) to show changes of each item (tag).



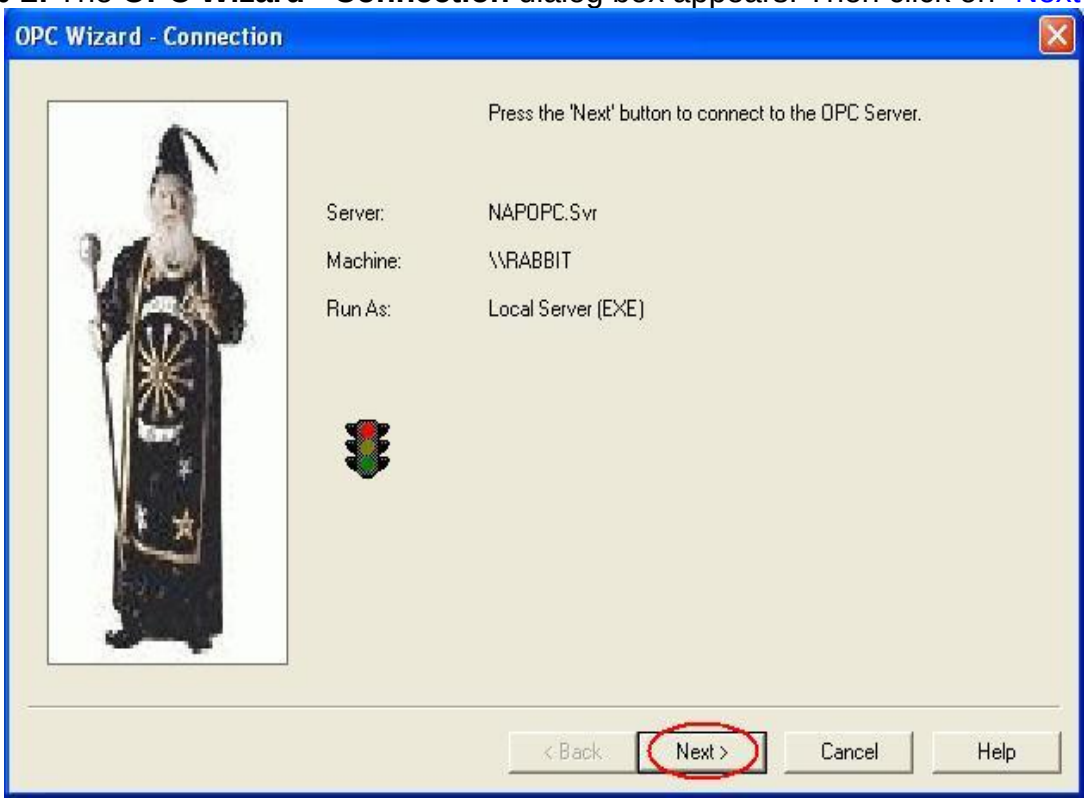
3.5 NATIONAL INSTRUMENTS

National Instruments is a comprehensive industrial automation company by providing the software, hardware, and technologies necessary to transform personal computers into powerful computer-based and networked measurement and automation systems. The ServerExplorer is one of their products for connecting to OPC Server. For more information, please visit http://www.ni.com

Step 1: Start ServerExplorer. Right-click on "NAPOPC.Svr", then select "Wizard".

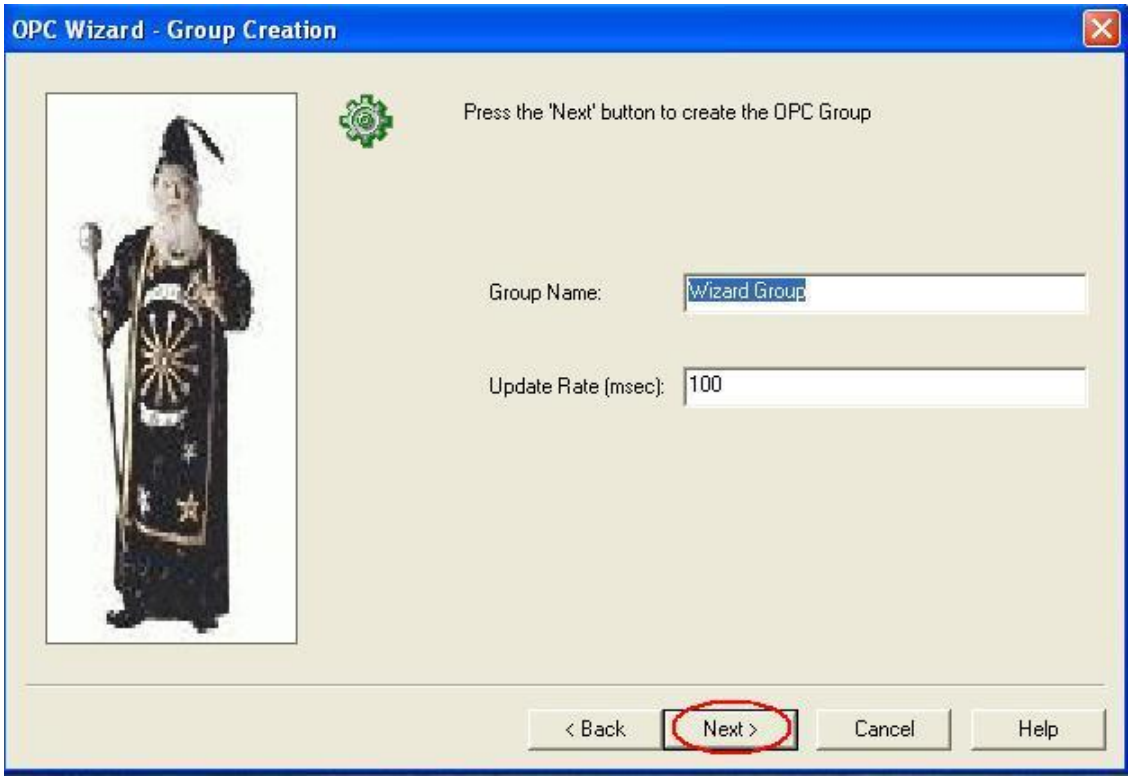


Step 2: The **OPC Wizard - Connection** dialog box appears. Then click on "Next>"

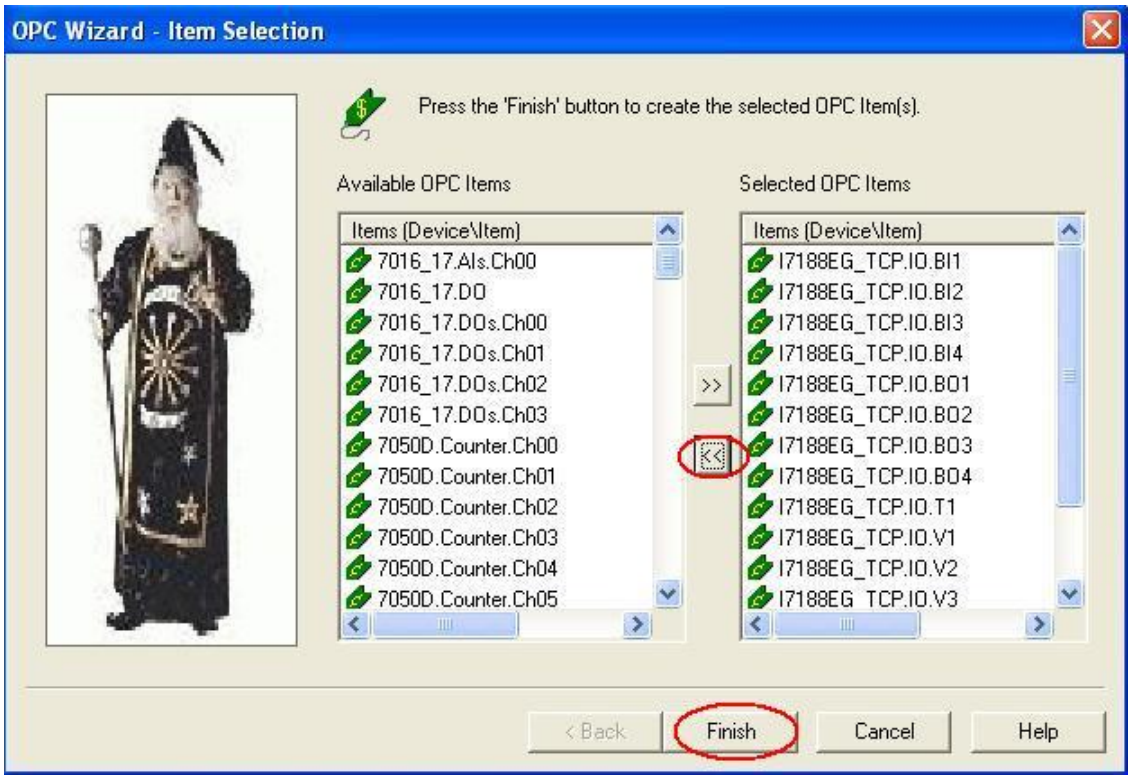


Step 3: Select a file which you want to use and click on the "OK" button.

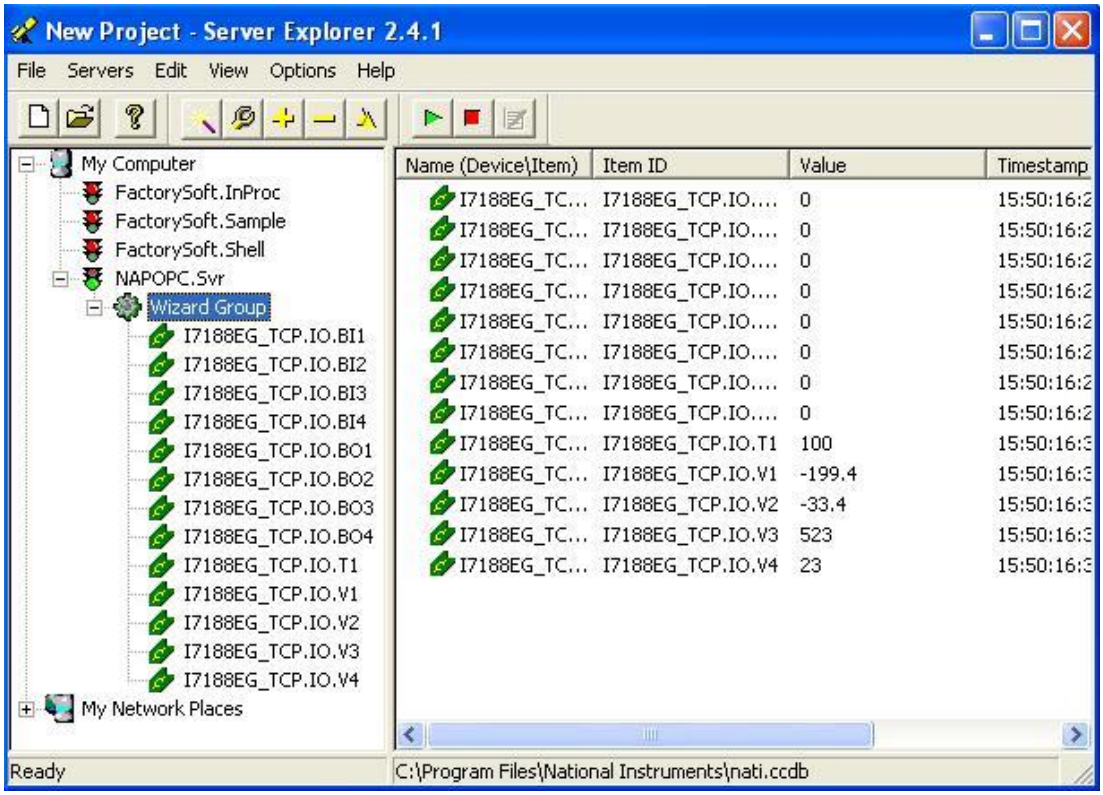
Step 4: Enter the **Group Name** and **Update Rate**. The name can be any name you want. Click "Next>" to continue.



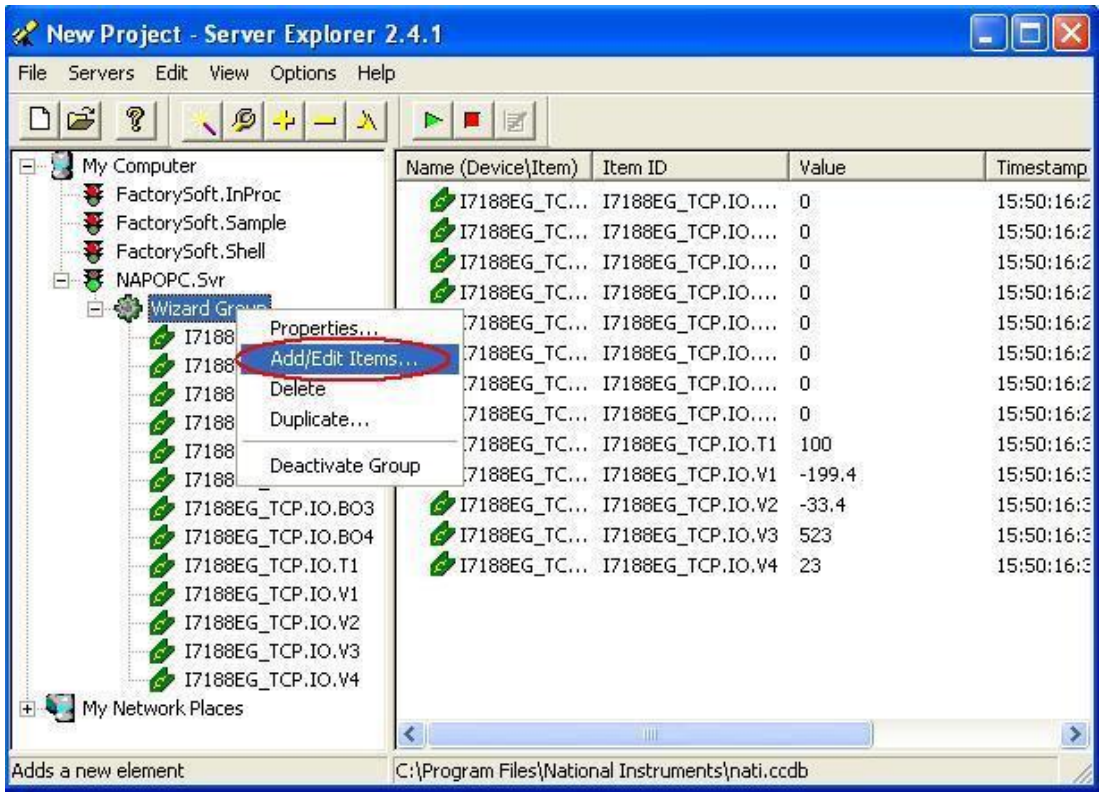
Step 5: Select all the items that you want to view from the **Available OPC Items** list. Then click "Finish".



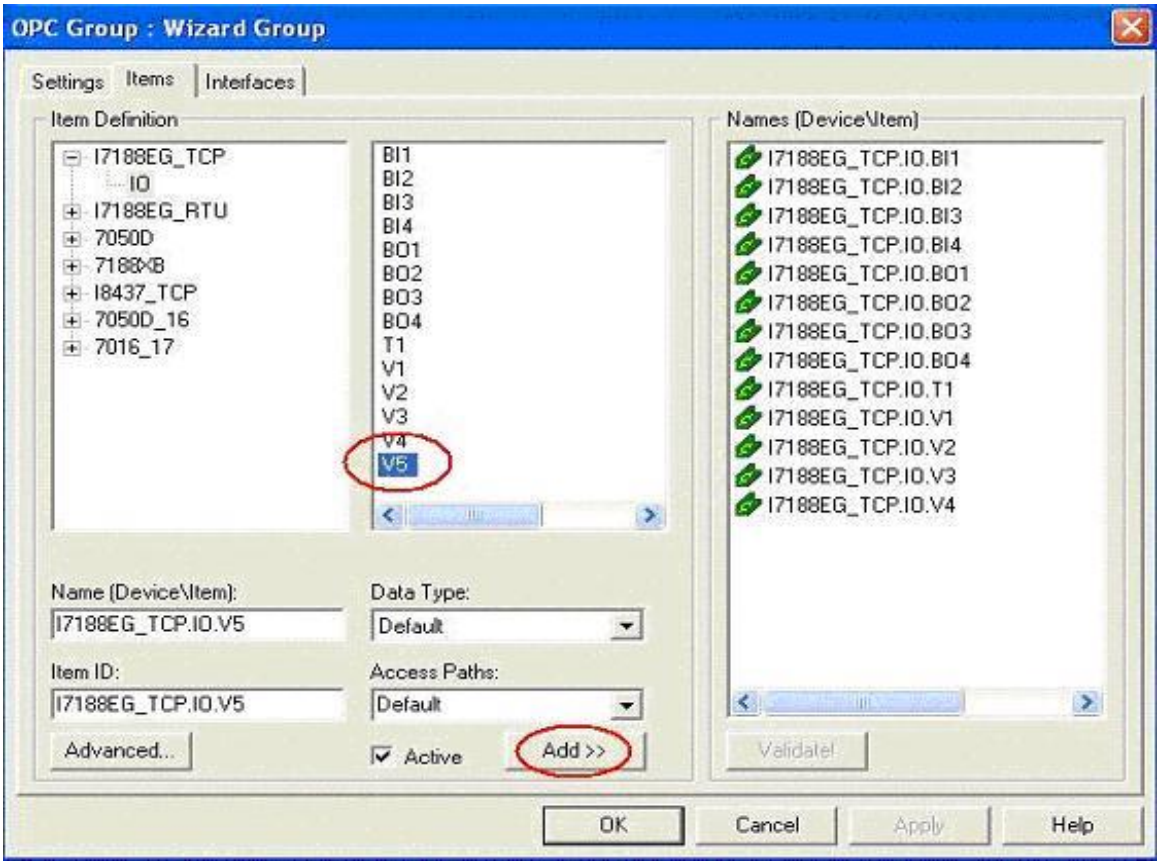
Step 6: Now you should be able to view the OPC connection that you just created.



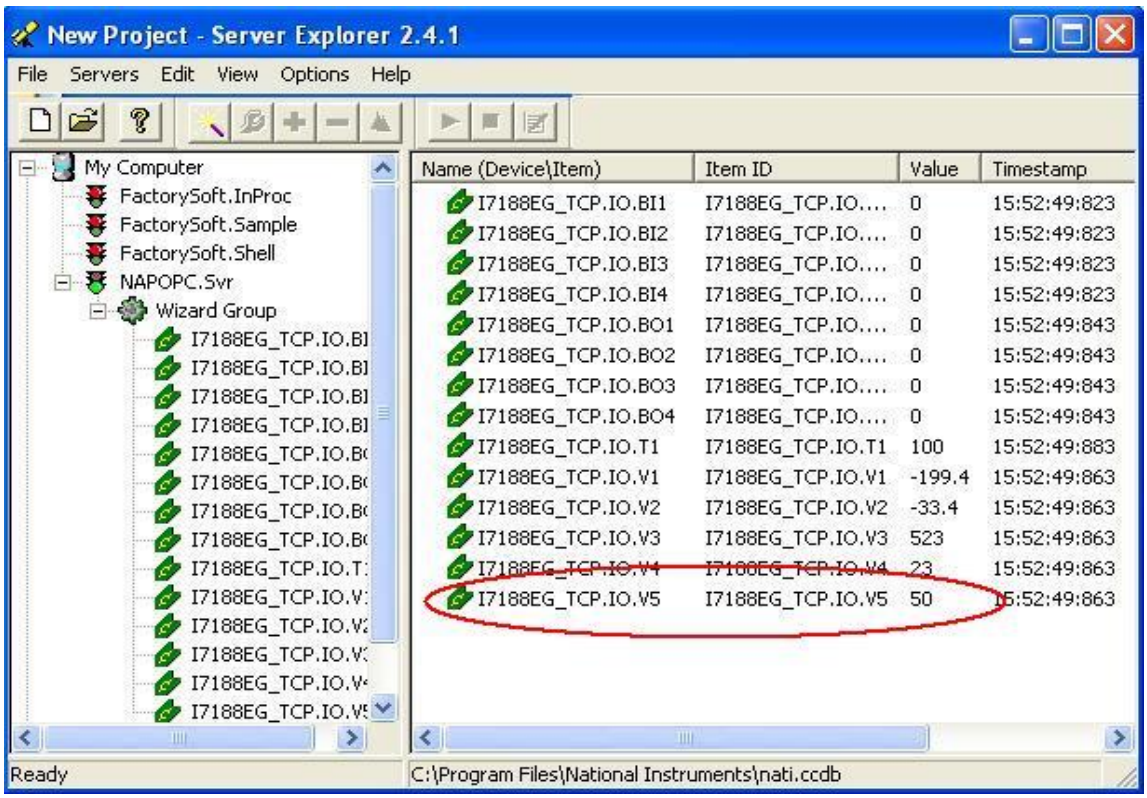
Step 7: To add new items, right-click on the group name then select "Add/Edit Items".



Step 8: Make sure the **Item ID** textbox at the bottom has the correct object and item name. Then click "Add>>" to add the item to the list on the right. In this example, we add the "I7188EG_TCP.IO.V5". Click "OK" when you are done.



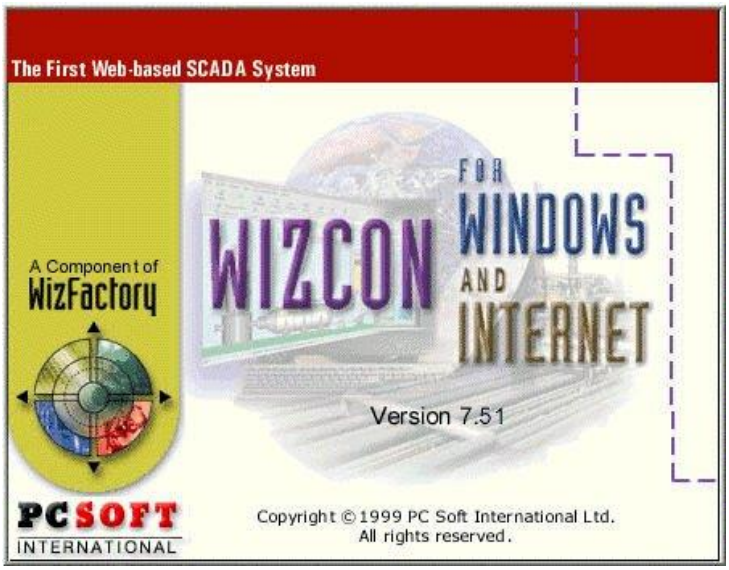
Step 9: Now you should be able to read all the items that you added in the main window.



3.6 WIZCON

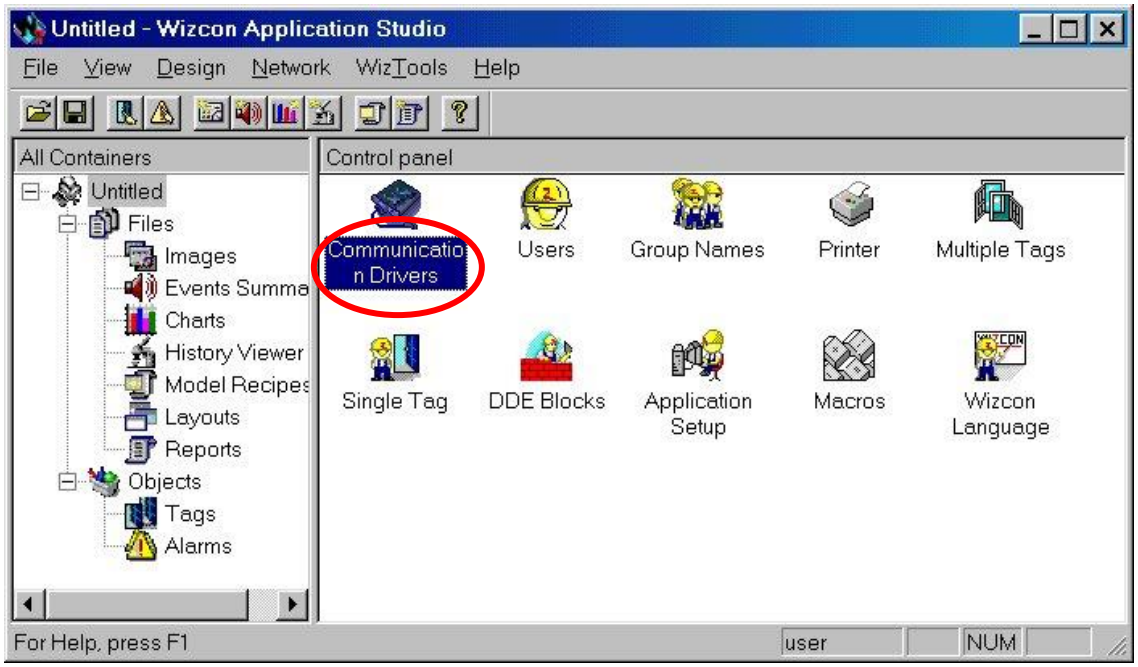
Wizcon for Windows and Internet is a powerful HMI/SCADA software package that delivers real-time and historical information from the plant floor to the boardroom and beyond. For more information, please visit http://www.emation.com

Step 1: Run WIZCON (Version 7.51 or newer) program.

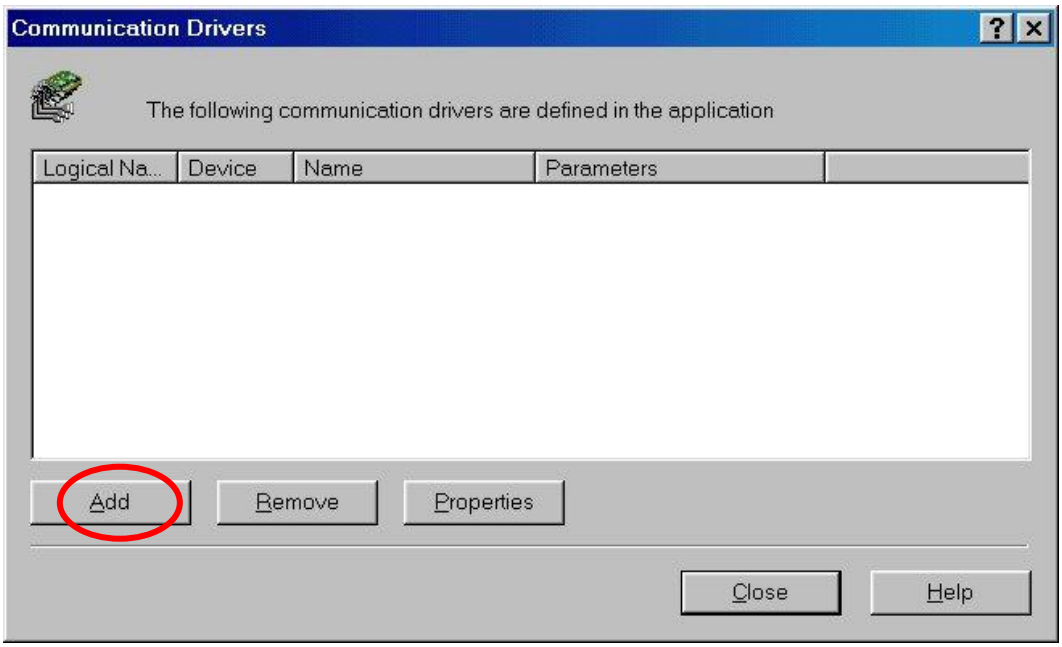


Step 2: Create a new project.

Step 3: Click on the "Communication Drivers" icon in the right hand window.

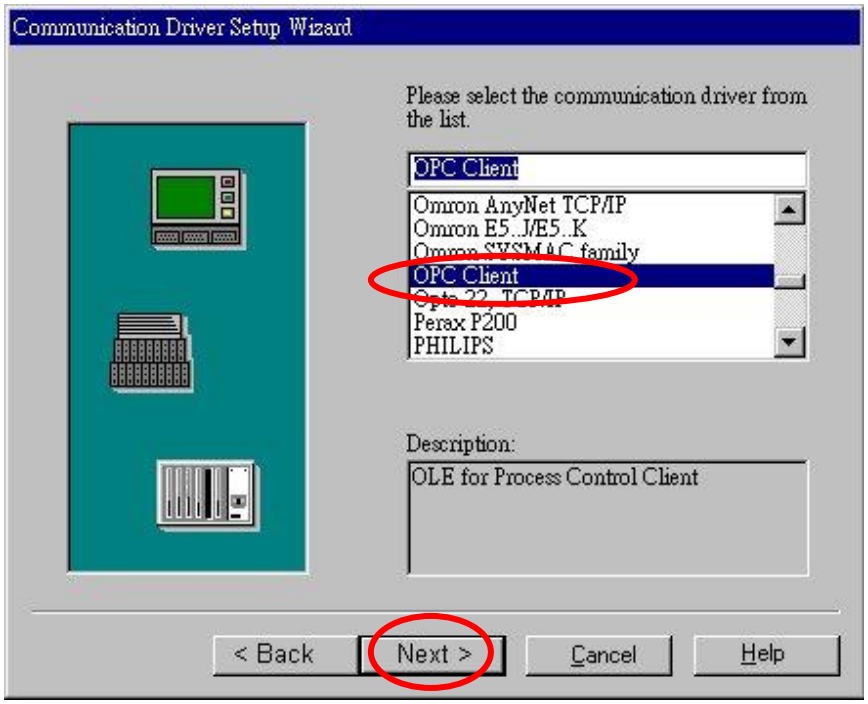


Step 4: Click on the "Add" button to add new drivers.



Step 5: Select the "OPC Client" item.

Step 6: Click on the "Next >" button.

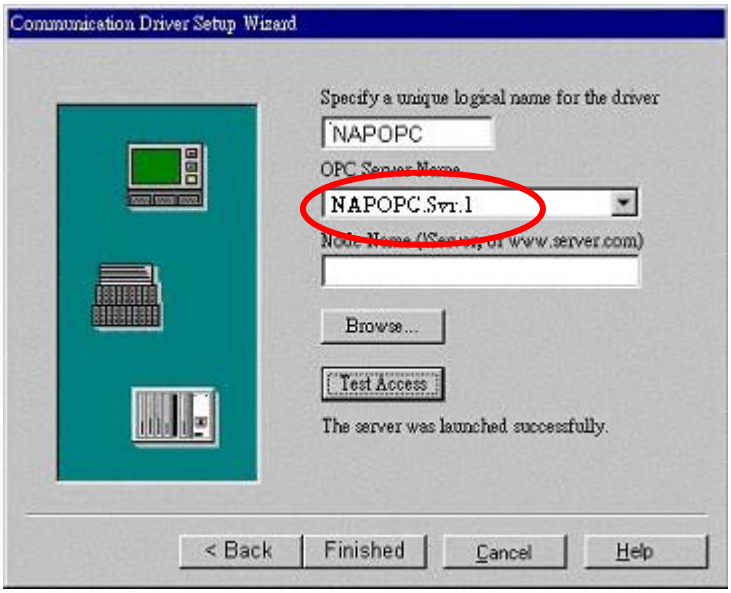


Step 7: Enter the driver name (for example: "NAPOPC").

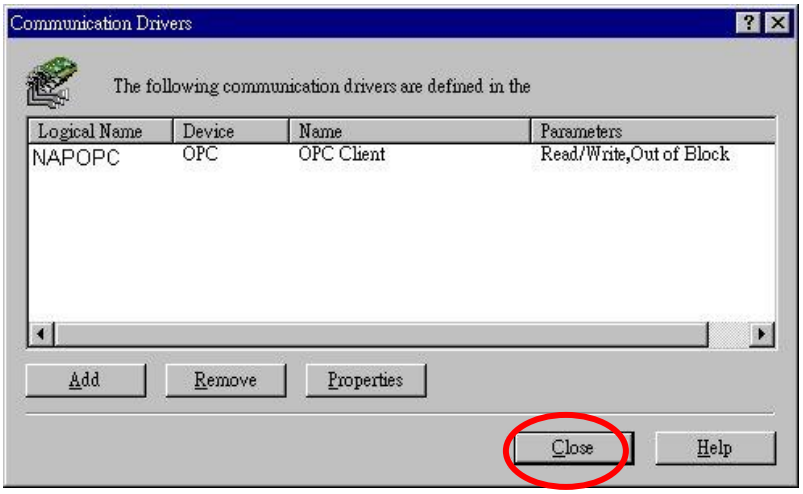
Step 8: Select the OPC Server Name as "NAPOPC.Svr.1".

Step 9: Click on the "Test Access" button to see if the OPC server can be accessed.

Step 10: Click on the "Finish" button.

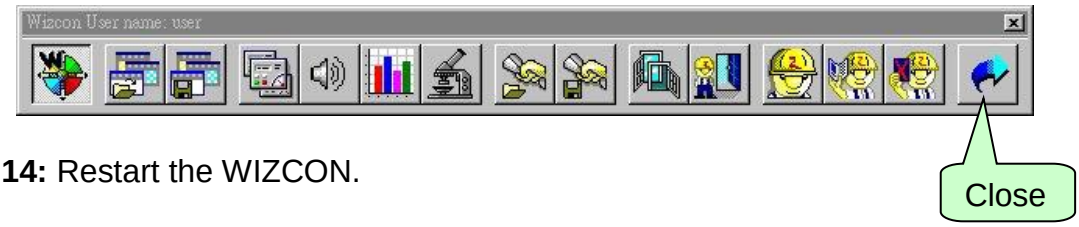


Step 11: Click on the "Close" button.



Step 12: Click on the "OK" button to close the window.

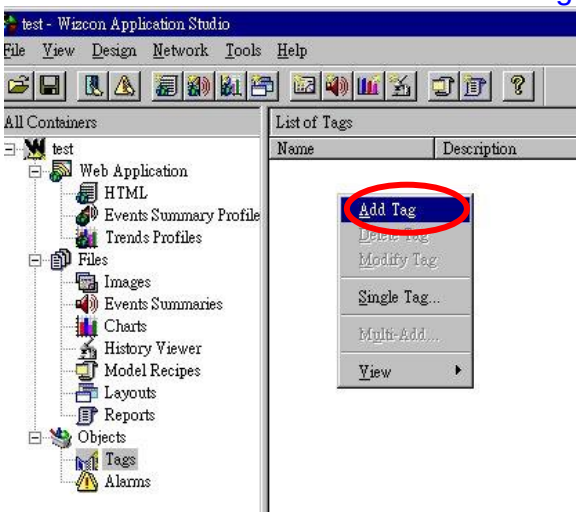
Step 13: Click on the far-right icon (the arrow) to close the WIZCON.



Step 14: Restart the WIZCON.

Step 15: Select the "Tags" item from the left-hand window.

Step 16: Right click the mouse button and select the "Add Tag" option to add tag(s).



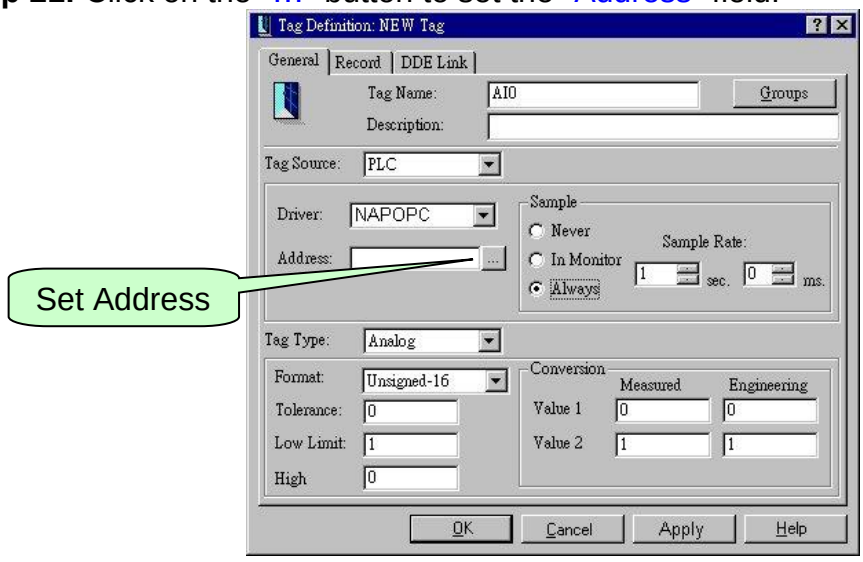
Step 17: Enter a tag name in the "Tag Name" field.

Step 18: Select "PLC" in the "Tag Source" field.

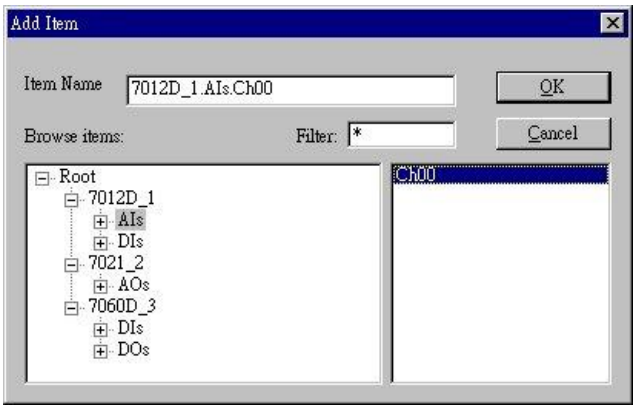
Step 19: Select "NAPOPC"" in the "Driver" field.

Step 20: Select "Always" in the "Sample" field.

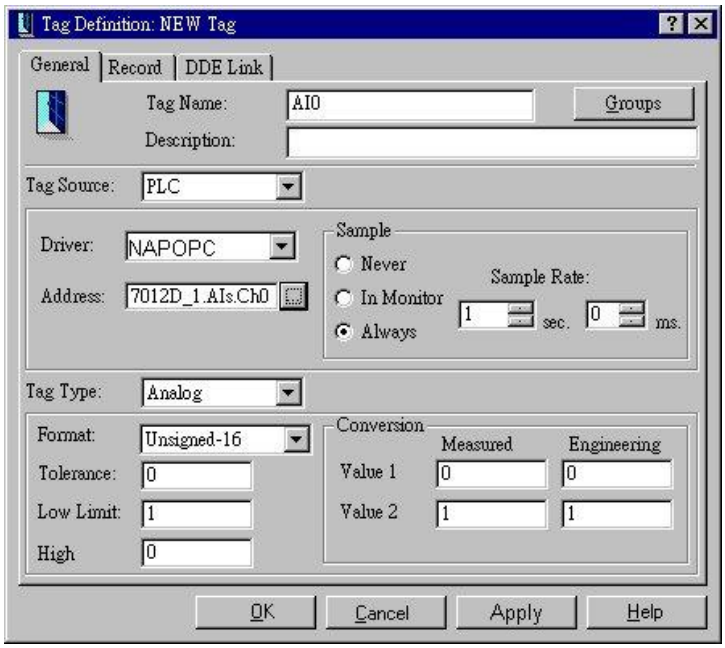
Step 21: Click on the "…" button to set the "Address" field.



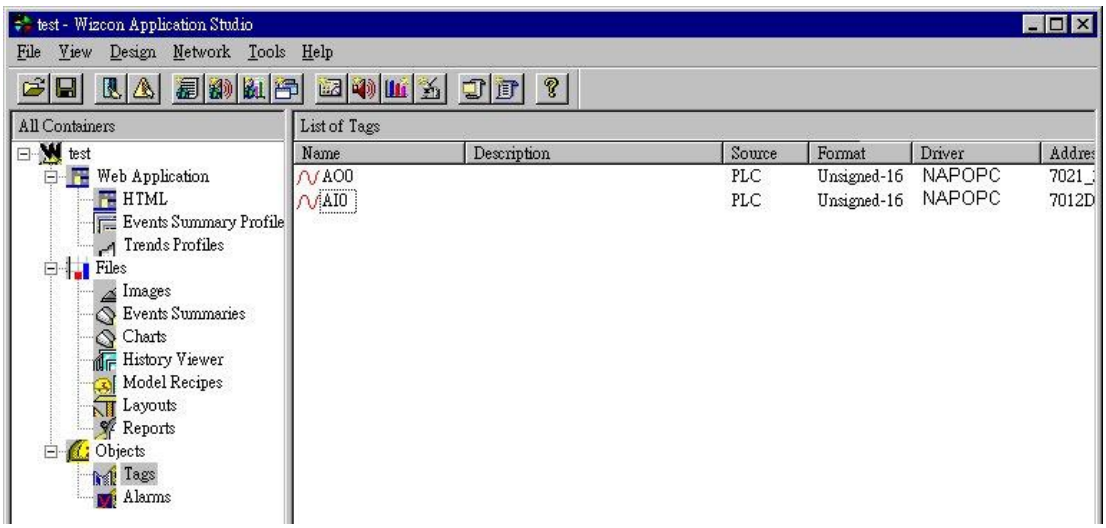
Step 22: Select a tag and click on the "OK" button.



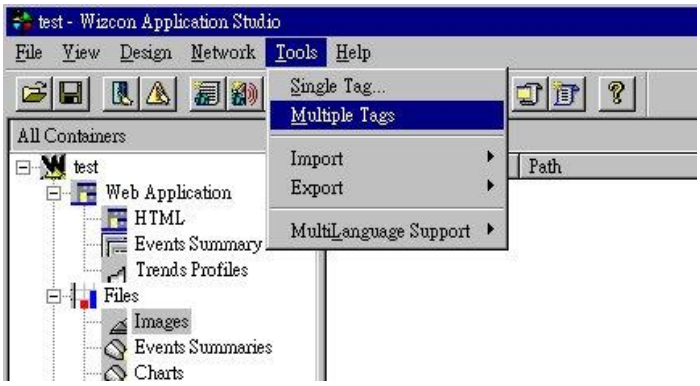
Step 23: Click on the "OK" button to close it.



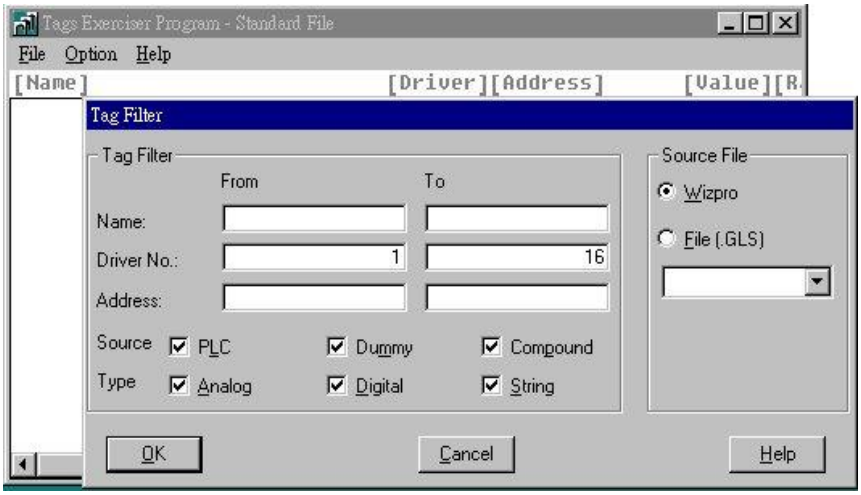
Step 24: The right hand window shows the tag(s) that were previously added.



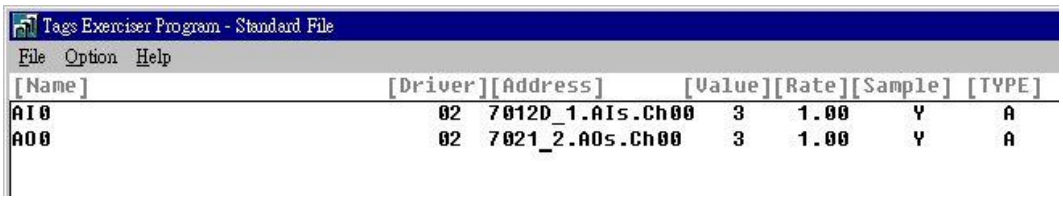
Step 25: Click on the "Tools/ Multiple Tags" menu item.



Step 26: Click on the "OK" button to close the "Tag Filter" window.



Step 27: The "Tags Exerciser Program" window shows tag(s) and value(s).

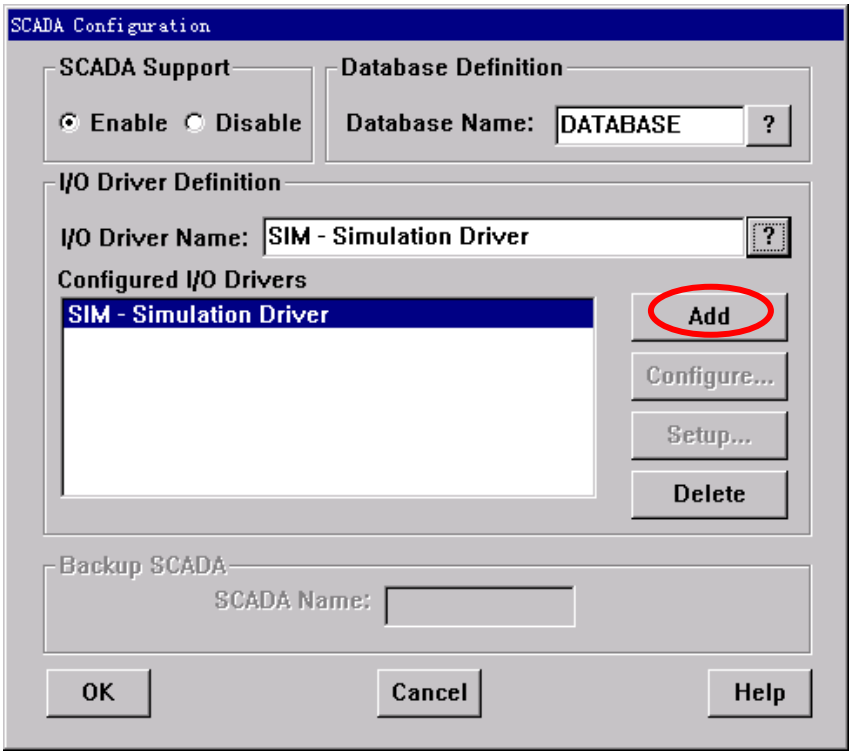


3.7 iFix

iFIX is a powerful HMI/SCADA system that features full process visualization, data collection and management, and supervisory control. iFIX, the HMI/SCADA component of the Installation Dynamics family of automation software, is a Windows based industrial automation solution for monitoring and controlling manufacturing operations. For more information, please visit http://www.intellution.com.

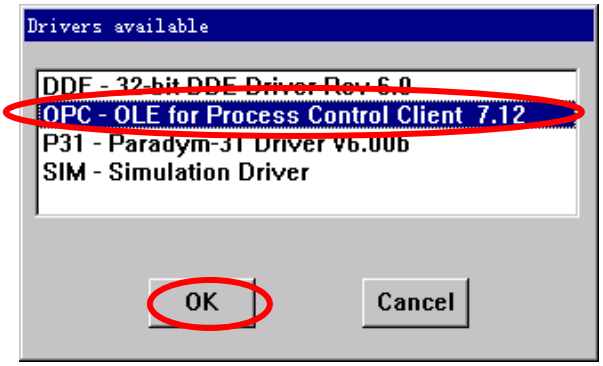
Step 1: Run iFix 2.1 and start system configuration.

Step 2: Click on the "Add" button to add I/O drivers.



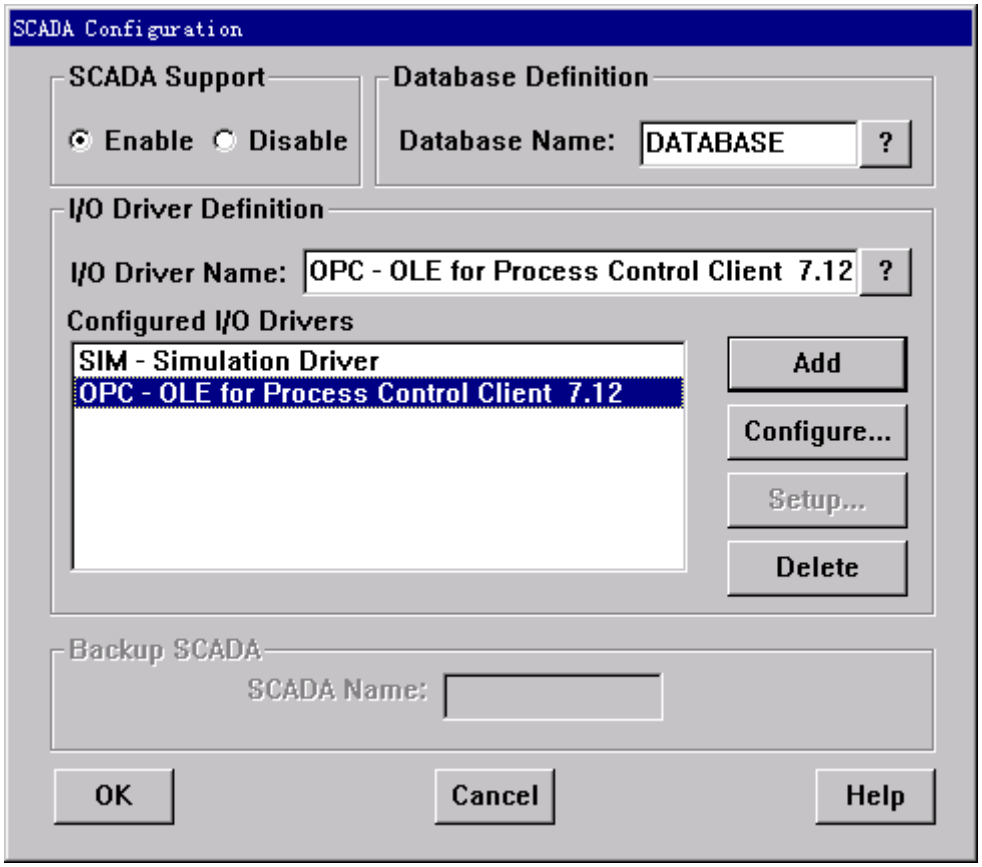
Step 3: Select the "OPC - OLE for Process Control Client 7.12" driver.

Step 4: Click on the "OK" button.

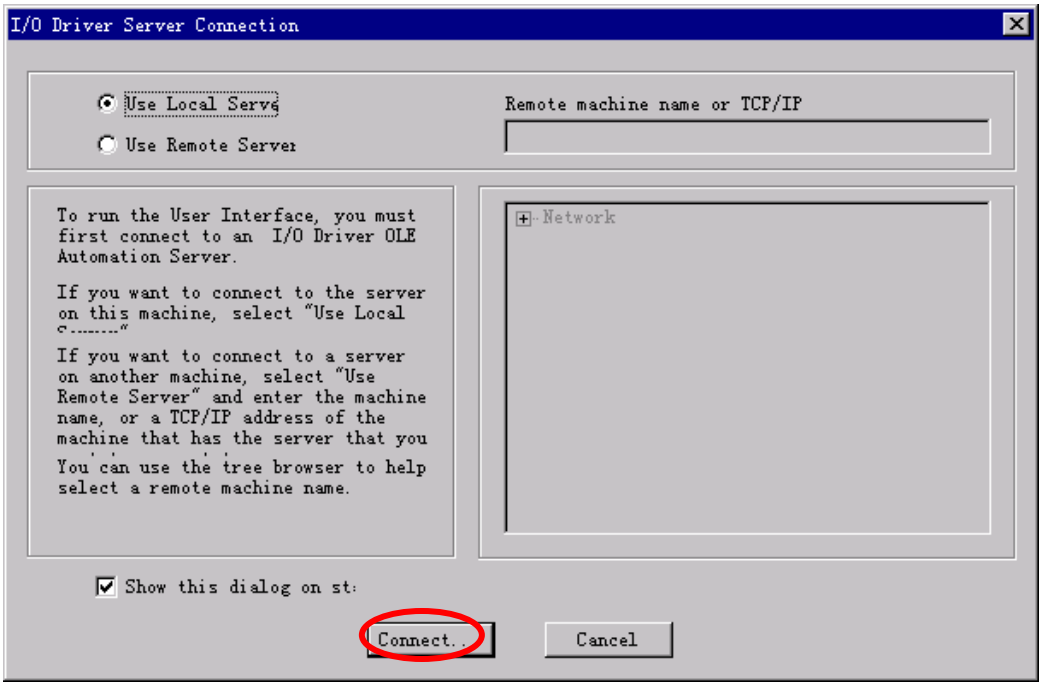


Step 5: Select the "OPC - OLE for Process Control Client 7.12" driver.

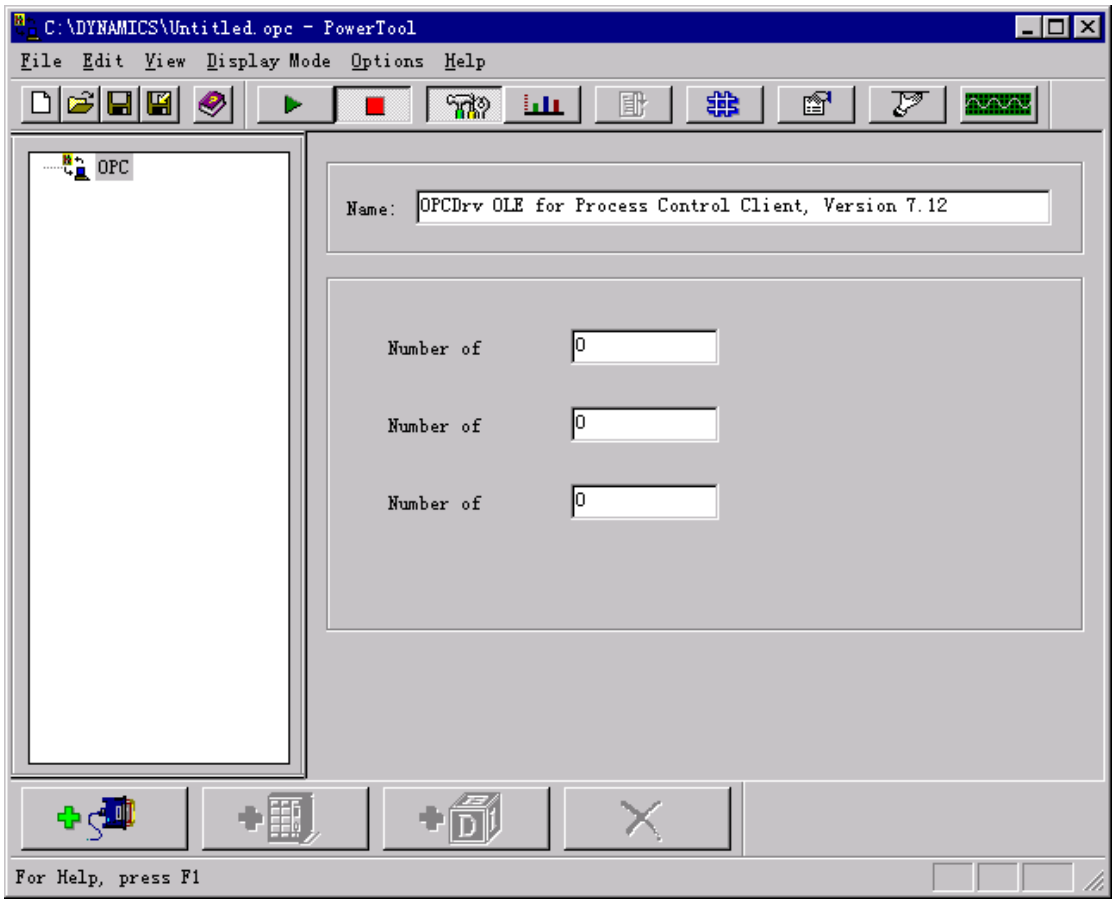
Step 6: Click on the "Configure…" button to configure the I/O driver.



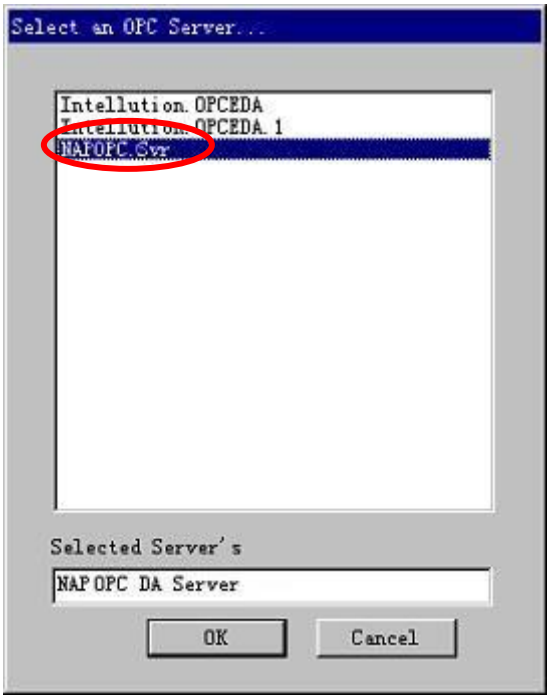
Step 7: Click on the "Connect…" button.



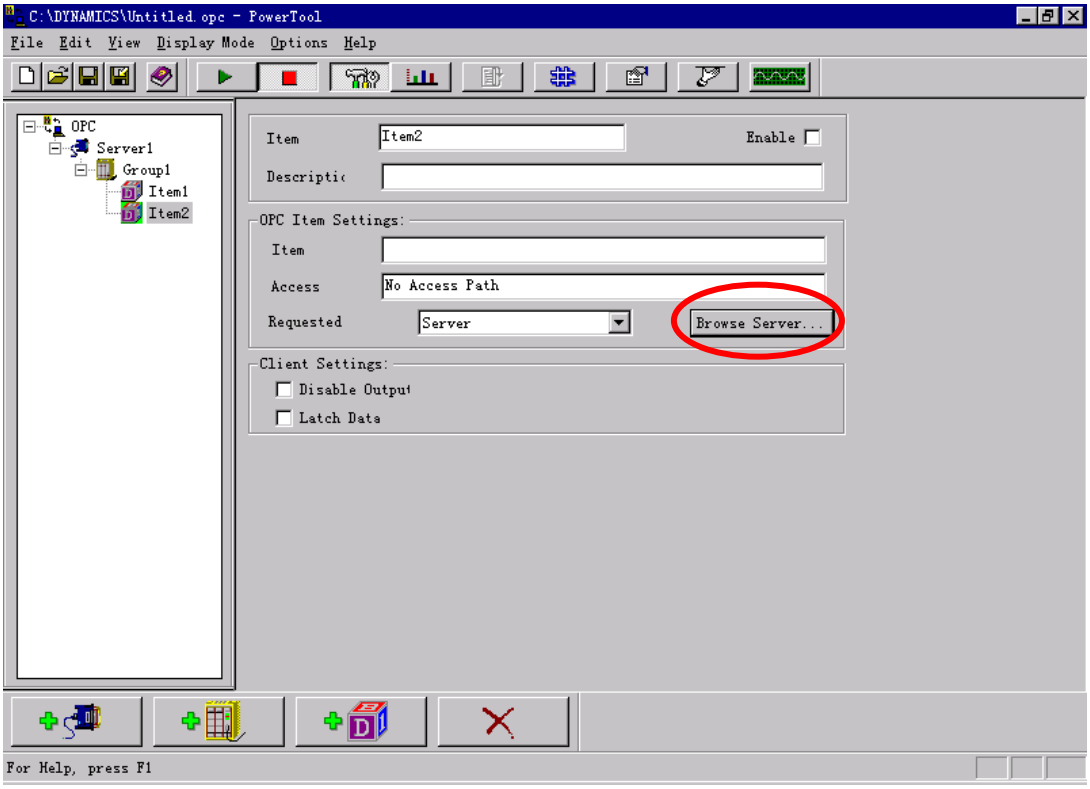
Step 8: To configure the OPC server.



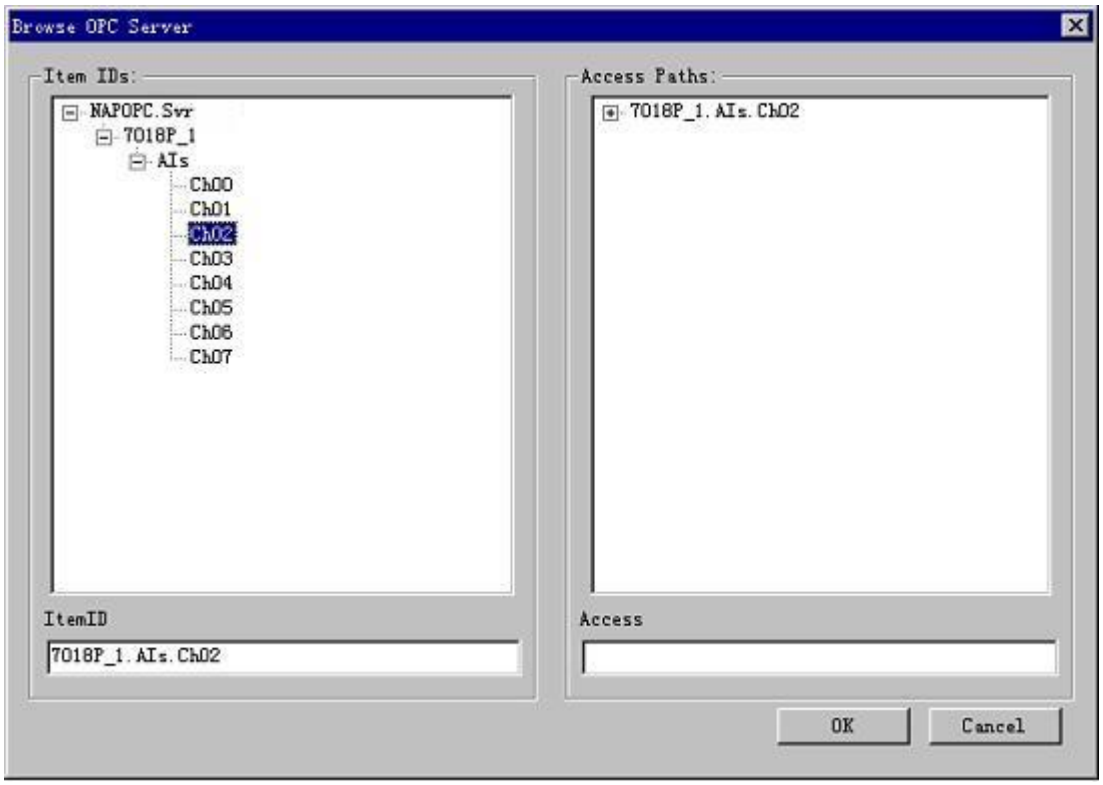
Step 9: Select the "NAPOPC.Svr" and click on the "OK" button.



Step 10: Add server, group and items. Fill properties by clicking on the "Browse Server…" button.



Step 11: Select the item which you need. Click on the "OK" button.



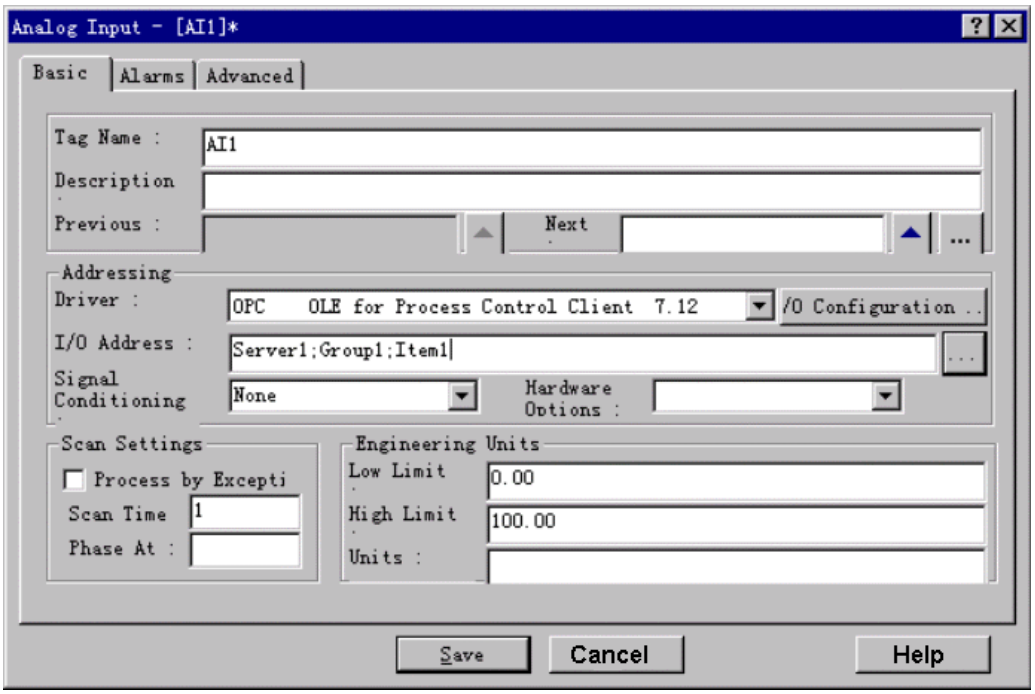
Step 12: Enter database manager of iFix 2.1.

Step 13: Add relative data units.

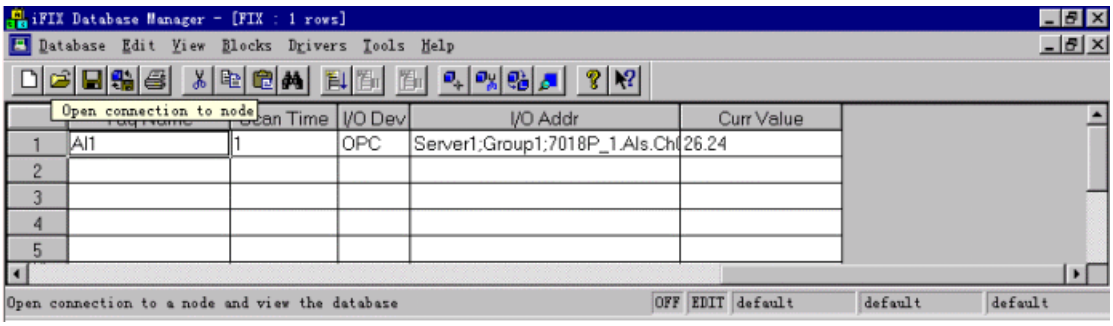
For example: AI.

Driver: "OPC OLE for Process Control Client 7.12".

I/O Address: "Server1;Group1:Item1".



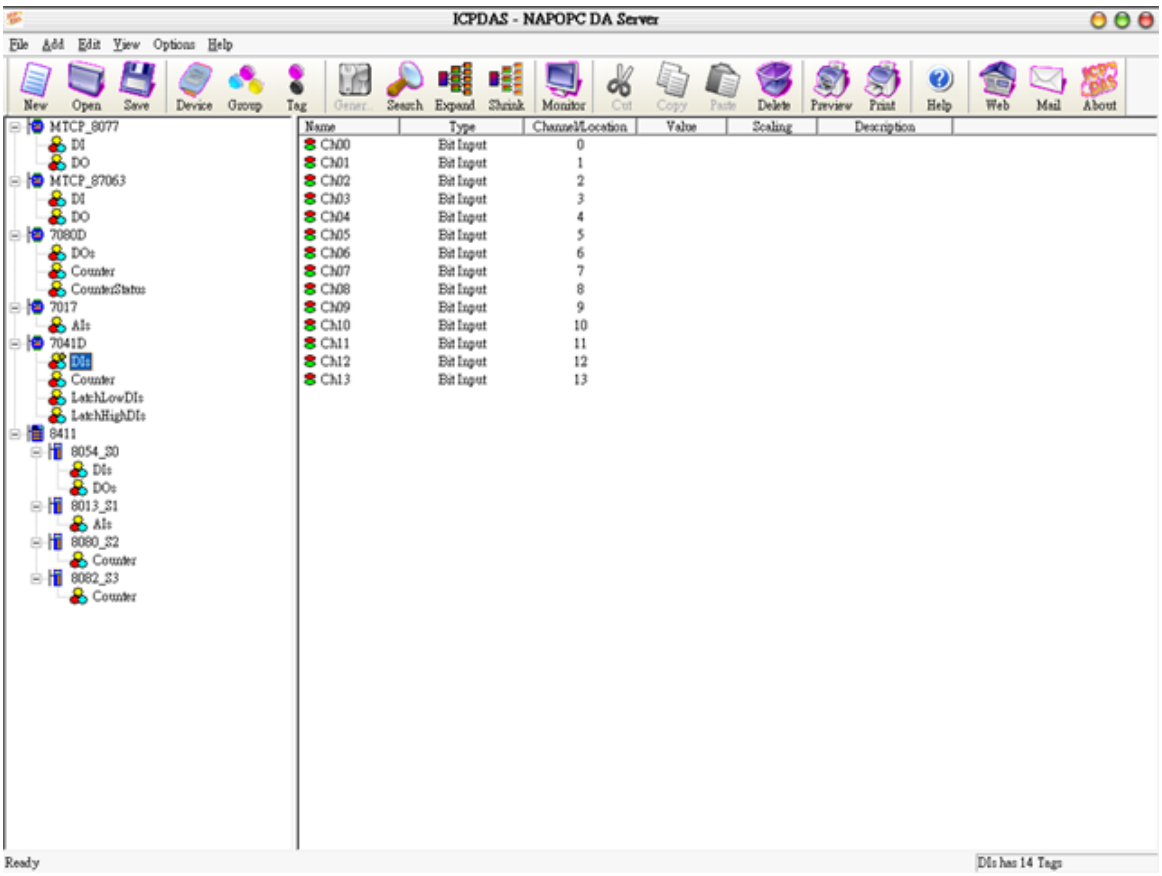
Step 14: The window displays the current value of the AI unit.



3.8 InduSoft

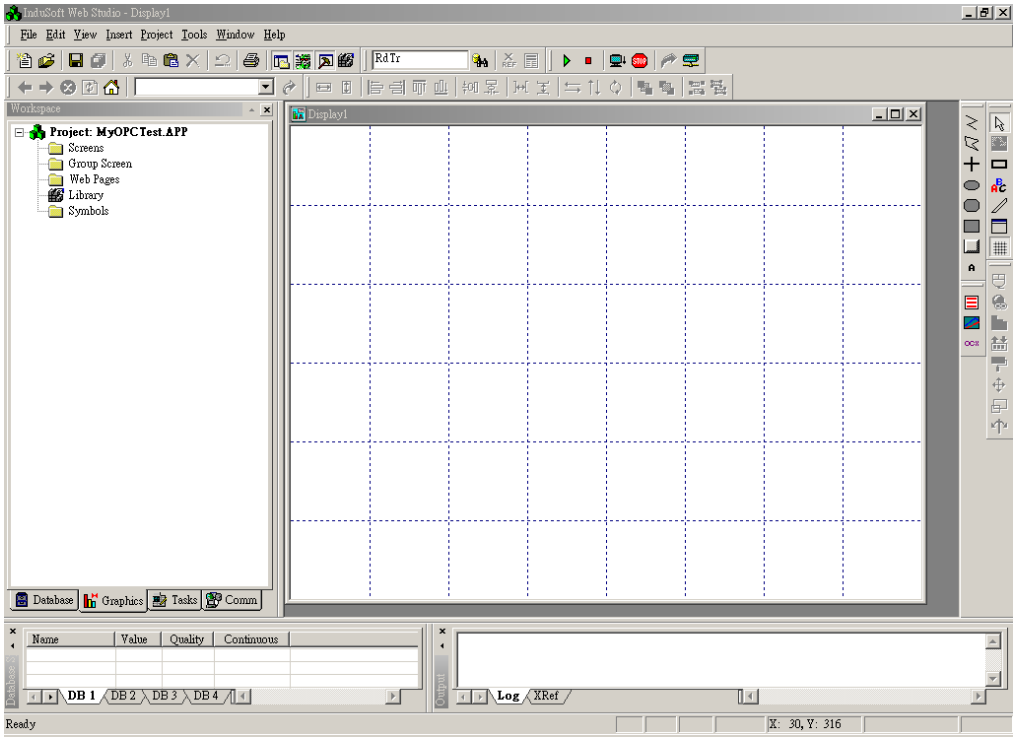
InduSoft Web Studio is a powerful, integrated collection of automation tools that includes all the building blocks needed to develop human machine interfaces (HMIs), supervisory control and data acquisition (SCADA) systems, and embedded instrumentation and control applications. Web Studio runs in native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

Step 1: Before using the InduSoft OPC Client module, you need to install and configure the OPC server in the machines you will run it.

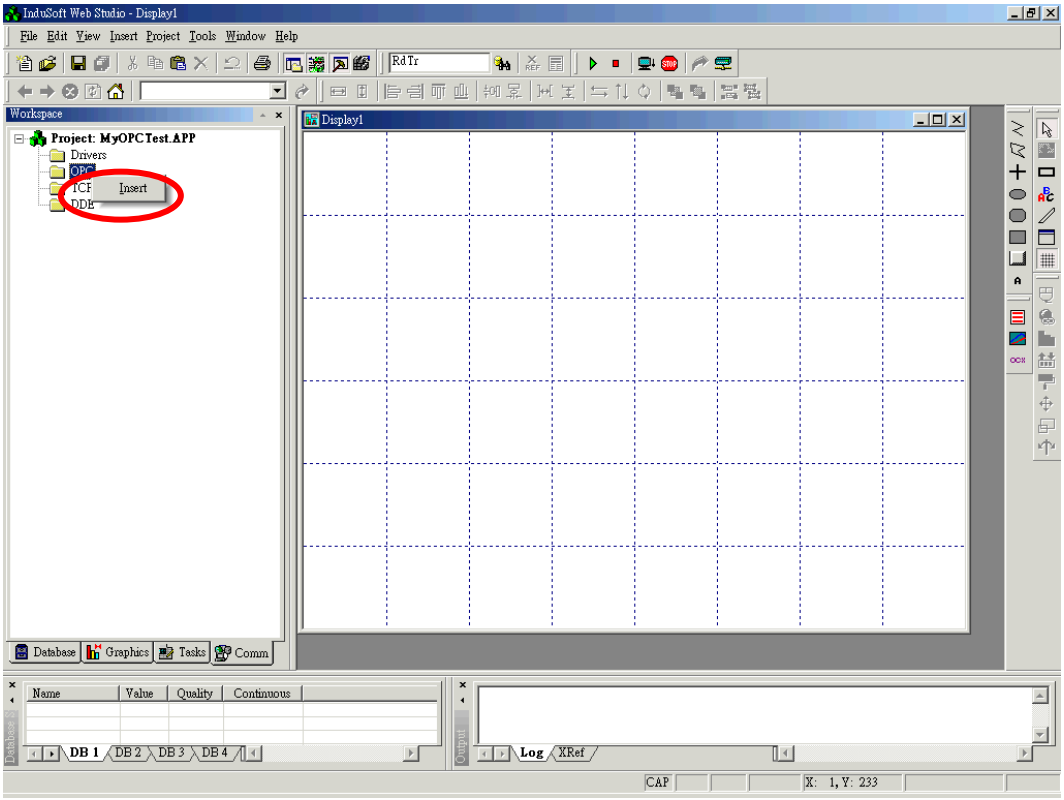


Step 2: Run the InduSoft (Version 4.1 or newer)

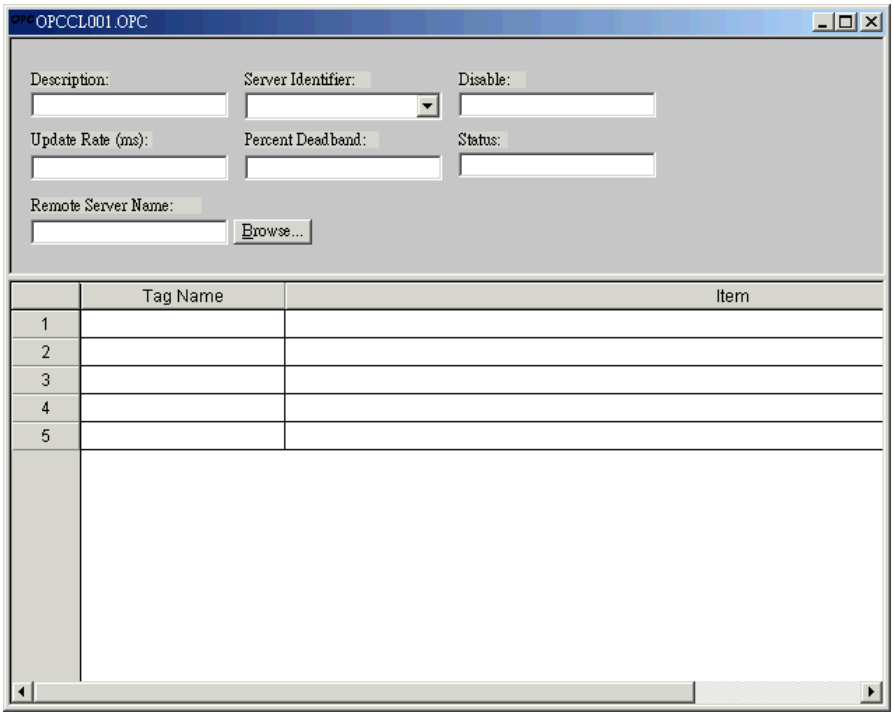
Step 3: Create the new project.



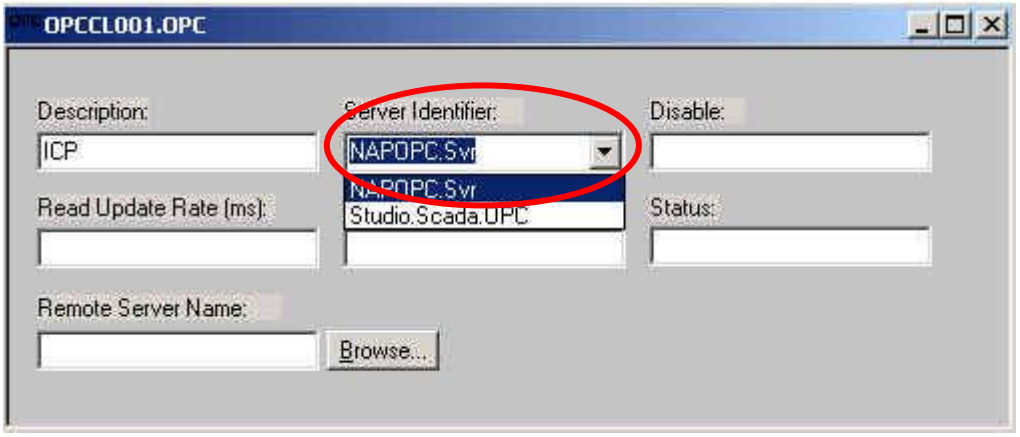
Step 4: In the Studio Workspace window, click the OPC tab, right-click the OPC folder, and click "Insert":



Step 5: OPC Attributes window pops up.



Step 6: Click on the Server Identifier: drop-down menu and select the "NAPOPC.Svr".

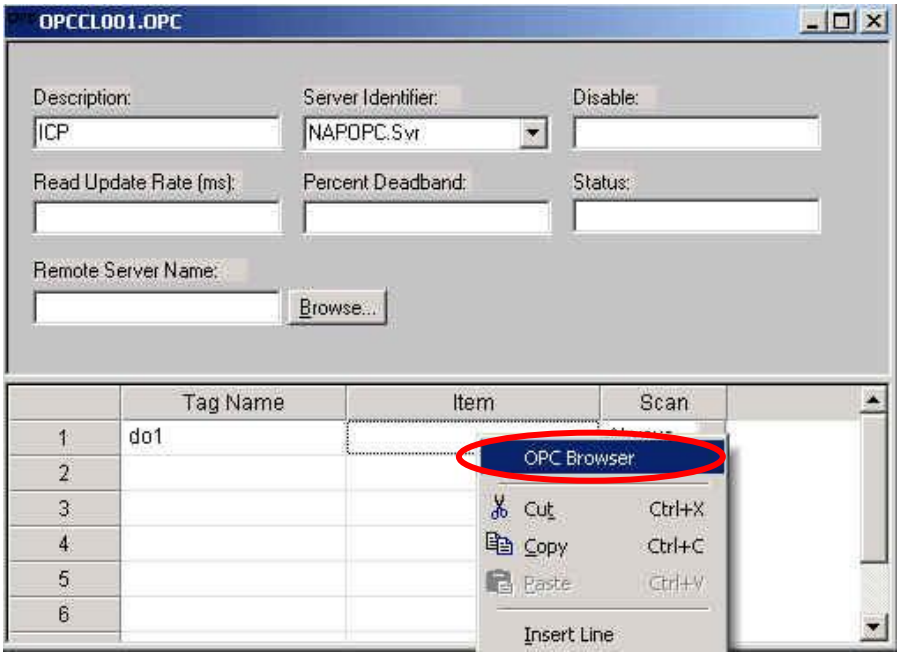


The configuration table for OPC has the following entries:

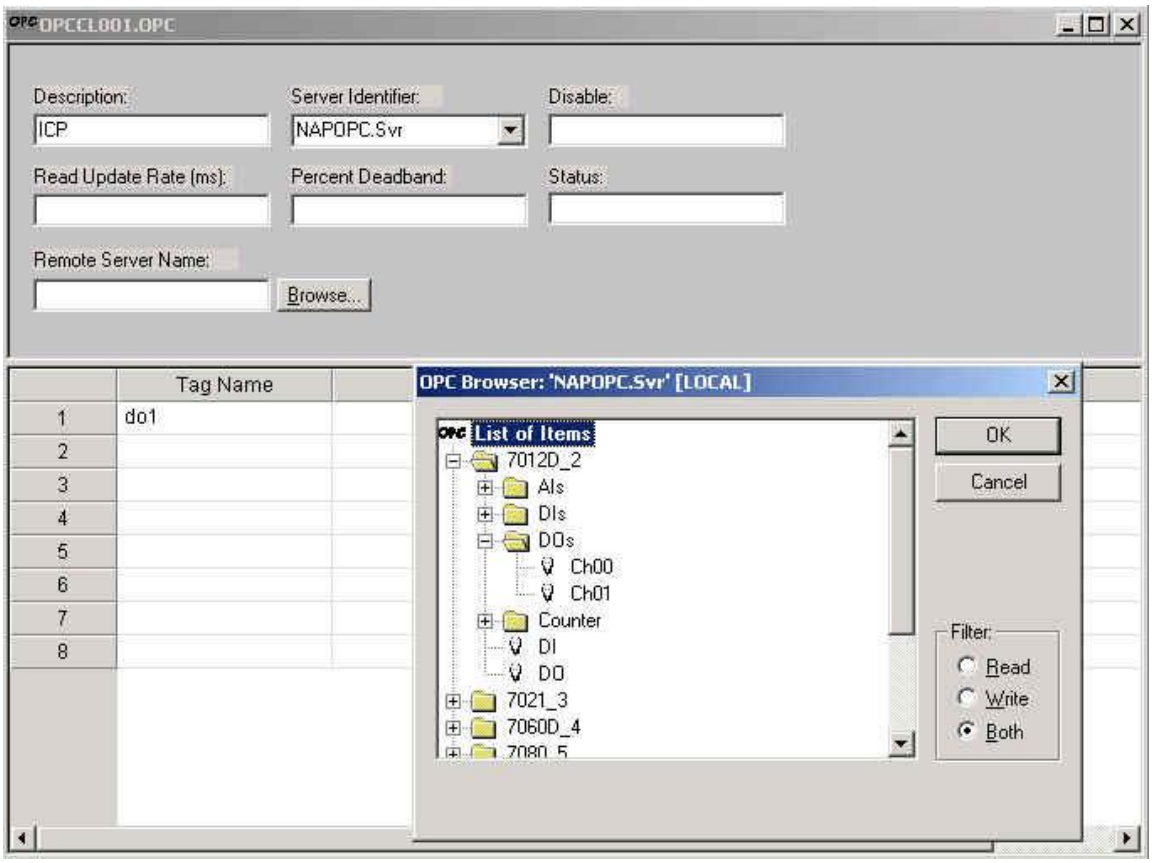
- Description: this field is used for documentation only. The OPC Client module ignores it.
- Server Identifier: this field should contain the name of the server you want to connect. If the server is installed in the computer, its name can be selected through the list box.
- Disable: this field should 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 can right-click it to get a menu.

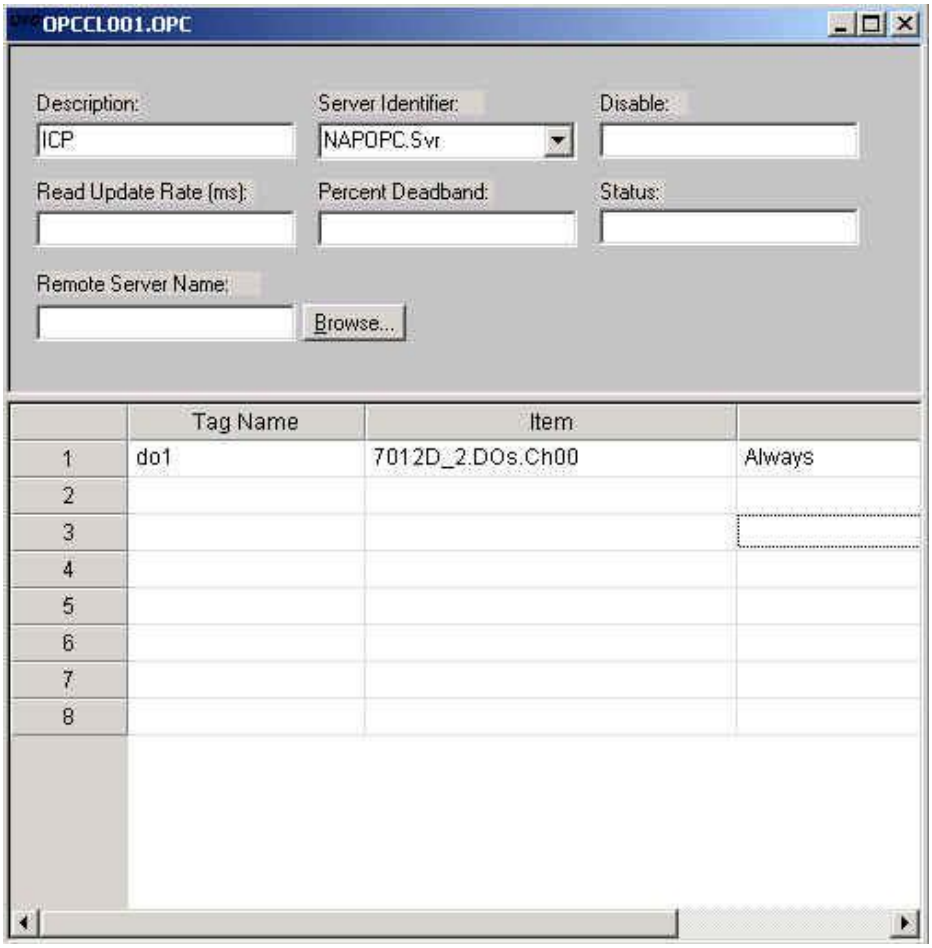


Step 9: Click the OPC Browser to appear the OPC Browser window.



Step 10: Select an item(tag) in the tree-view.

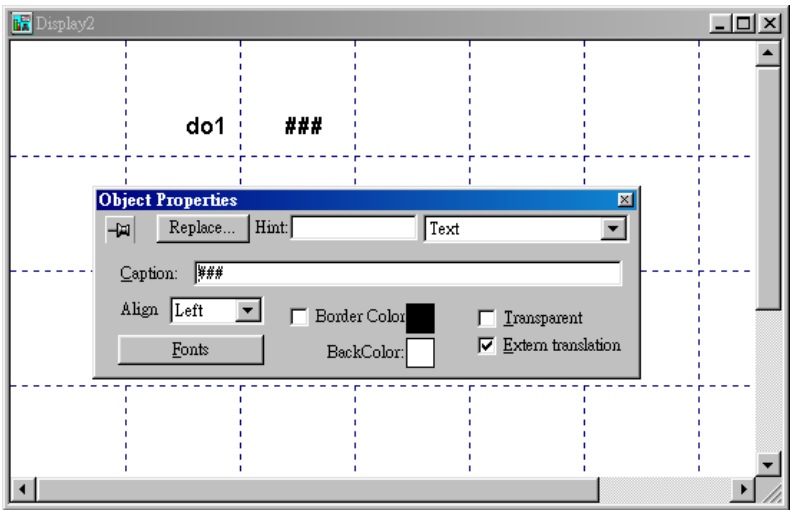
Step 11: Click the "OK" button to add this one.



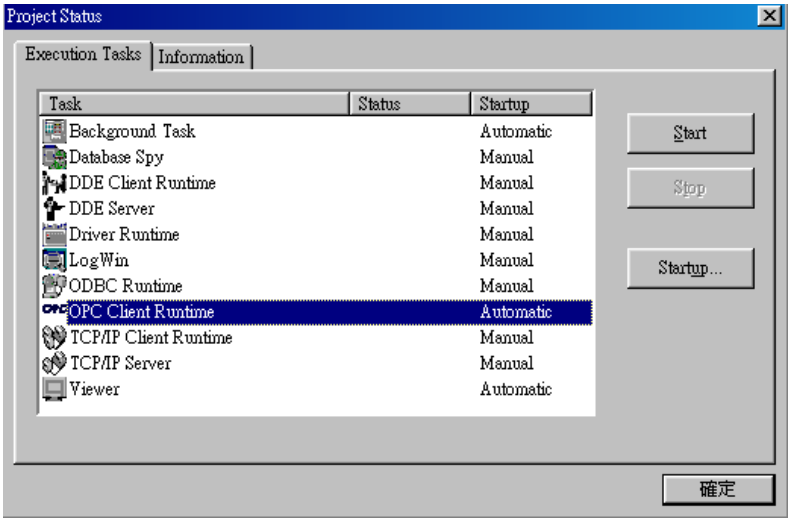
Step 12: Repeat the step between 7 to 11 to add more tags.

Step 13: Creating a Text String for the Input/Output Dynamic. Click the Text icon on the Object Editing toolbar. Position the crosshairs in the Display2.scr. Press the"#" key three times to display "###" in the gray square.

Step 14: 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.

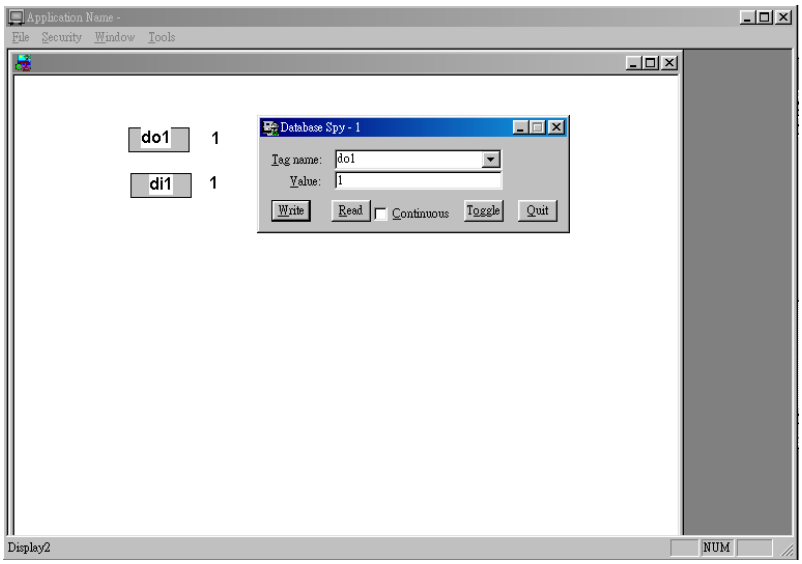


Step 15: From the Project select status. Then select the OPC Client Runtime in "Execution Tasks" tab. Click on the Startup button to setup the Startup as Automatic.



Step 16: Run the program InduSoft OPC Client Runtime module automatically or by the menu "Project->Status". After running this program, a small icon will appear in your system tray. To close the InduSoft OPC Client module, right-click its icon in the system tray, and select "Exit".

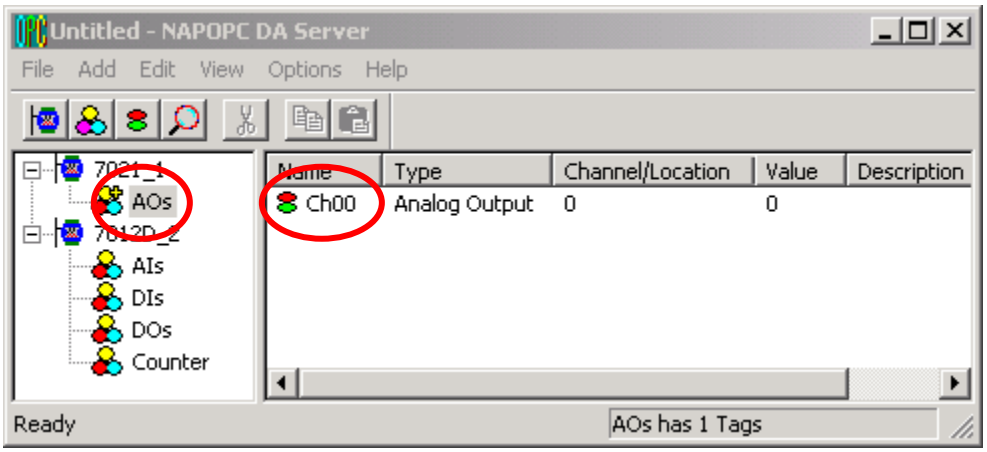
Step 17: Database Spy allows you to monitor and forces application tags, reading and writing to the database. You can find it in Tools menu.



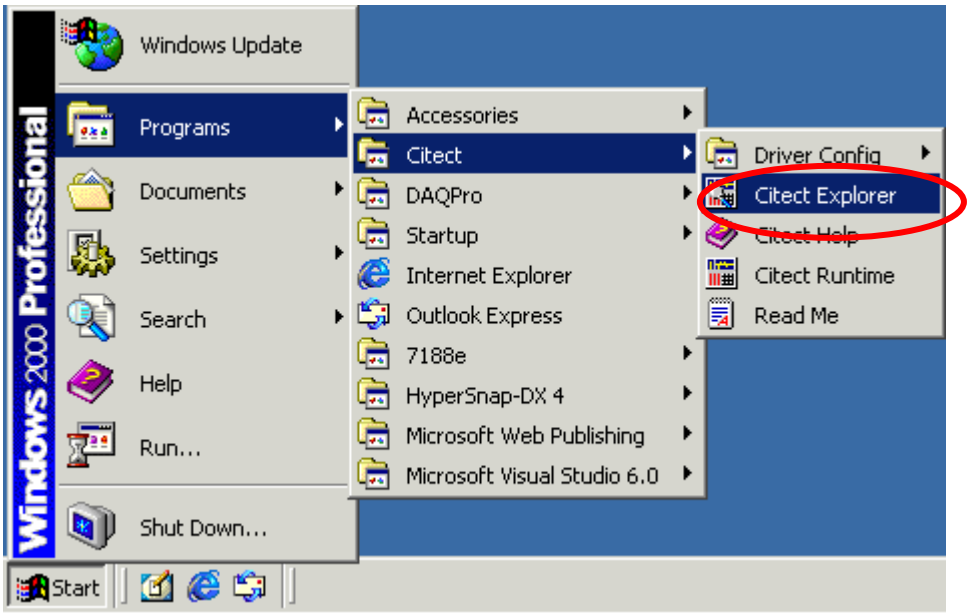
3.9 Citect SCADA

CitectSCADA is a reliable, scaleable and high performance SCADA system that includes over 100 drivers and free development software. Used in a wide range of industries, CitectSCADA enables users to reduce costs by optimizing process operations. Furthermore, it not only reduce risk with built-in redundancy for servers, networks and communications, but open data connectivity via OPC client & server, OLE DB, ODBC, DDE and API as well as over a hundred native drivers. CitectSCADA can implement in Windows. Visit http://www.citect.com for more information about CitectSCADA

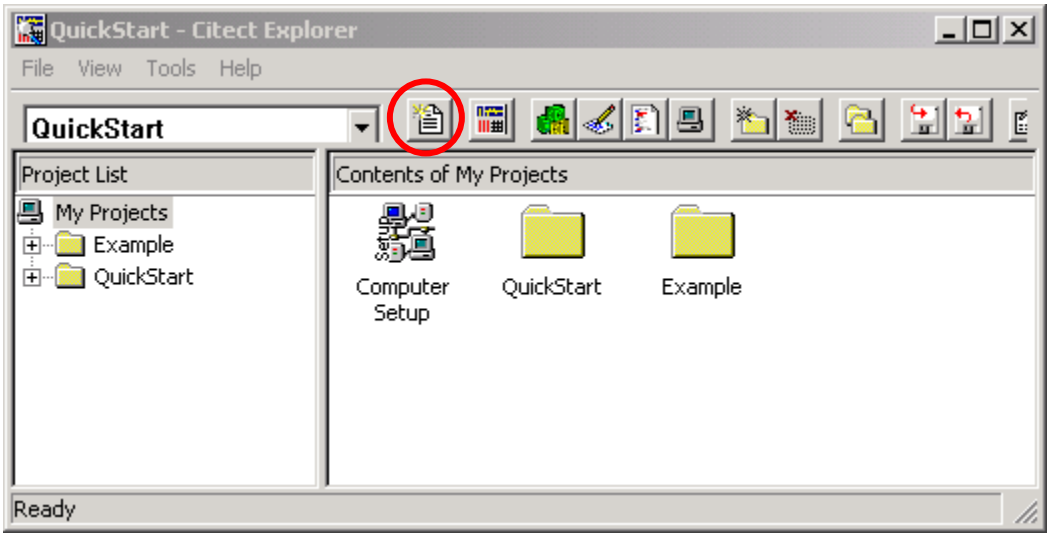
Step 1: Before using the CitectSCADA, you need to install and configure the OPC server in the machines you will run it (see Chapter 1).



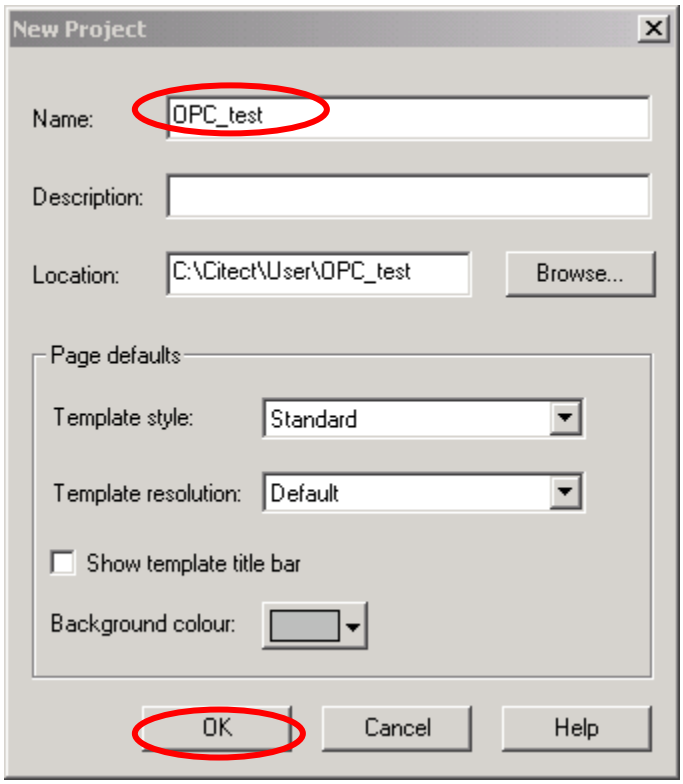
Step2: Start up the CitectSCADA with version 5.40.



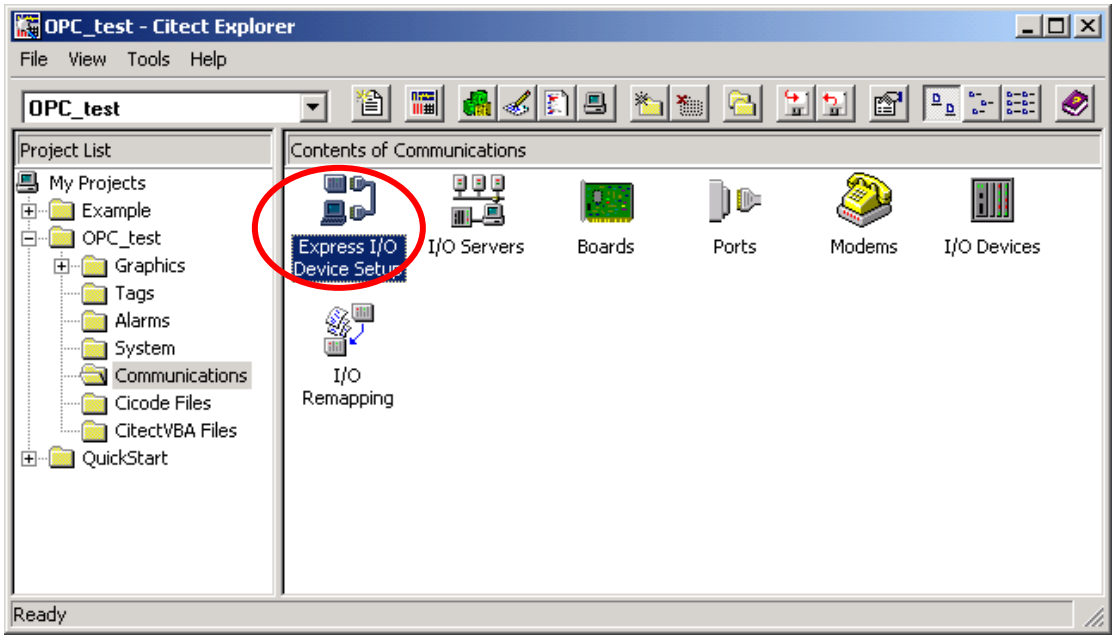
Step3: Left click the "page-marked button" or select "File / New Project…" from the CitectSCADA window menu to build a new project in the CitectSCADA.



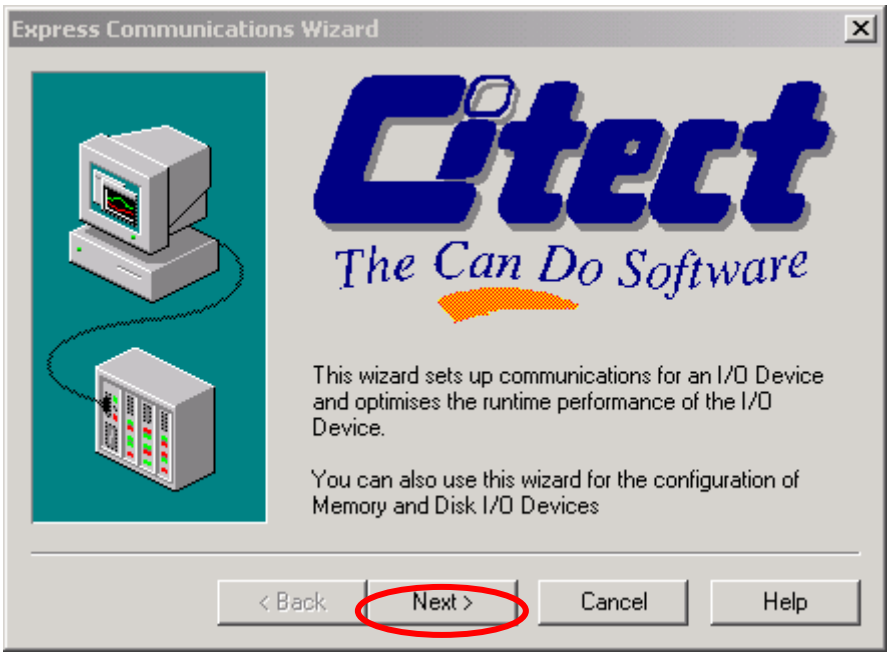
Step4: Fill a name of new project in the blank and then click "OK button" to finish this process.



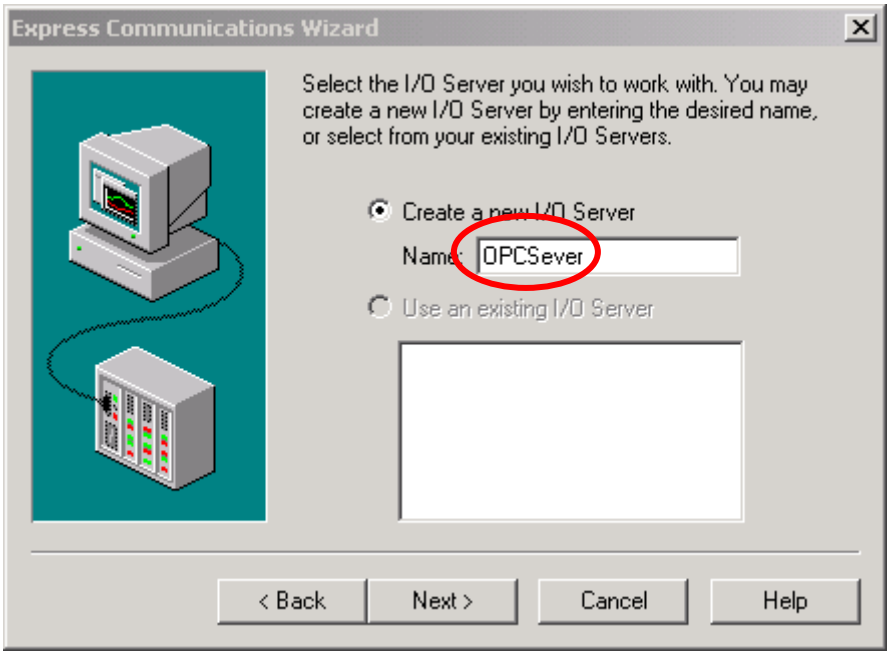
Step5: Click the "Express I/O Device Setup" icon to set all communication parameters.



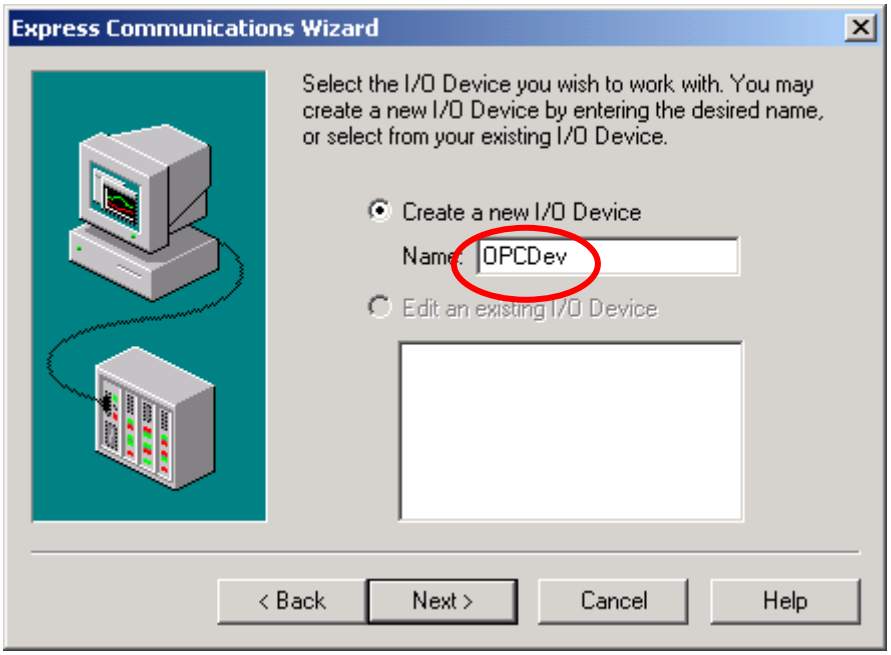
Step6: Start up the "Express Communications Wizard Dialog".



Step7: Create a new I/O Server and define a name called "OPCServer" for that one.

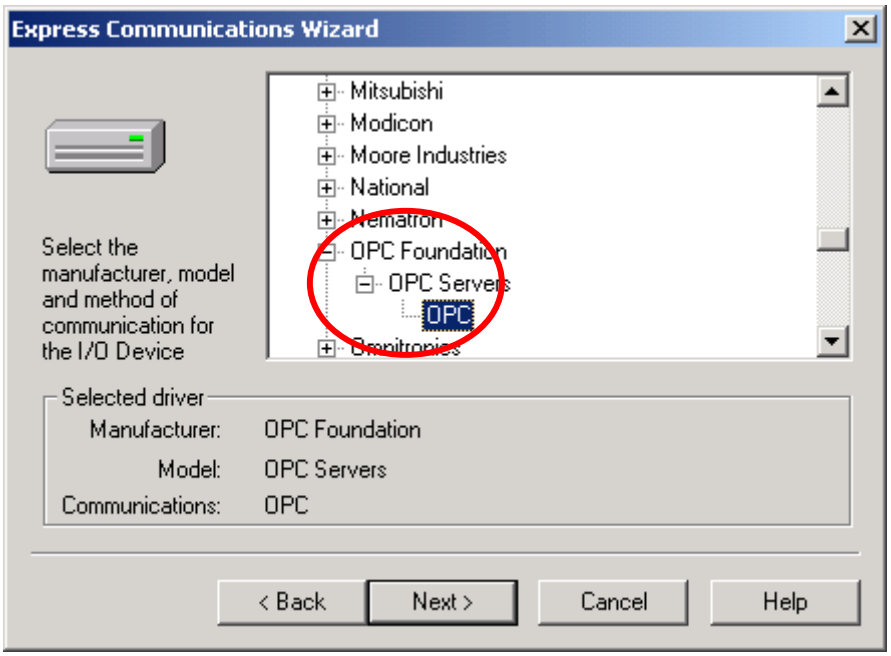


Step8: Create a new I/O Device under the I/O Server that created previously and define a name called OPCDev for that one.

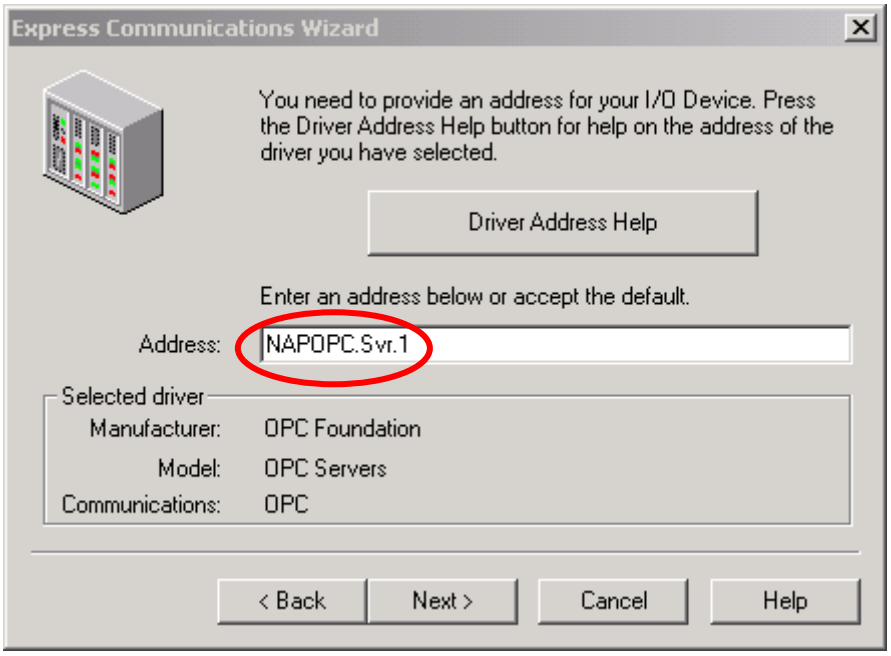


Step9: Select "External I/O Device" to be the type of OPCDev I/O Device.

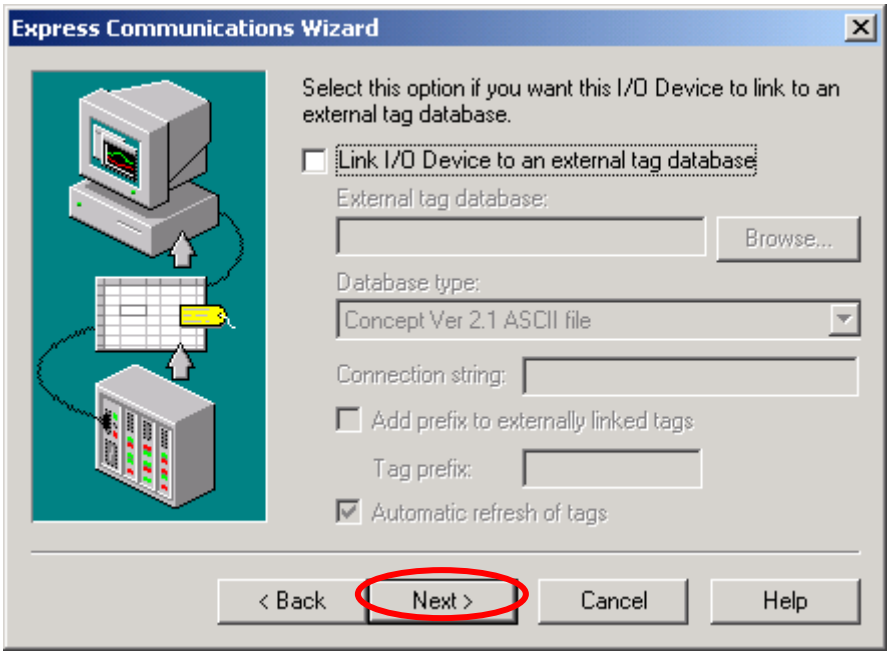
Step10: Set OPC to be the method of communication for OPCDev I/O Device.



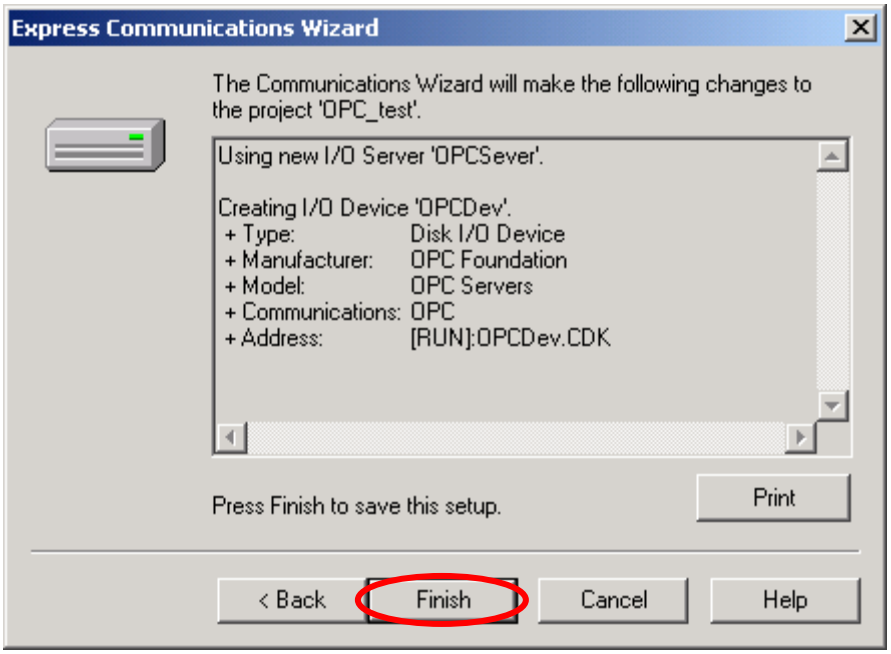
Step11: Set Address to be "NAPOPC.Svr.1" for OPCDev I/O Device.



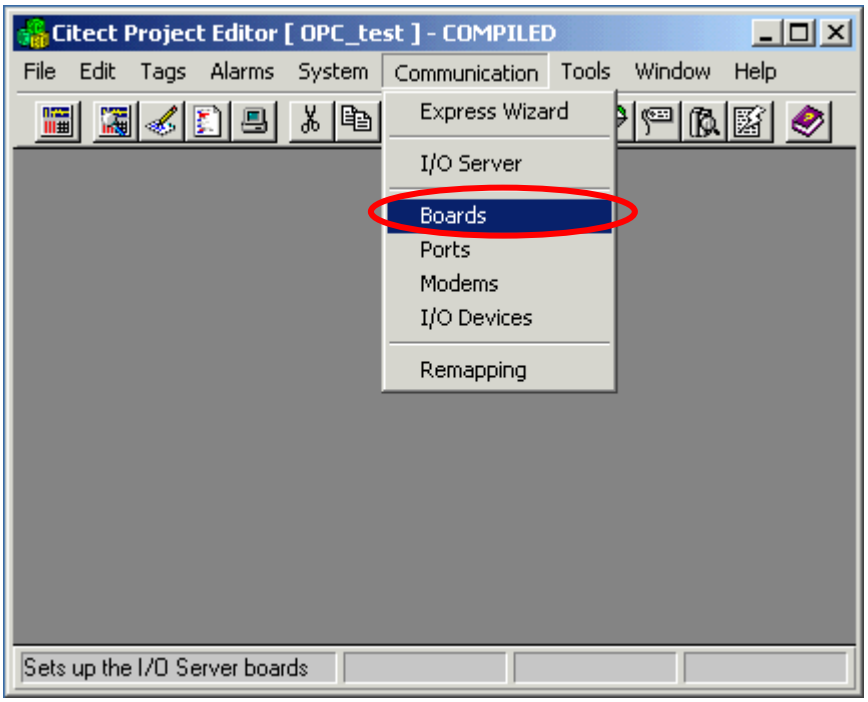
Step12: Do not set any parameter in this step.



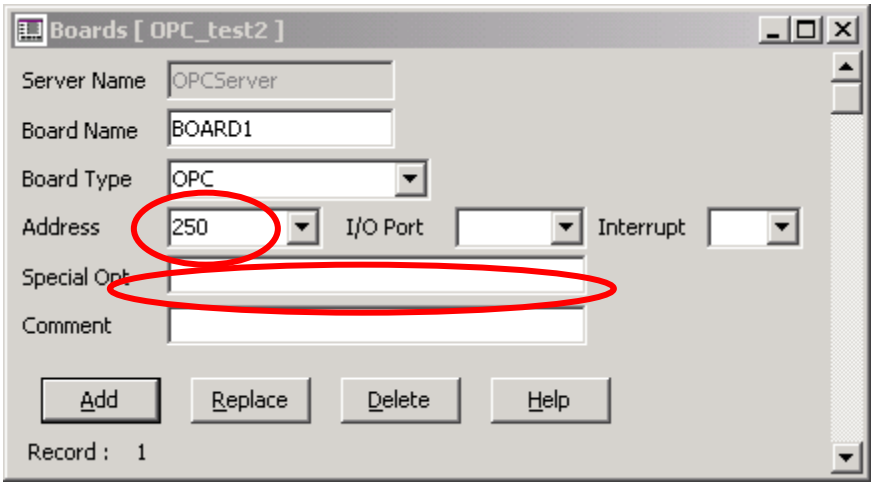
Step13: Click the "OK button" to finish the setting of communication parameters.



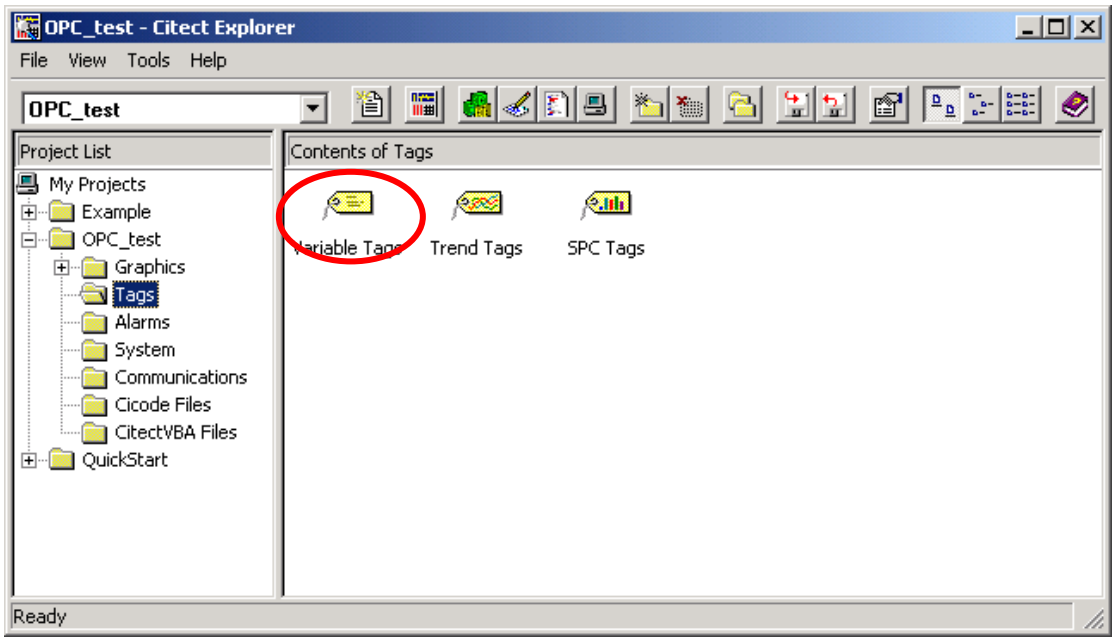
Step14: Open the "Citect Project Editor window" to edit Boards parameters.



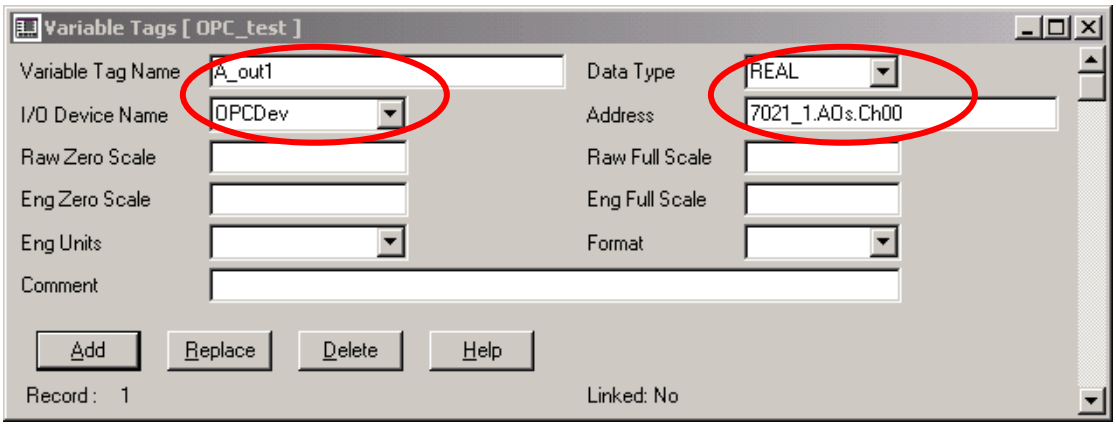
Step15: Edit Boards parameters. The Address (scanning period) is set to be "250ms", and the Special Opt is set to be blank.



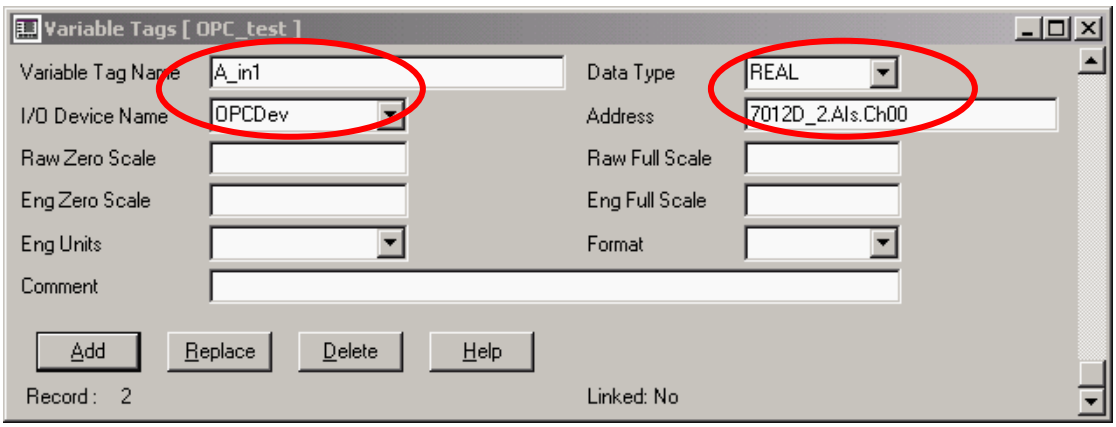
Step16: Define two Variable Tags.



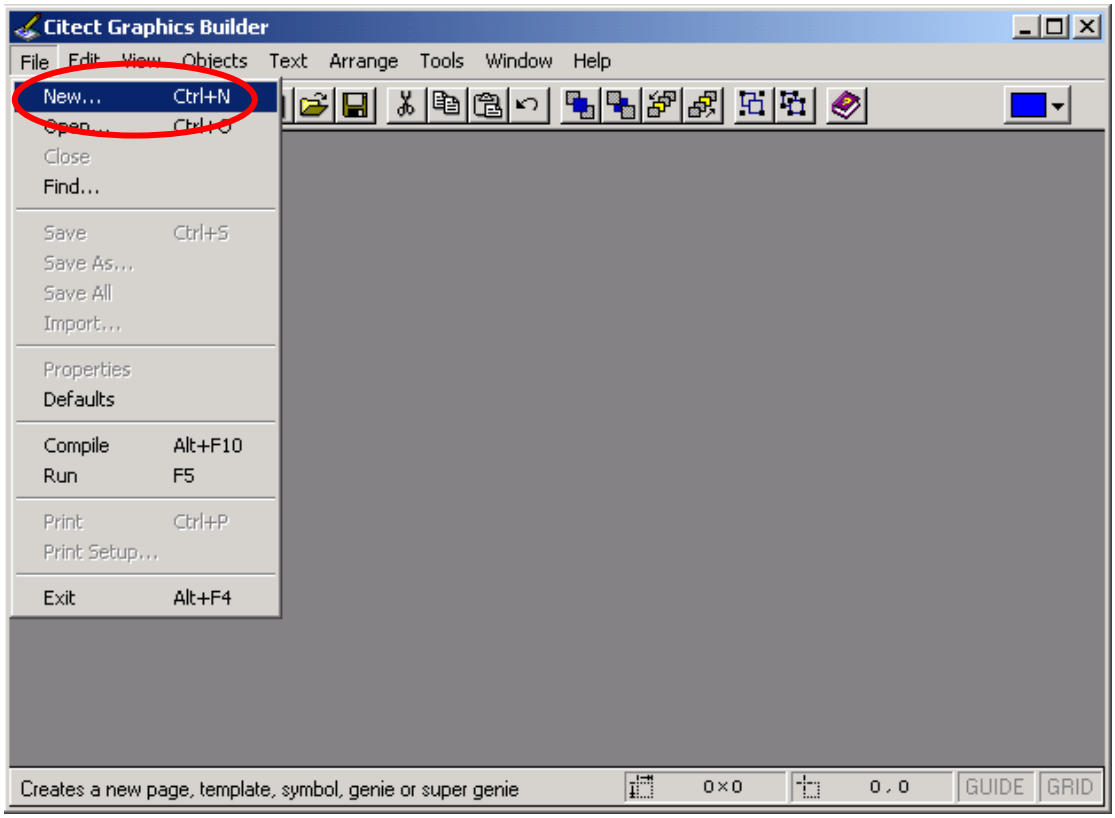
Step17: Define a variable tag for analog output of the modules 7021. The Variable Tag Name is "A_out1", the Data type is "REAL", the I/O Device Name is selected to be "OPCDev", and the Address is "7021_1.AOs.Ch00".(see Step 6)



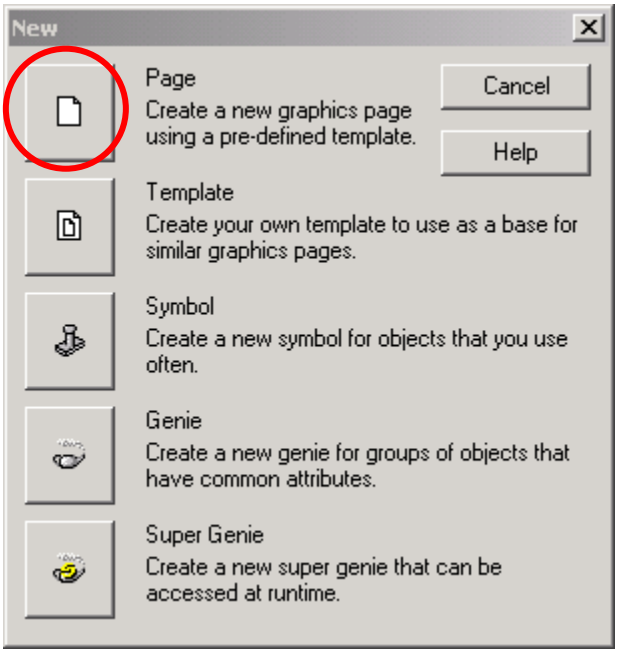
Step18: Define another variable tag for analog input of the modules 7012D. The Variable Tag Name is "A_in1", the Data type is "REAL", the I/O Device Name is selected to be "OPCDev", and the Address is "7012D_2.AIs.Ch00". (See Step 6)



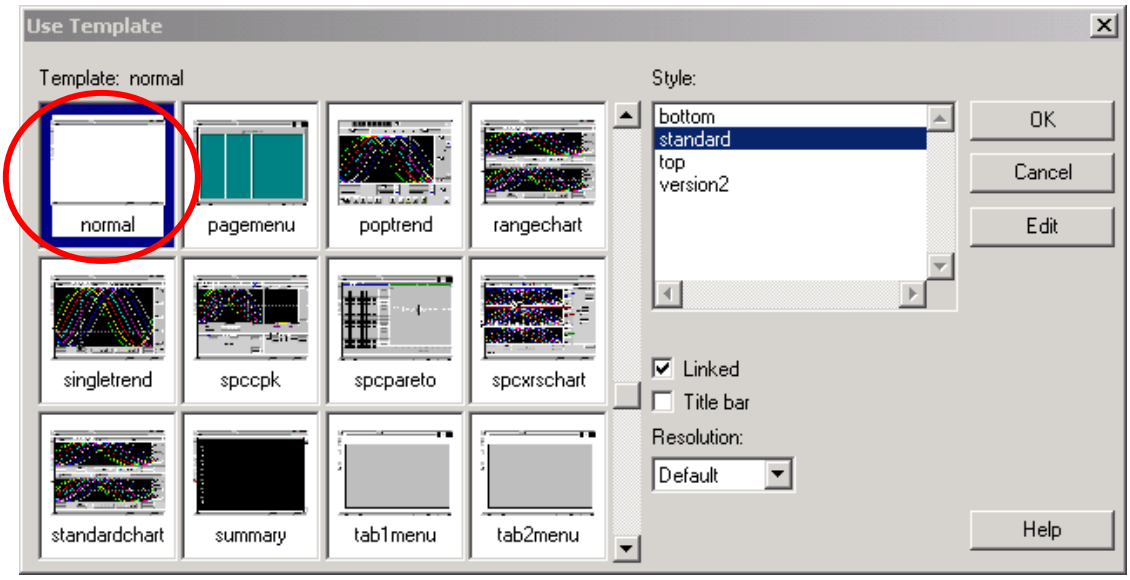
Step19: Create a new page in the "Citect Graphics Builder window".



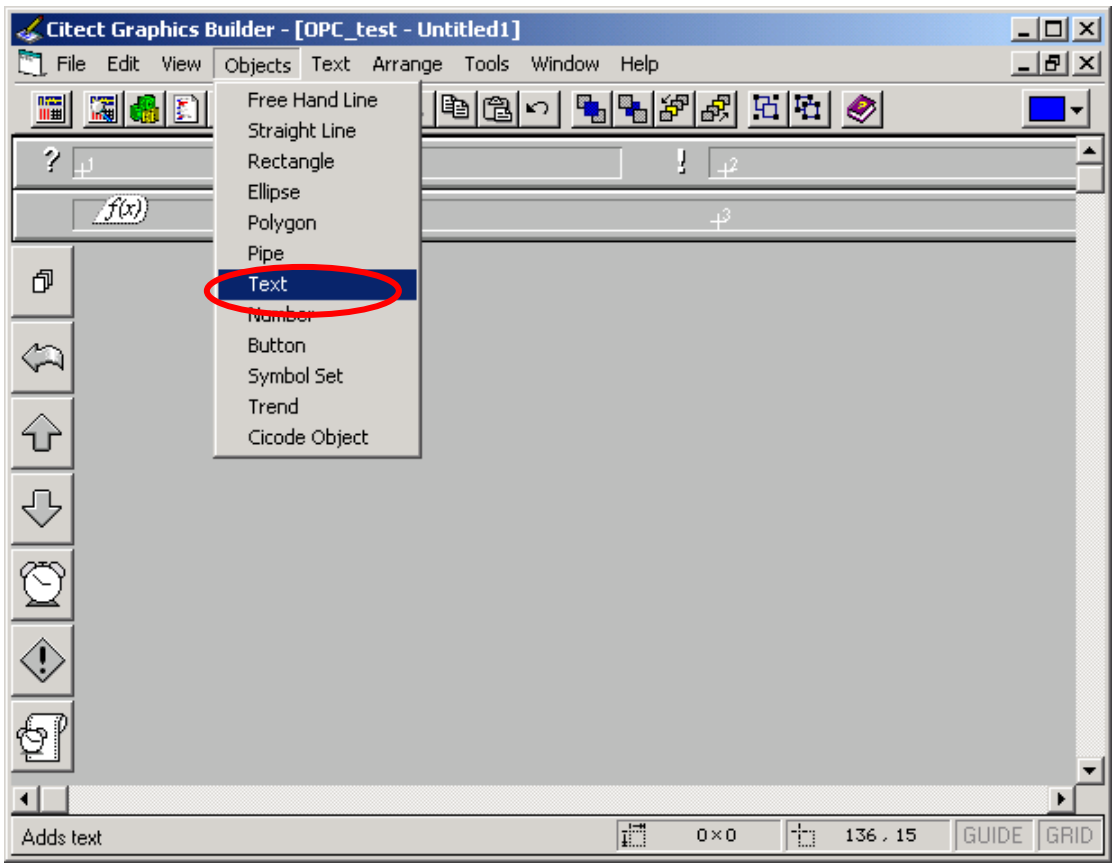
Step20: Click the "page-marked button" to create a new page.



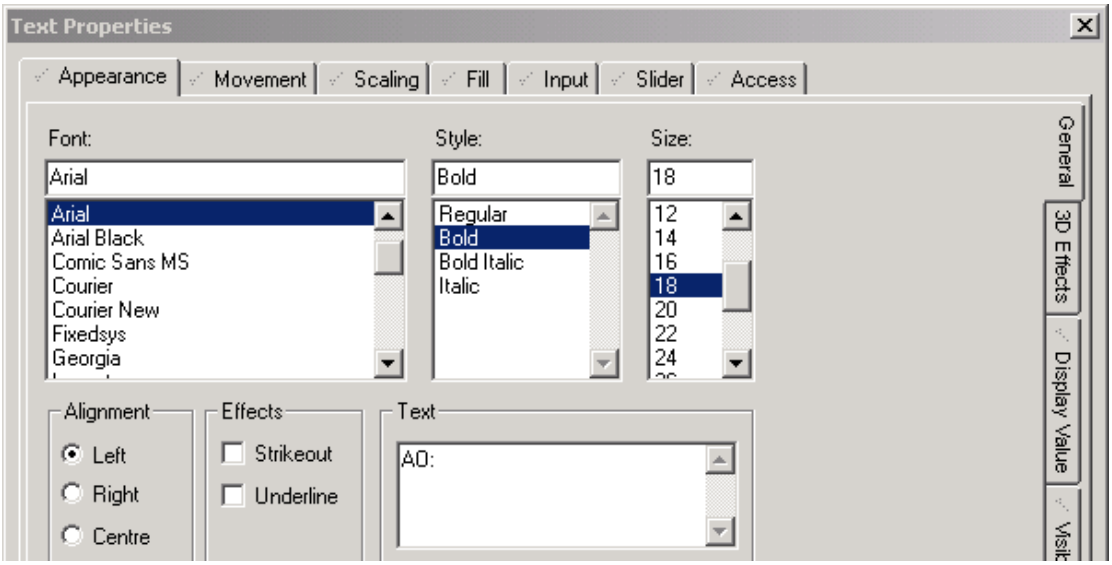
Step21: Select normal template to be the background and function of this page.



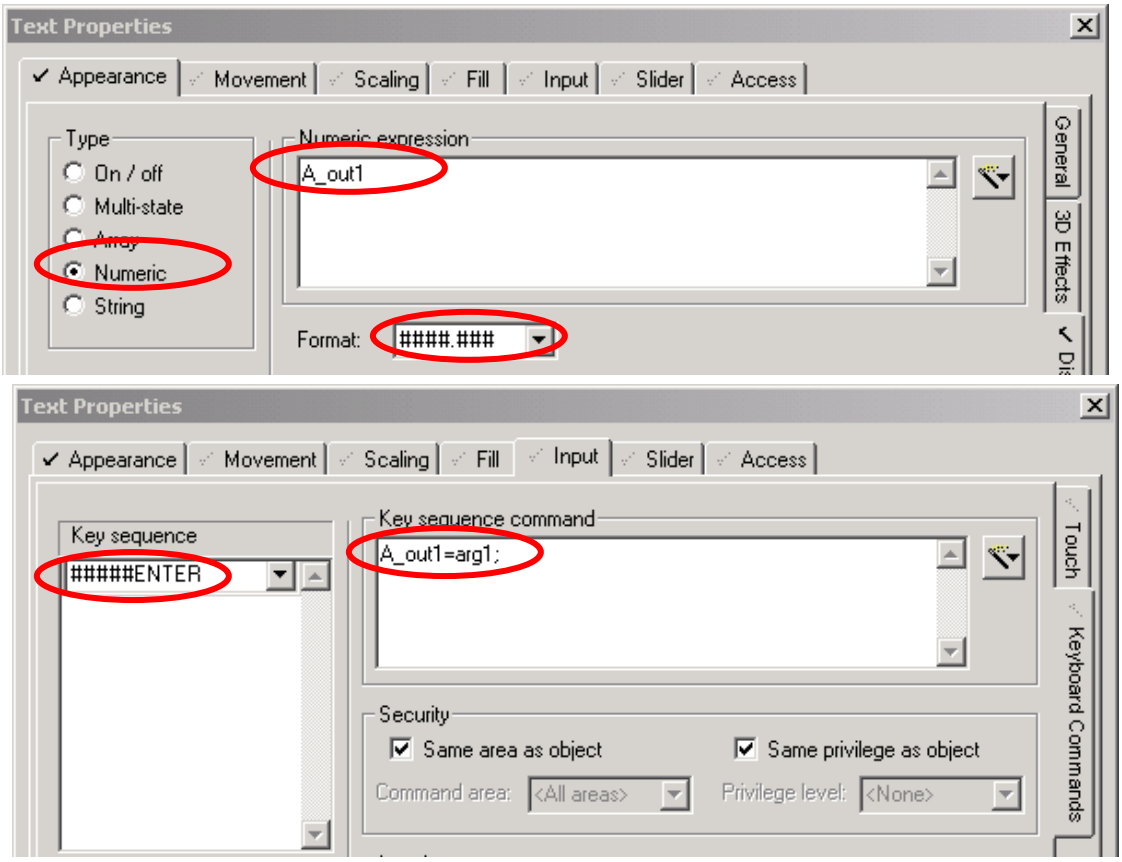
Step22: Select "Objects /Text " from "Citect Graphics Builder window menu" to insert a "Text Object" on the page.



Step23: Key-in the words "AO:" in the Text object. Then, left click to put the Text object on the page and set "Appearance parameters" of Text object.

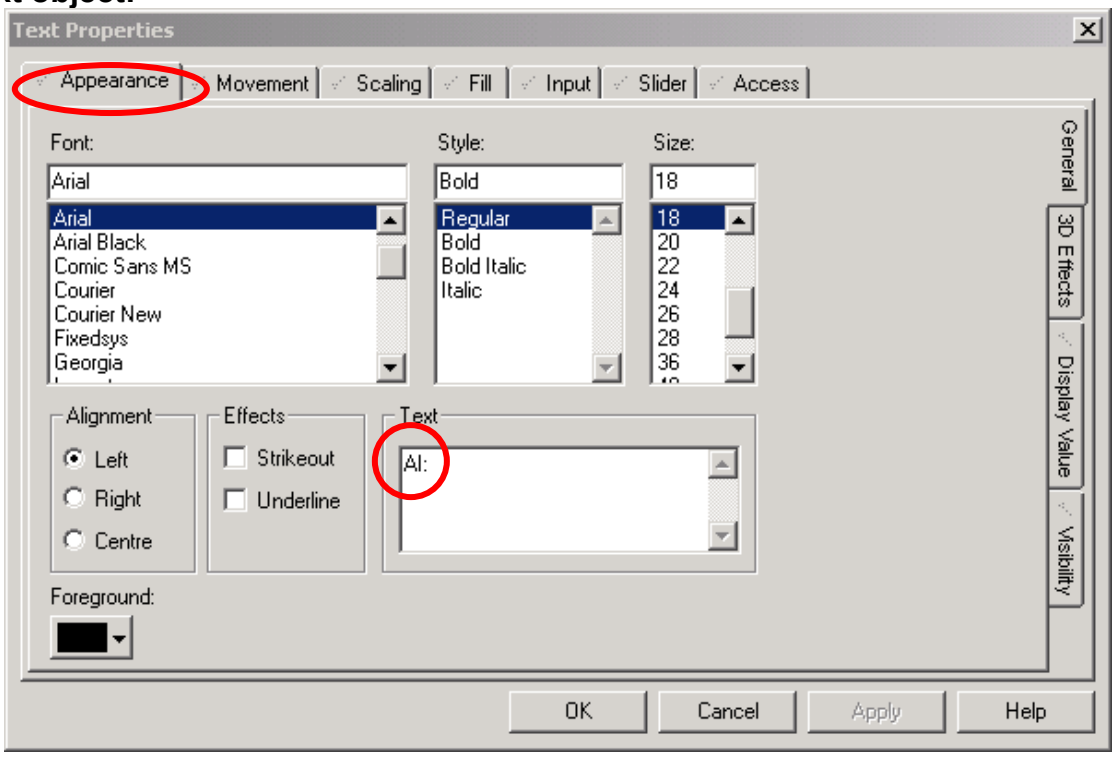


Step24: Select "Objects/Number" from "Citect Graphics Builder window menu" to insert a "Number Object" on the page. Left Click to put Number object on the page. Then, set "Appearance attributes" and "Input attributes" for this Number object.

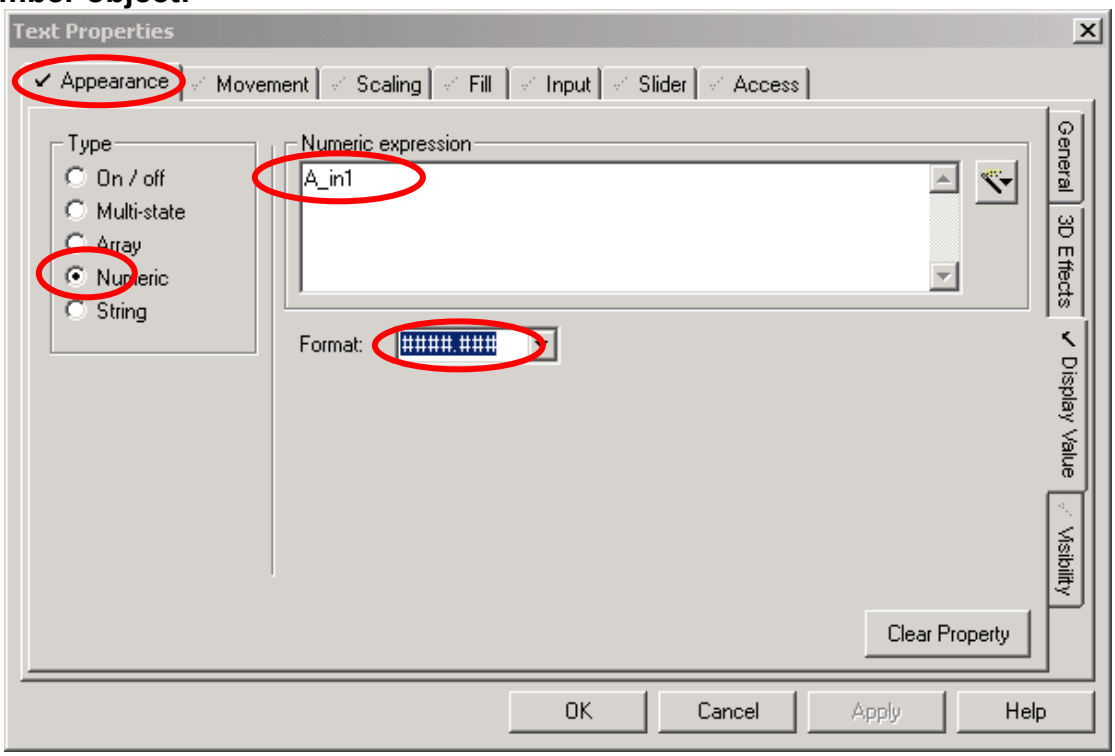


Step25: Set another Text object and Number object by the same way, and the "Appearance attributes" of Text and Number object are showed below.

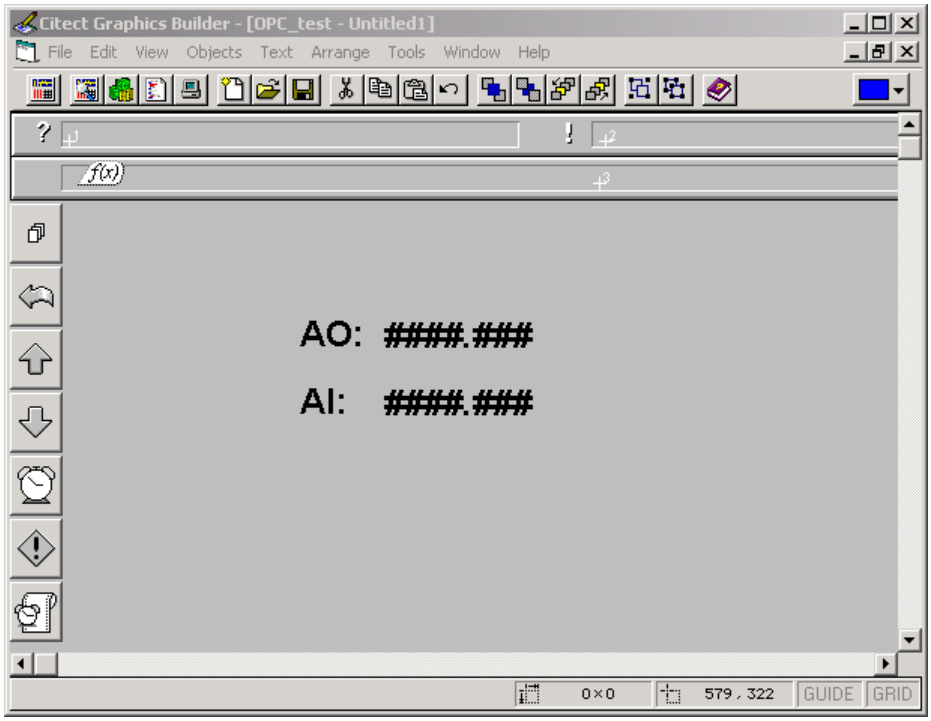
Text object:



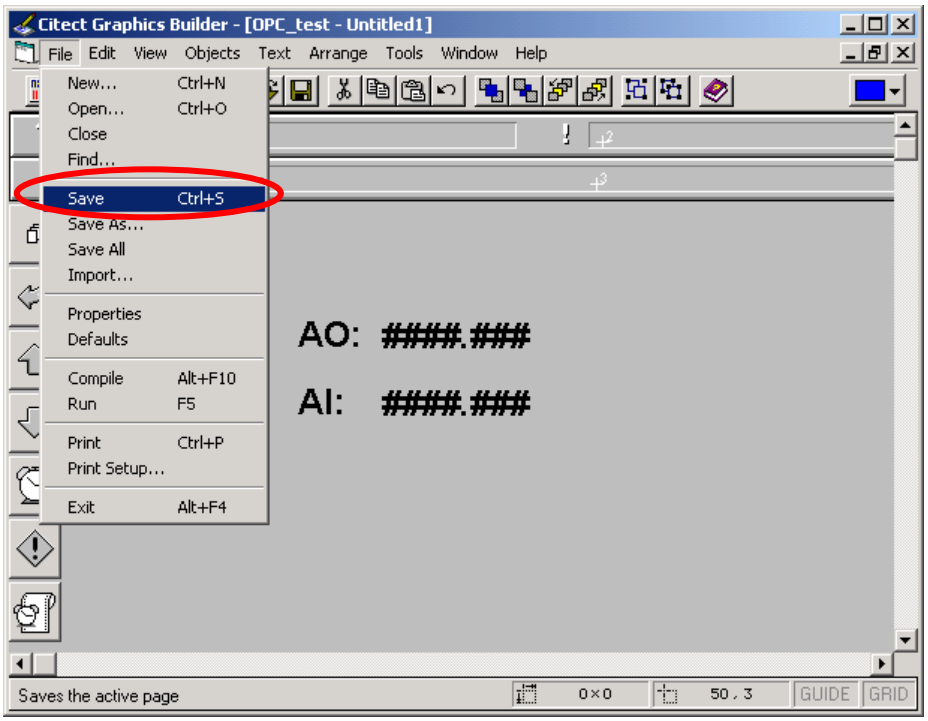
Number object:



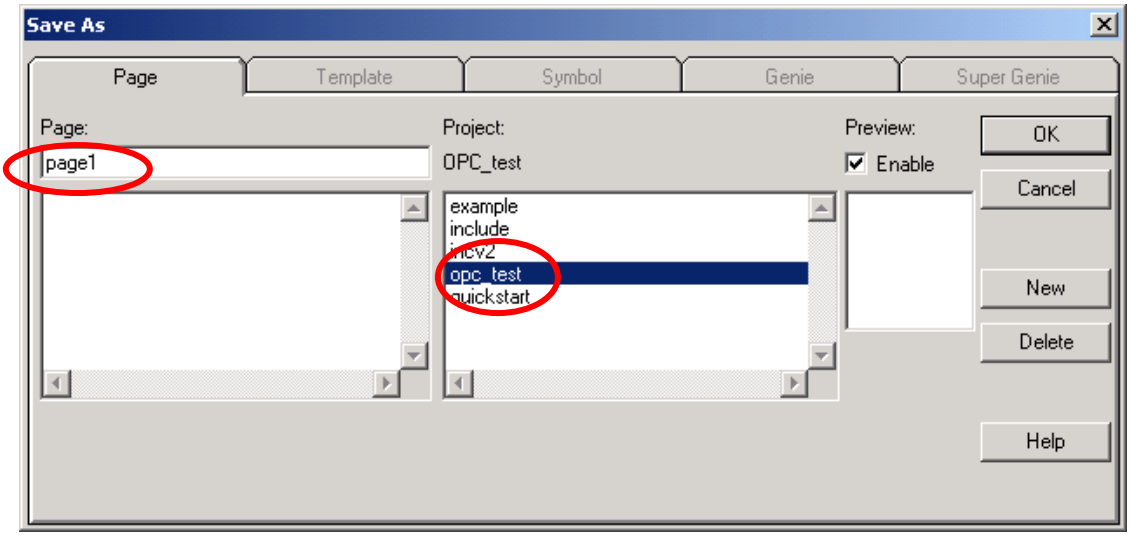
Step26: When finish the all object and attribute setting, the page is looked like as one, which shows below.



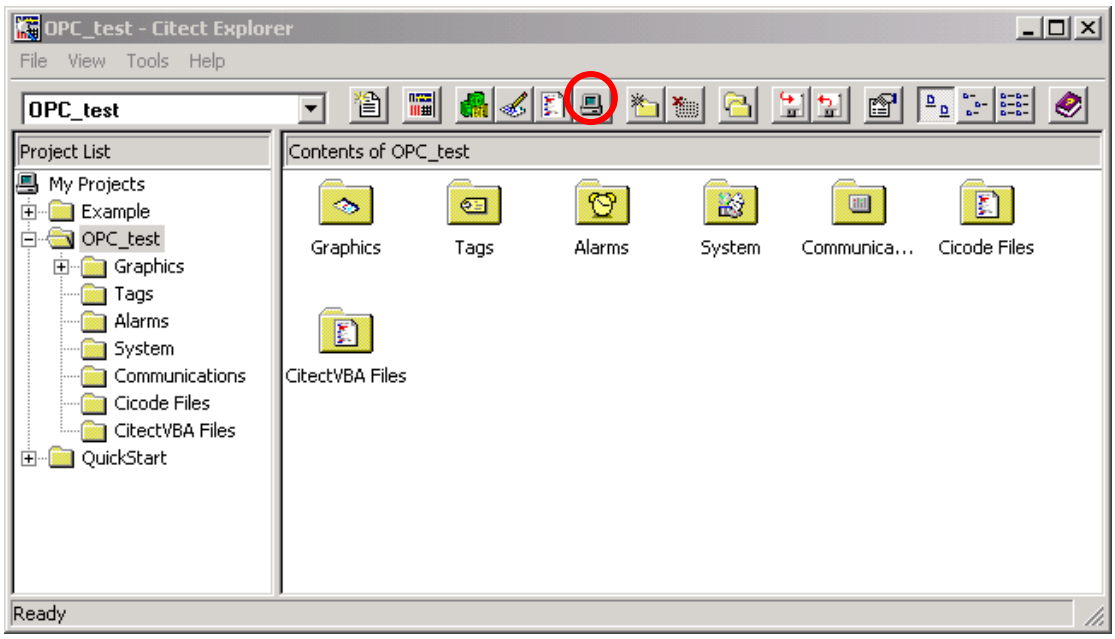
Step27: Select "File/Save " from "Citect Graphics Builder window menu" to save this page.



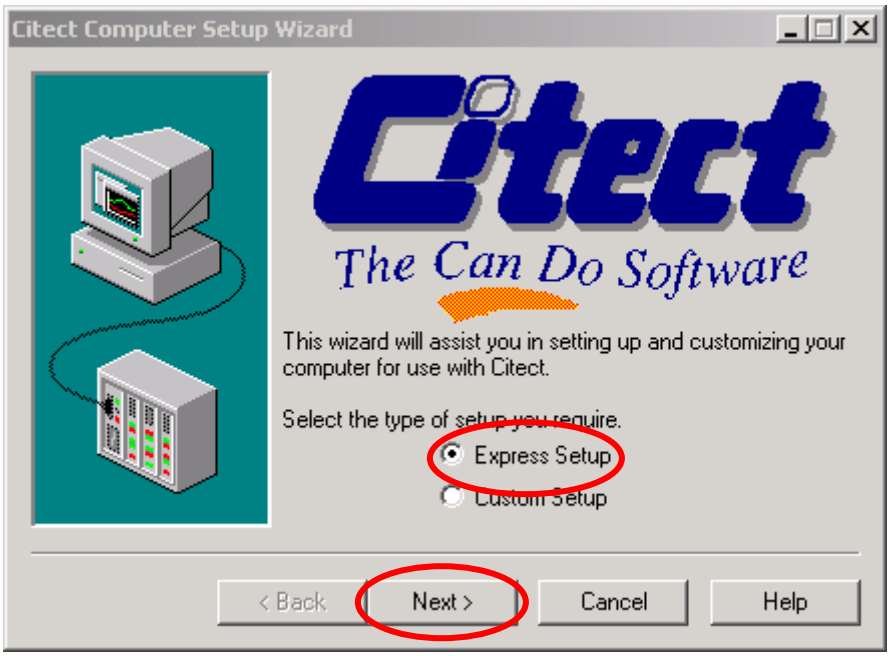
Step28: Fill the name of this page and save it under OPC_test project.



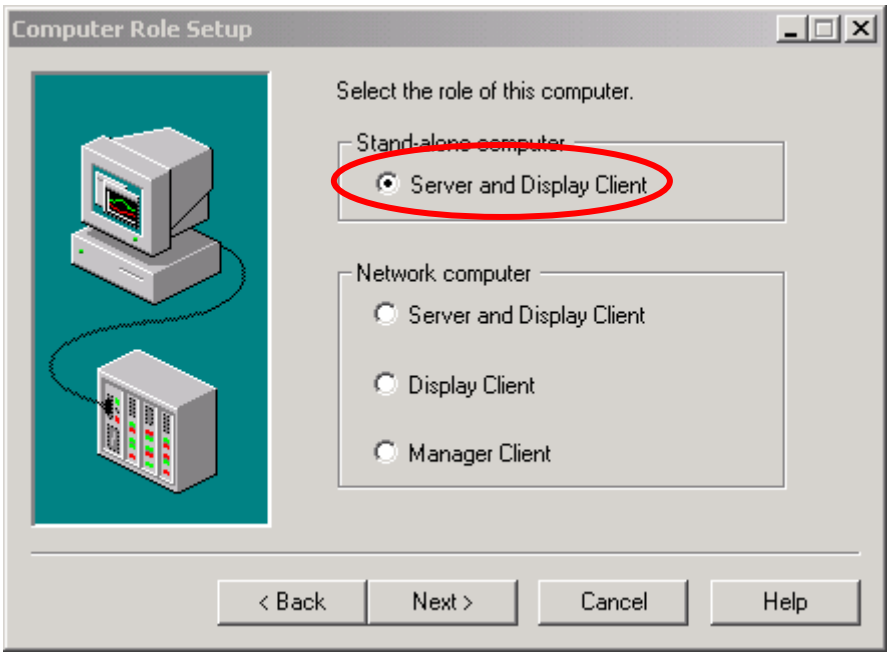
Step29: Left click the "computer-marked button" to define the role of this computer.



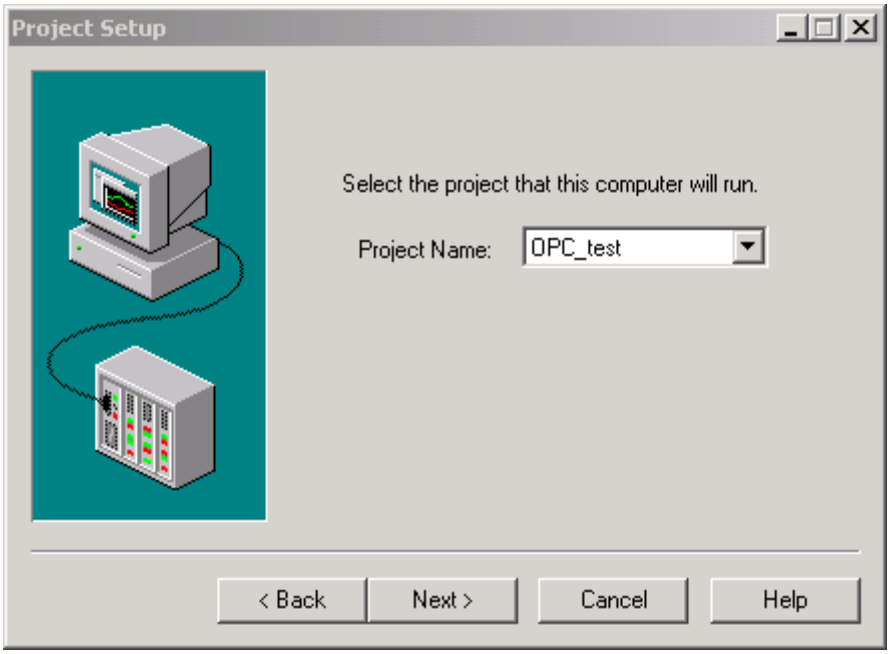
Step30: Start up the Citect Computer Setup Wizard.



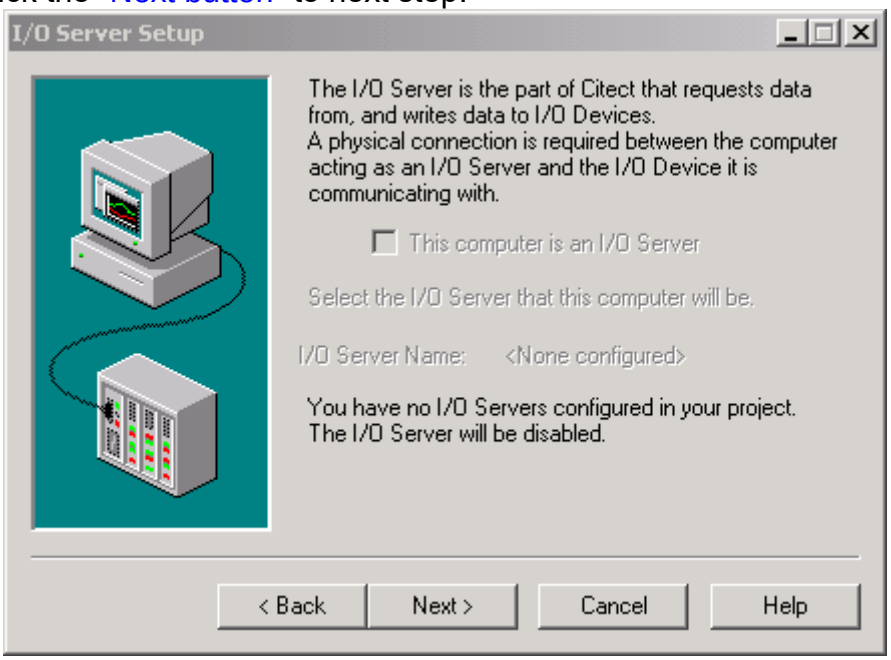
Step31: Select the "Stand-alone computer" item to be the role of this computer.



Step32: Select the project "OPC_test" to be the project that this computer will run.

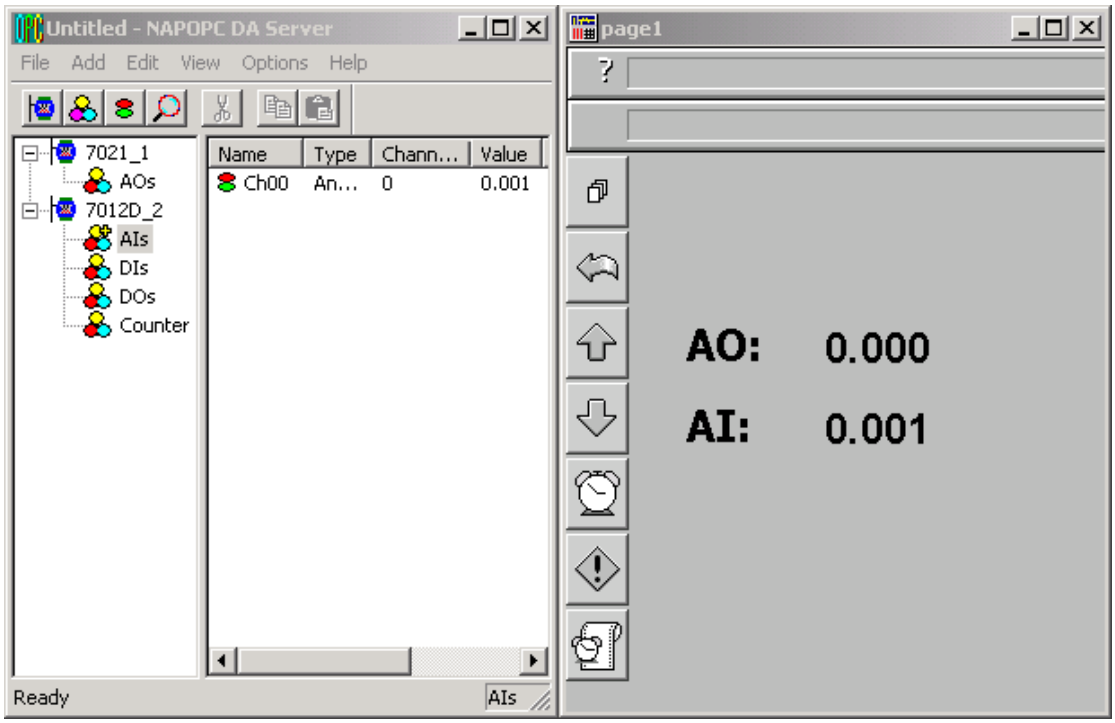


Step33: Click the "Next button" to next step.

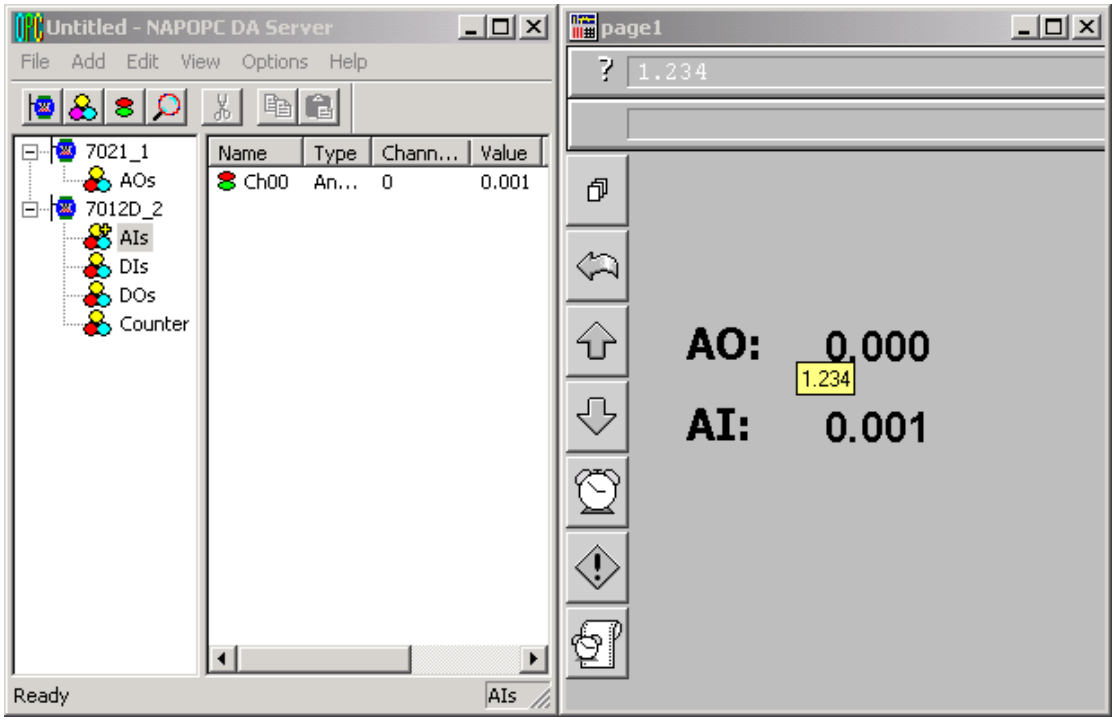


Step34: Left click the "Finish button" to finish the computer setup.

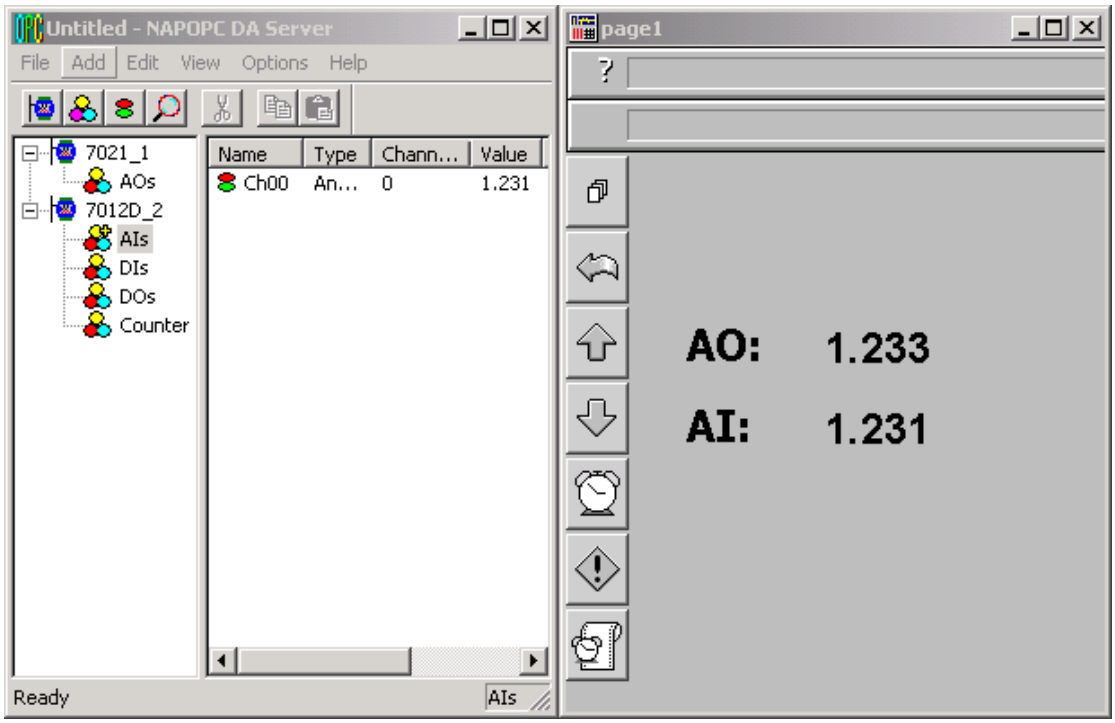
Step35: Select OPC_test project and press "F5" to run this project. Compare the NAPOPC Server monitor and CitectSCADA runtime window.



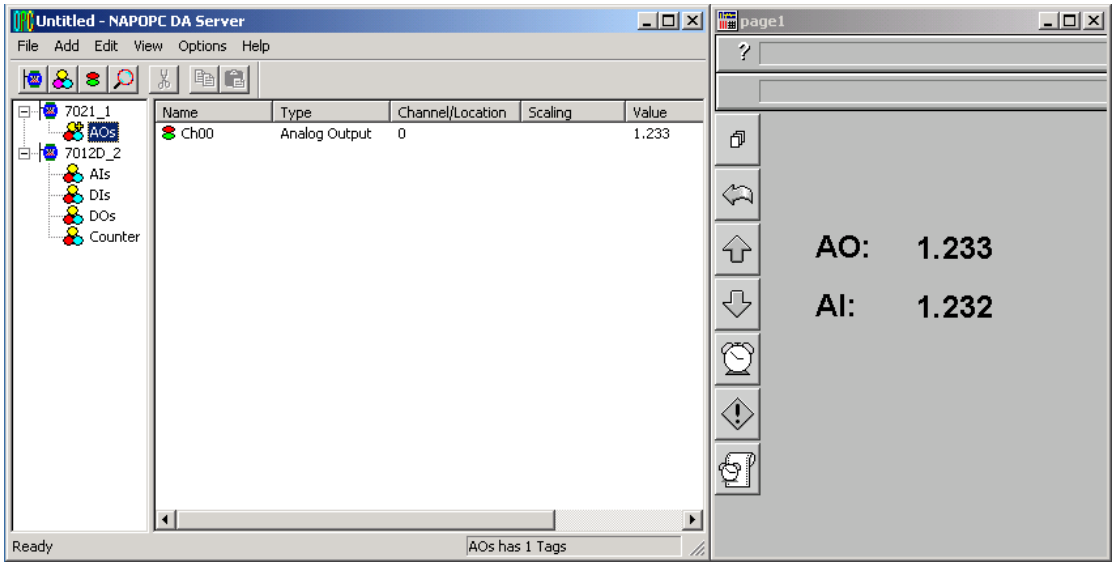
Step36: Key-in the value "1.234" to output a voltage via the module 7021.



Step37: Both NAPOPC_ST Server monitor and CitectSCADA runtime window show the analog input of the modules 7012D is 1.231V.



Step38: Check if the analog output value of the modules 7021 that showed in the NAPOPC_ST Server monitor is 1.234V.



4 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.

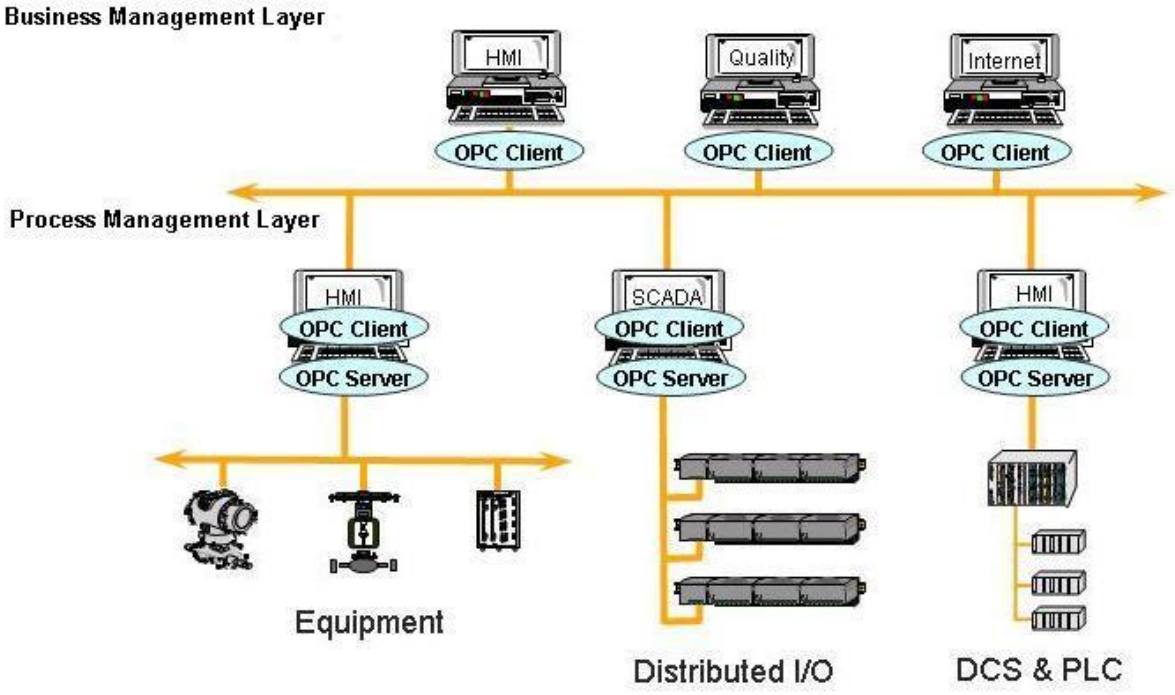


Figure 4-0-1 Local access and Remote access architecture.

4.1 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 application 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 connect 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 four combinations of DCOM settings related to NAPOPC_ST DA Server. You can see XPAC and PC can be client site and server site with each other, but WinPAC only can be server site against 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.

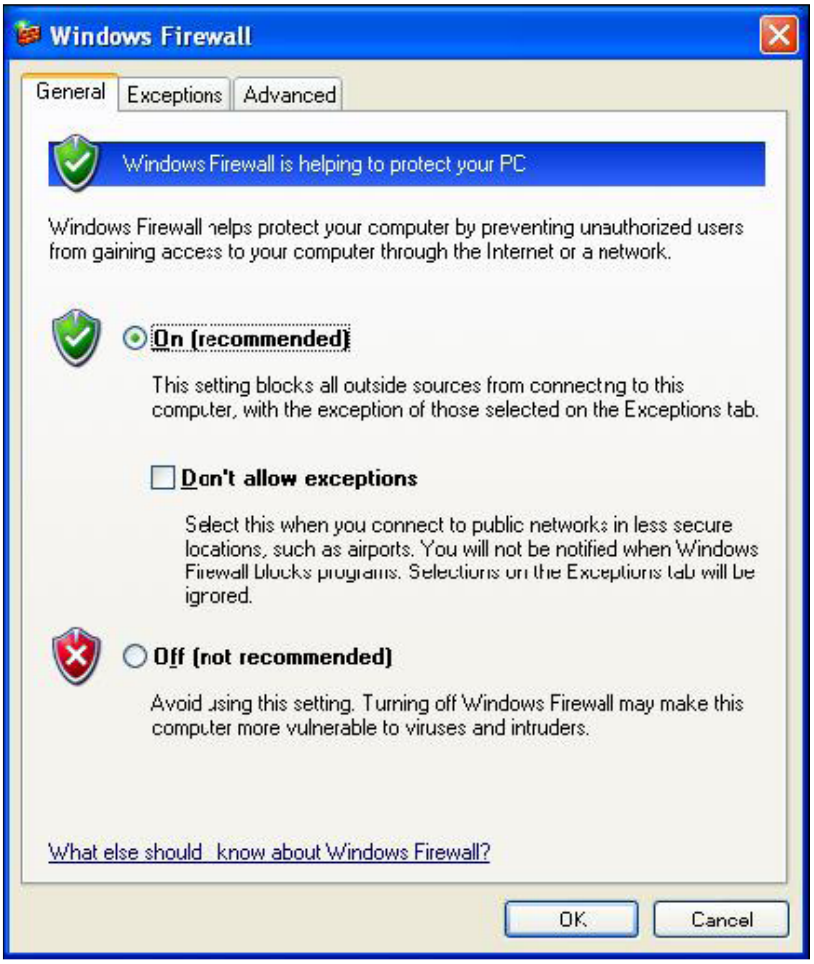
Client Site	Server Site
PC(NAPOPC_ST Server)	PC(NAPOPC_ST Server)
PC(NAPOPC_ST Server)	XPAC(NAPOPC_XPE Server)
XPAC(NAPOPC_XPE Server)	PC(NAPOPC_ST Server)
PC(NAPOPC_ST Server)	WinPAC(NAPOPC_CE5 Server)

4.1.1 Configuring On the Server Site (XPAC)

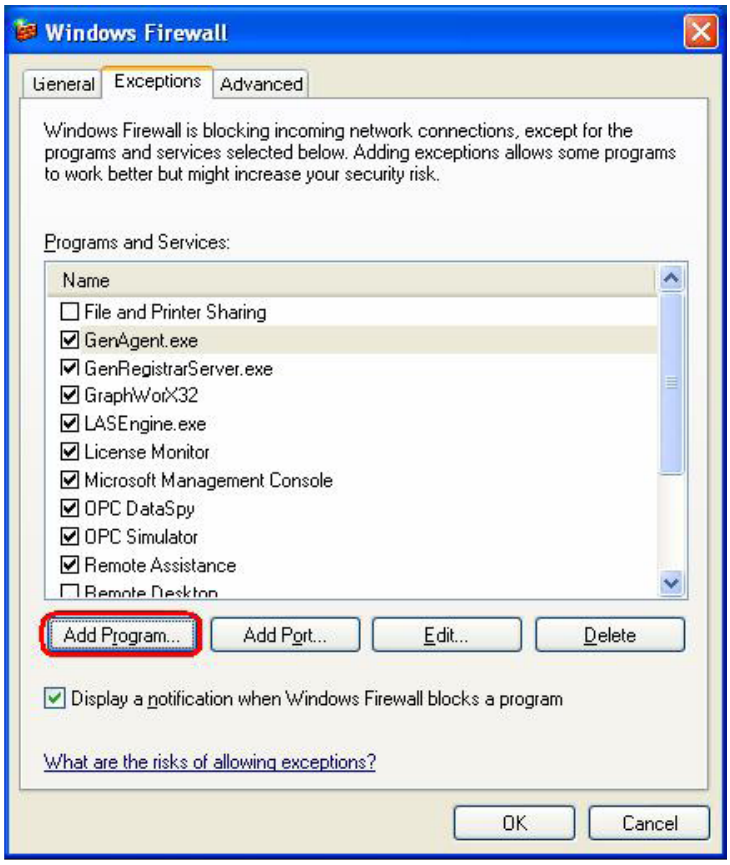
Configuring the Firewall

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.

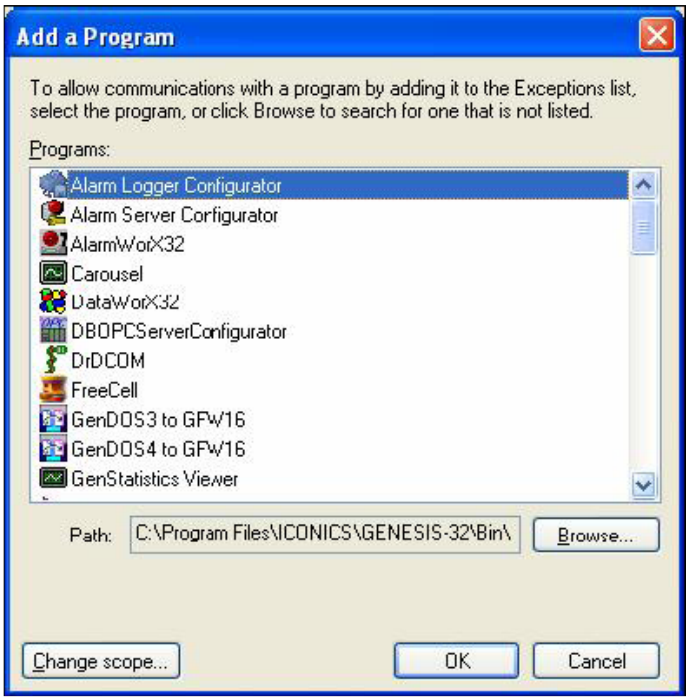


Step 2: Select the .Exceptions tab and add all OPC Clients and Servers to the exception list. Also add Microsoft Management Console (used by the DCOM configuration utility in the next section) and the OPC utility OPCEnum.exe found in the Windows\System32 directory.

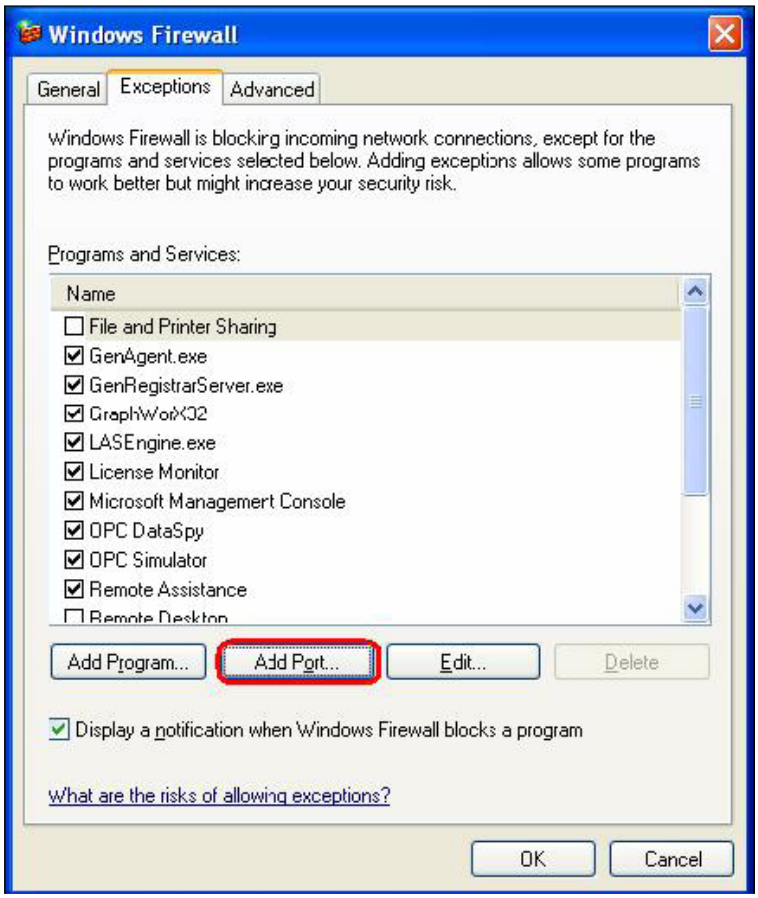


In the Add a Program dialog, there is a listing of most applications on the machine, but note that not all of them show up on this list. Use the "Browse" button to find other executables installed on the computer.

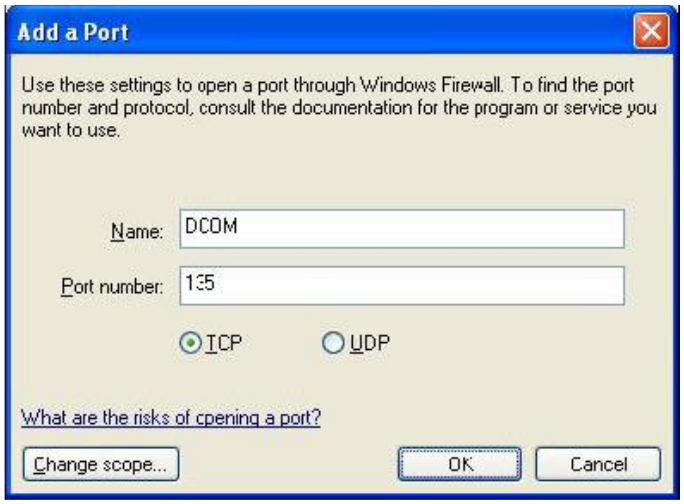
Note: Only EXE files are added to the exceptions list. For in-process OPC Servers and Clients (DLLs and OCXs) you will need to add the EXE applications that call them to the list instead.



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.

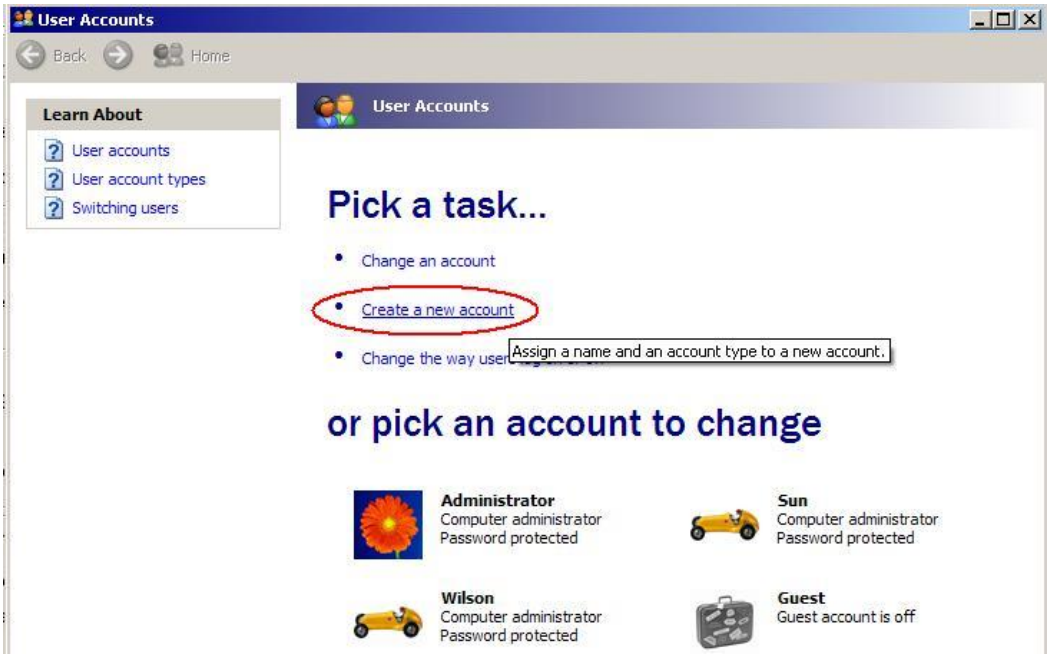


In the Add a Port dialog, fill out the fields as follows:
Name: DCOM
Port number: 135
Choose the TCP radio button



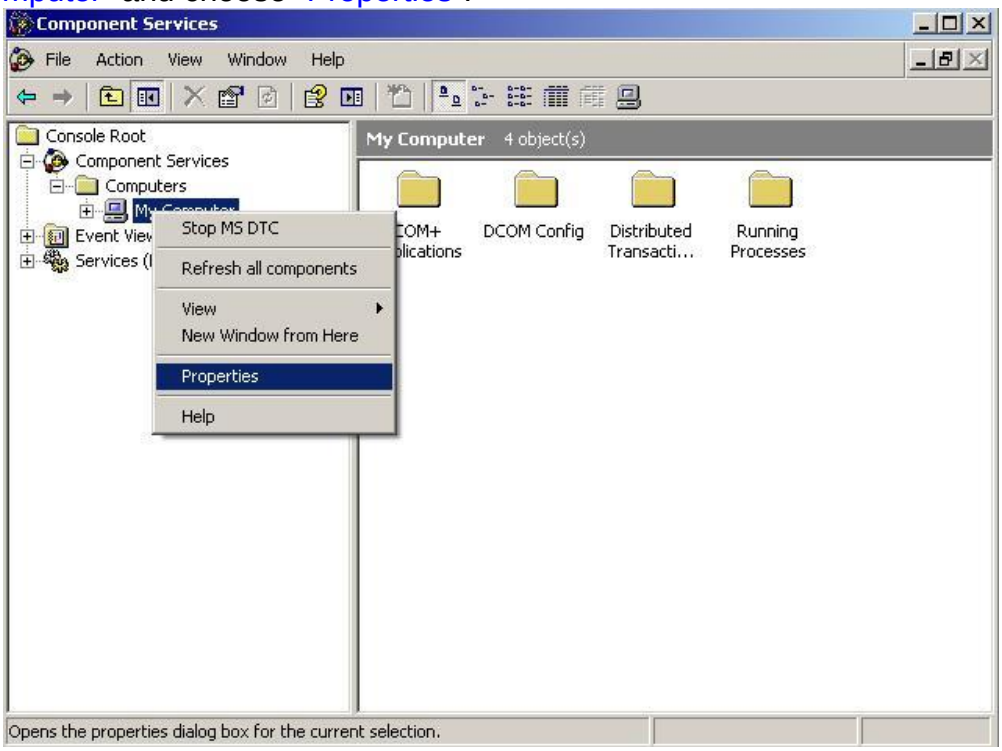
Creating the Account

Step 1: Create a account which must be the same with the account of client site.



Configuring DCOM

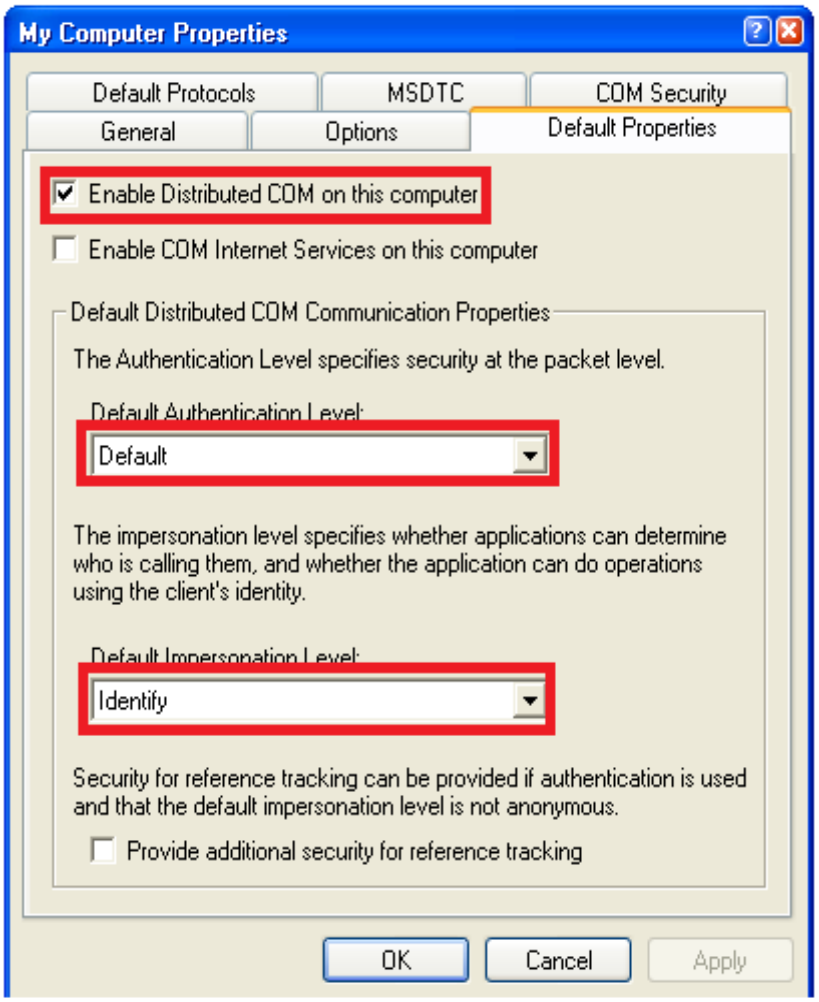
Step 1: Run the dcomcnfg.exe program to launch component services. Right clieck "My Computer" and choose "Properties".



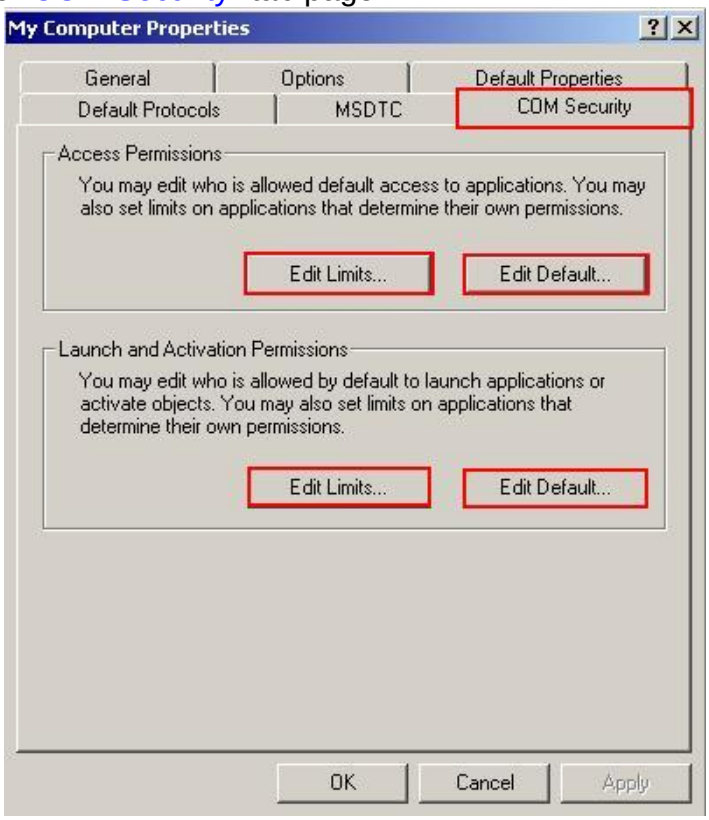
Step 2: Select the "Default Properties" tab page.

Step 3: Use the following settings:

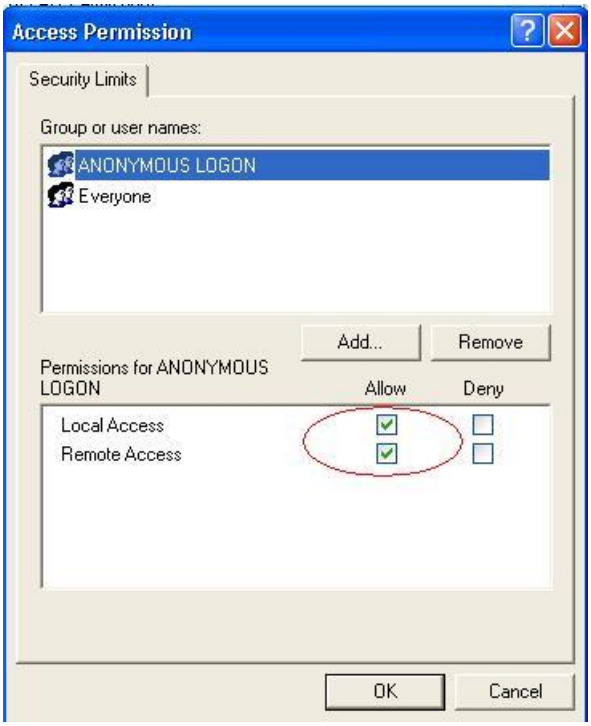
Field Name	Set to
Enable Distributed COM on this computer	Checked
Default Authentication Level:	Default
Default Impersonation Level:	Identify

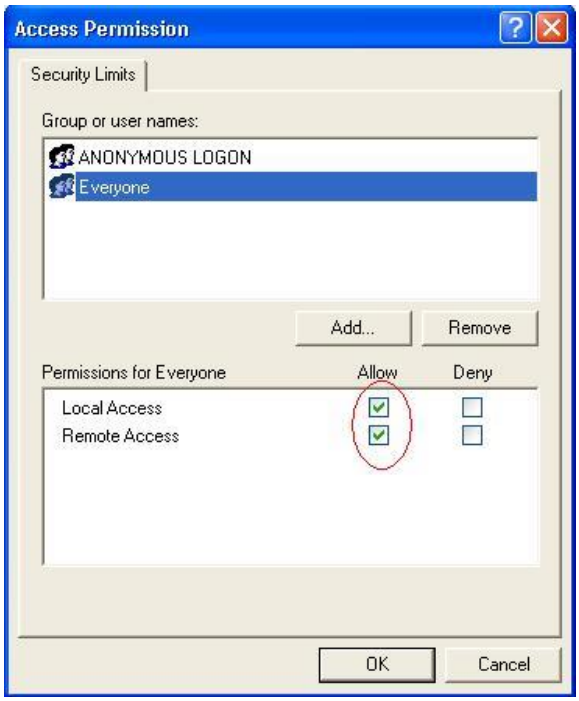


Step 4: Select the "COM Security" tab page.

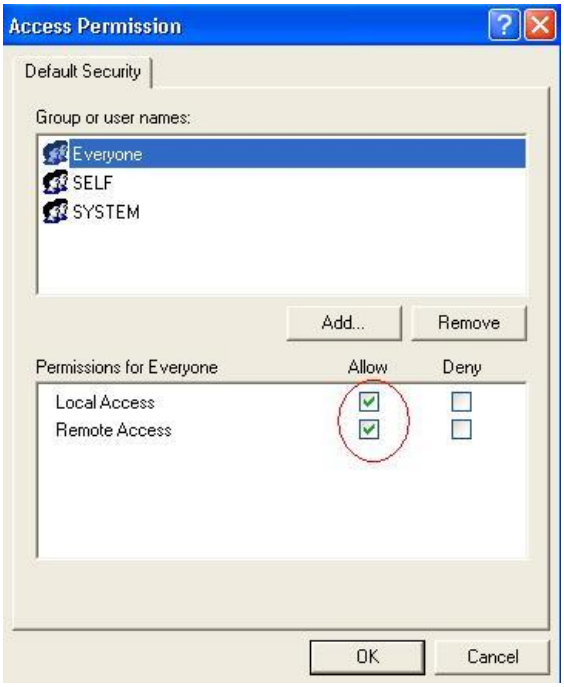


Step 5: Click on the "Edit Limits…" of "Access Permissions" button to set.

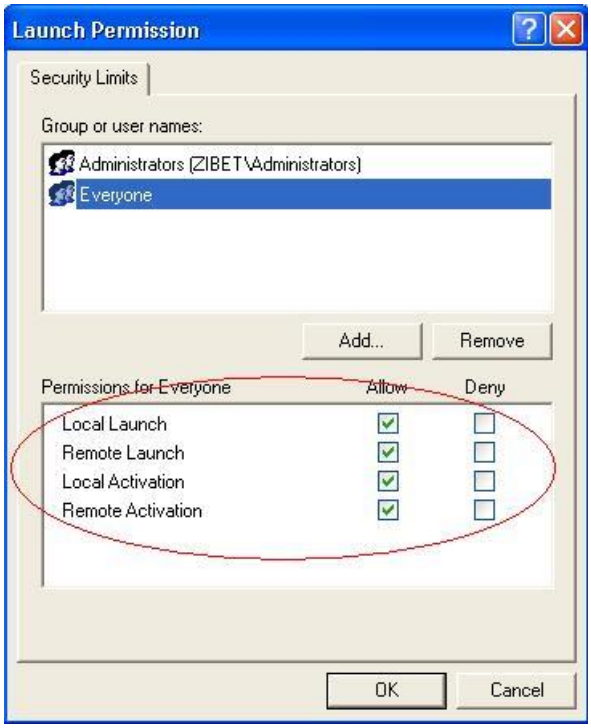




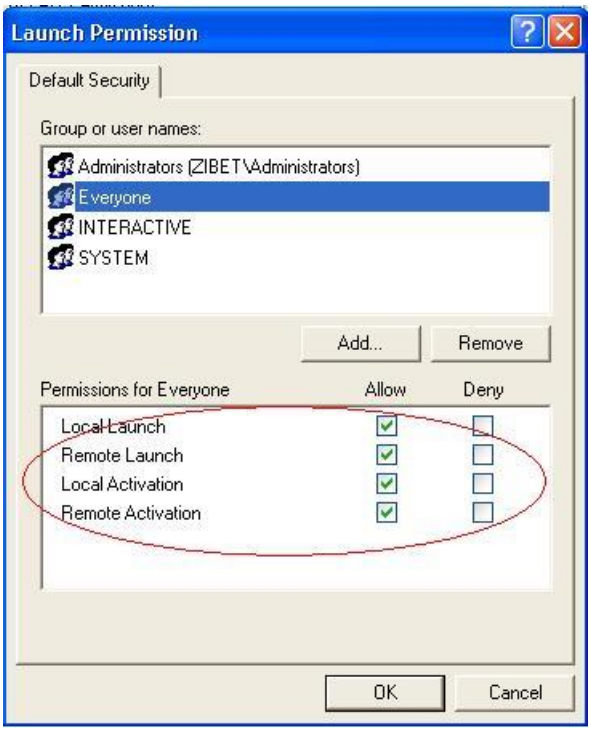
Step 6: Click on the "Edit Default…" of "Access Permissions" button to set.



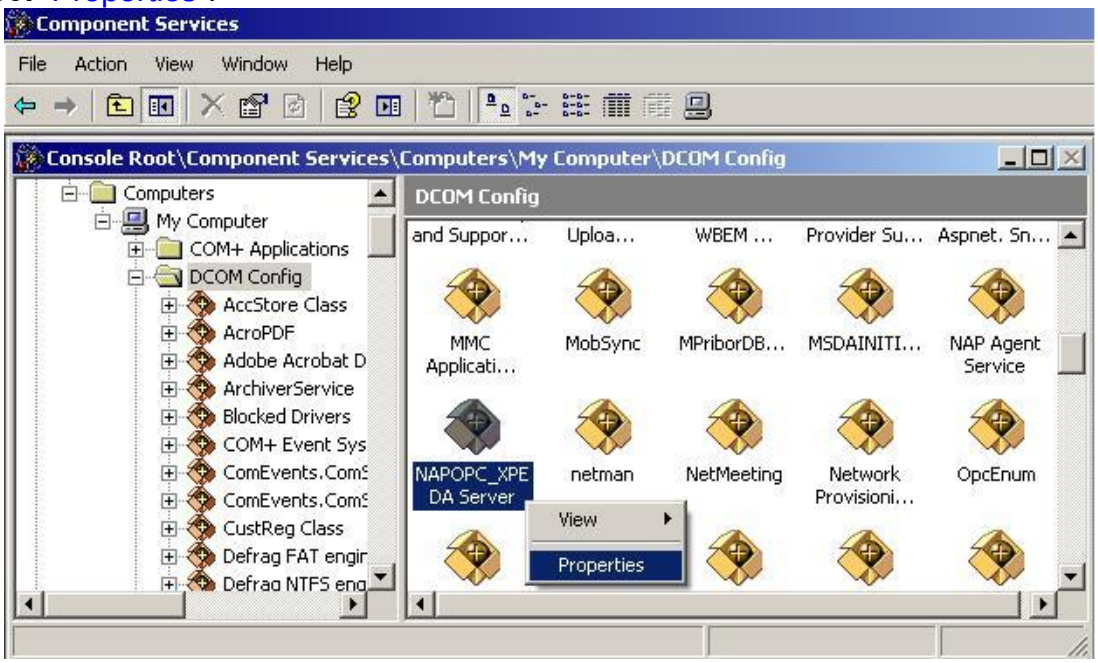
Step 7: Click on the "Edit Limits…" of "Launch and Activation Permissions" button to set.



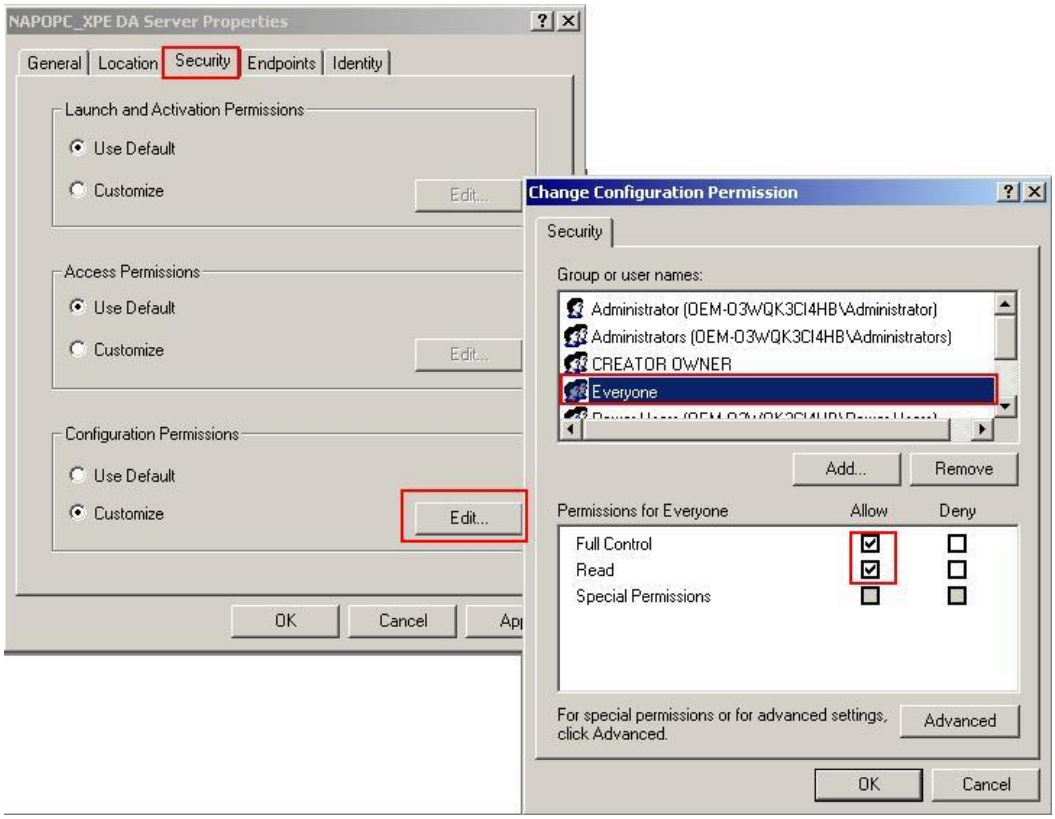
Step 8: Click on the "Edit Limits…" of "Launch and Activation Permissions" button to set.



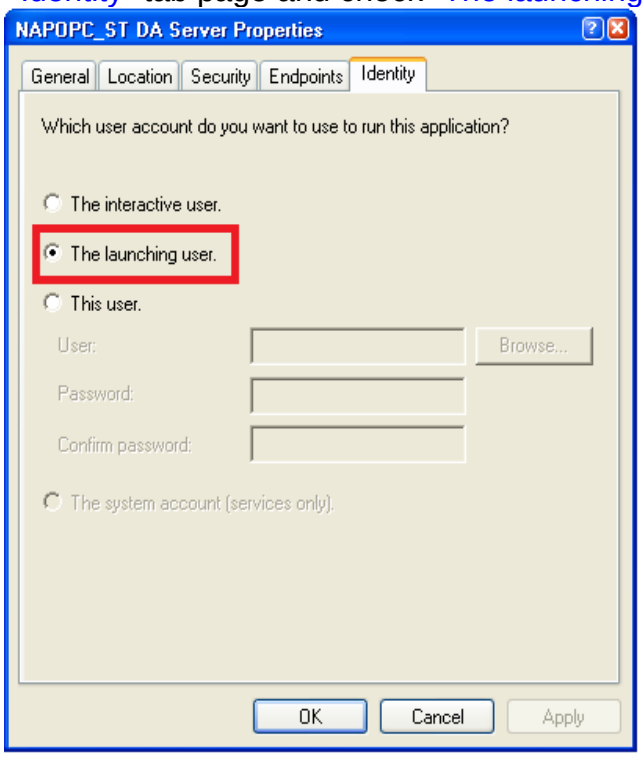
Step 9: Right click on the "NAPOPC_XPE DA Server" of "DCOM Config" button and select "Properties".



Step 10: Select the "Security" tab page and click "Edit…" of "Configuration Permissions". To make sure there is "Everyone" in "Group or user names" and allow "Full Control" and "Read"



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



Step 12: Restart XPAC

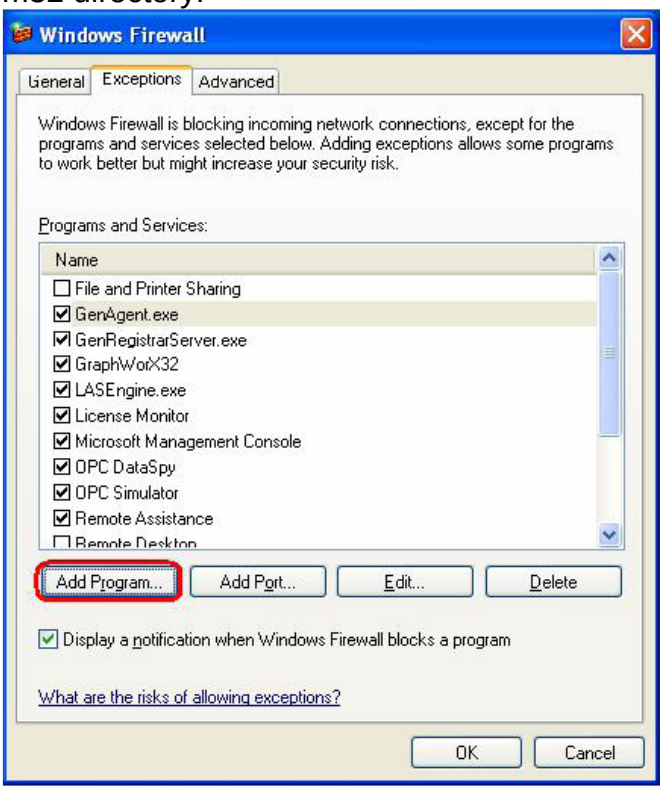
4.1.2 Configuring On the Server Site (PC)

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.

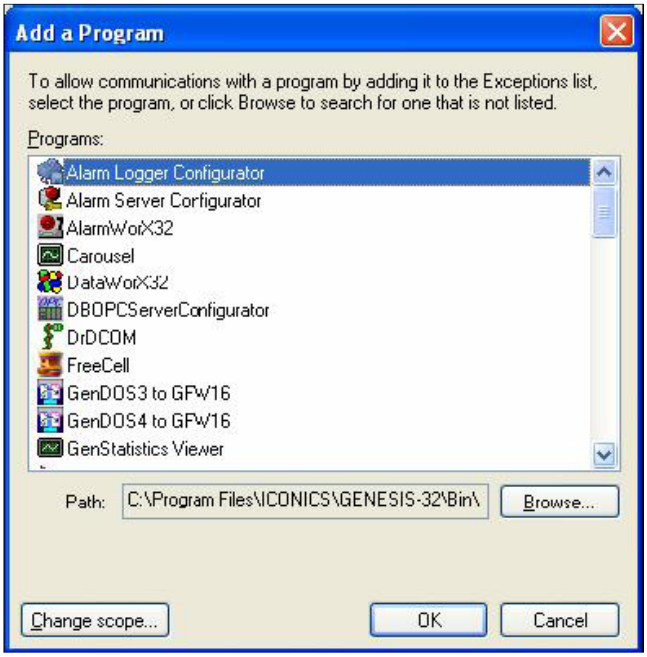
Step 2: Select the .Exceptions tab and add all OPC Clients and Servers to the exception list. Also add Microsoft Management Console (used by the DCOM configuration utility in the next section) and the OPC utility OPCEnum.exe found in the Windows\System32 directory.



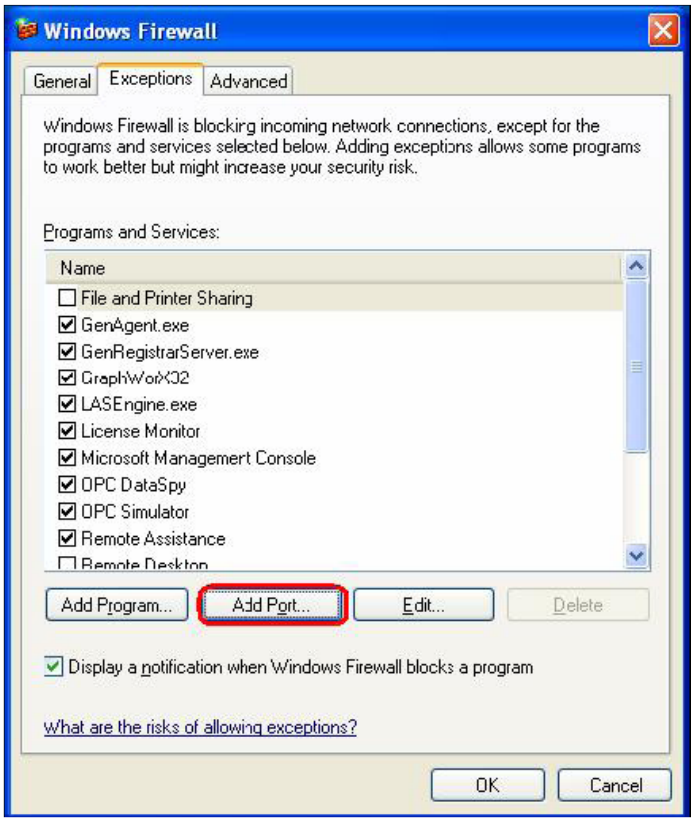
In the Add a Program dialog, there is a listing of most applications on the machine,

but note that not all of them show up on this list. Use the "Browse" button to find other executables installed on the computer.

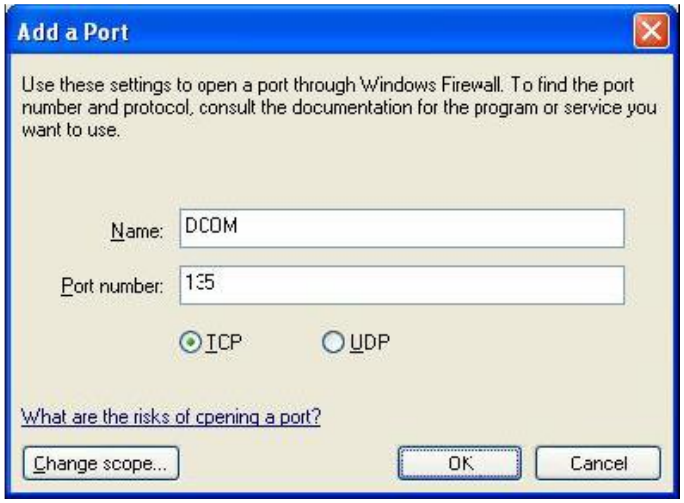
Note: Only EXE files are added to the exceptions list. For in-process OPC Servers and Clients (DLLs and OCXs) you will need to add the EXE applications that call them to the list instead.



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.

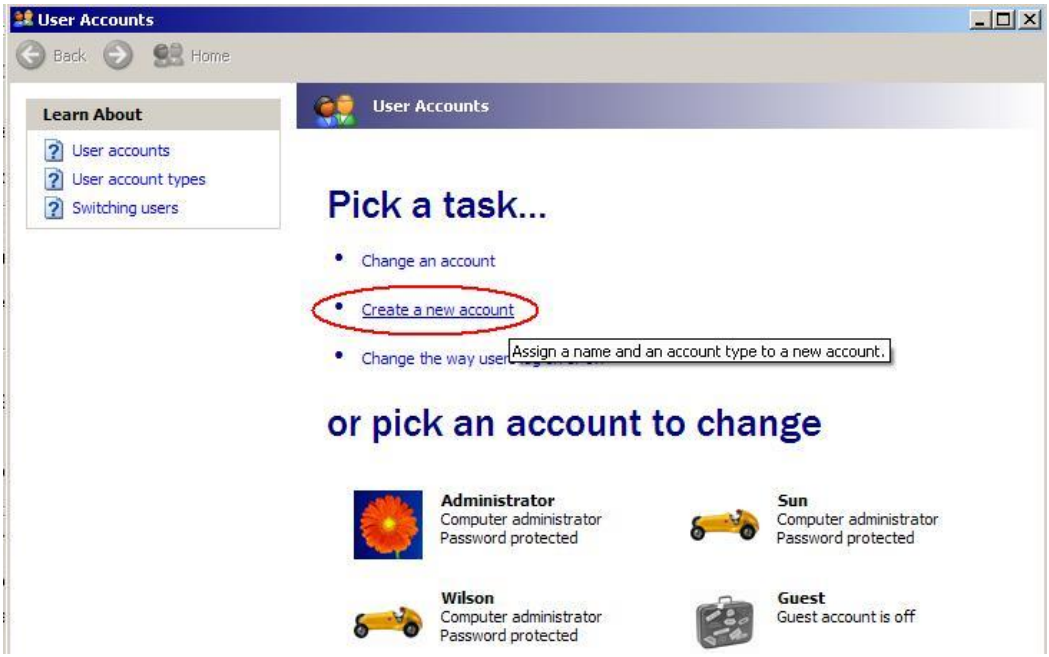


In the Add a Port dialog, fill out the fields as follows:
Name: DCOM
Port number: 135
Choose the TCP radio button



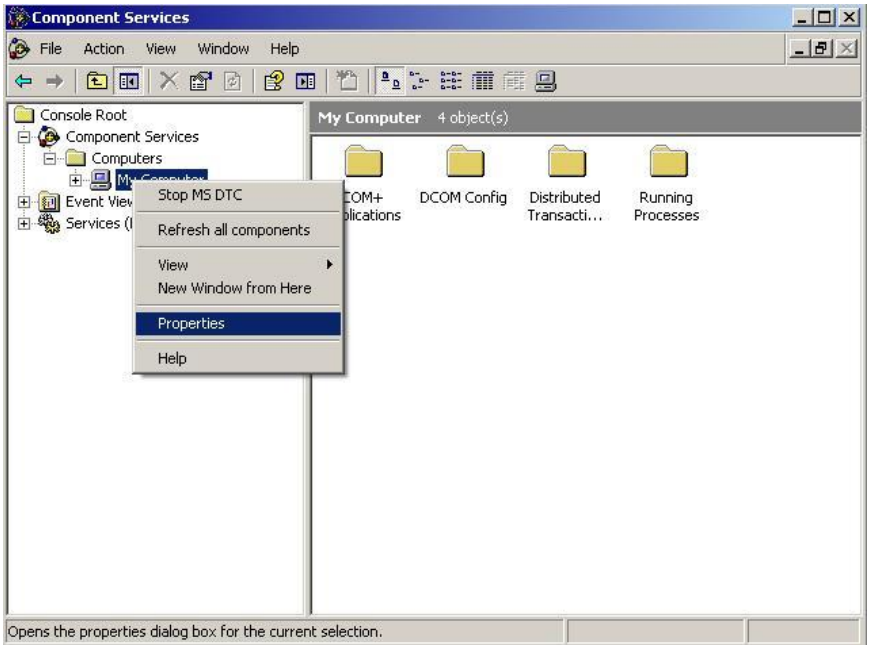
Creating the Account

Step 1: Create a account which must be the same with the account of client site.



Configuring DCOM

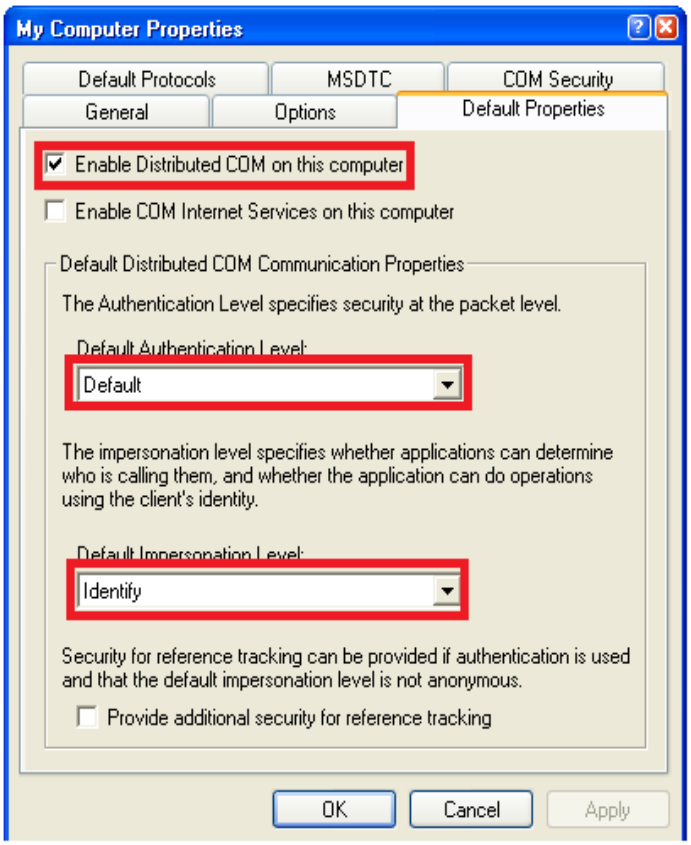
Step 1: Run the dcomcnfg.exe program to launch component services. Right clieck "My Computer" and choose "Properties".



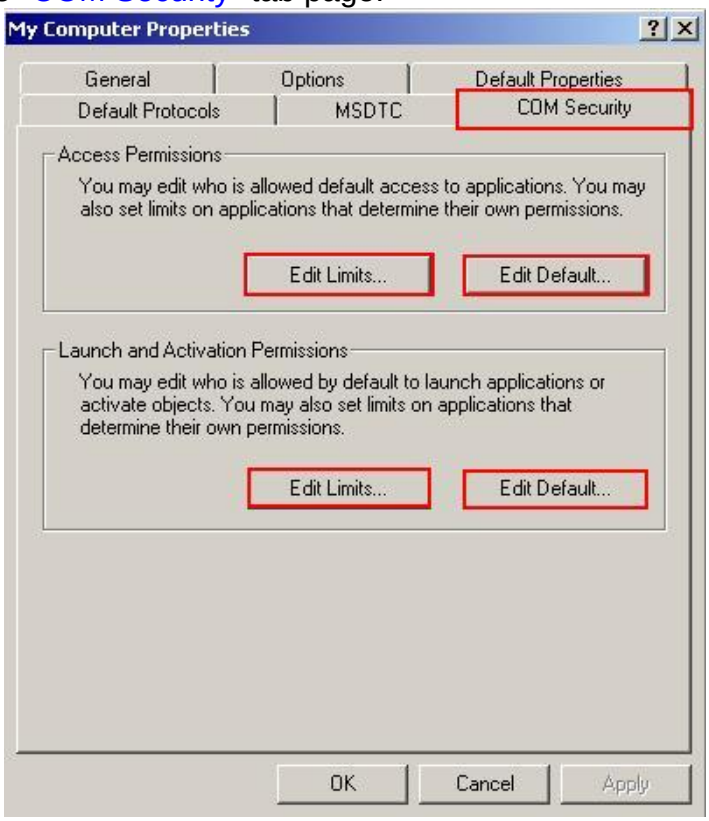
Step 2: Select the "Default Properties" tab page.

Step 3: Use the following settings:

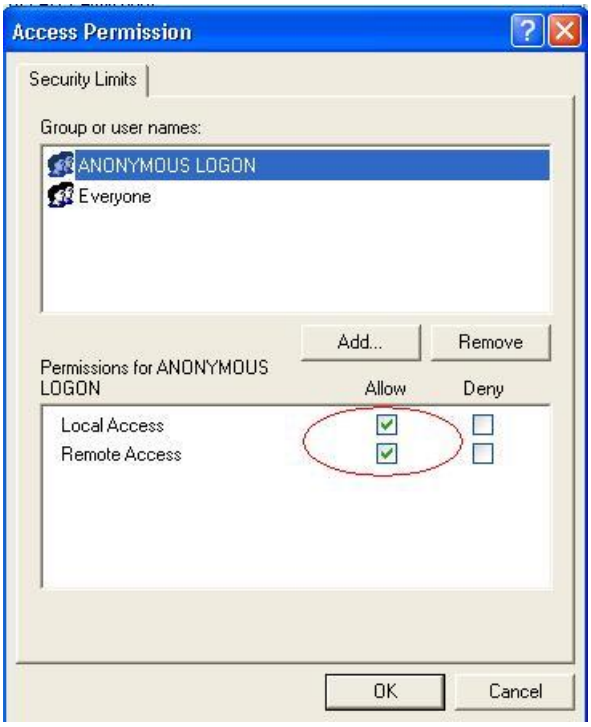
Field Name	Set to
Enable Distributed COM on this computer	Checked
Default Authentication Level:	Default
Default Impersonation Level:	Identify

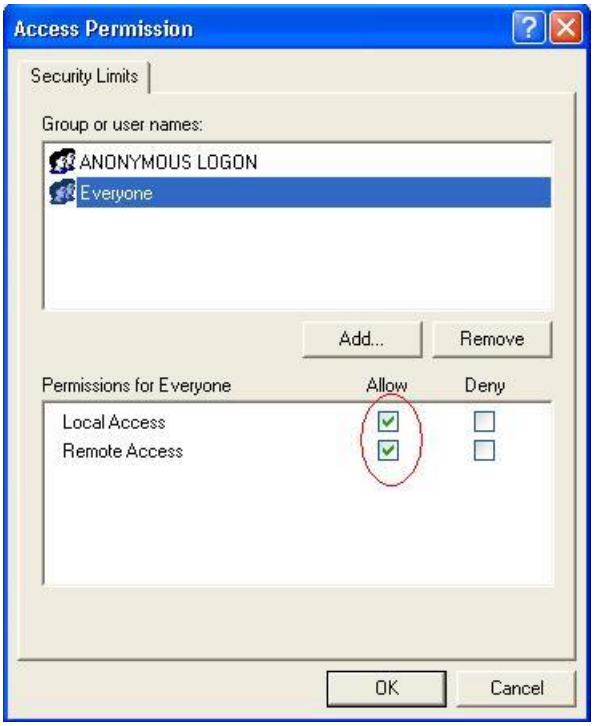


Step 4: Select the "COM Security" tab page.

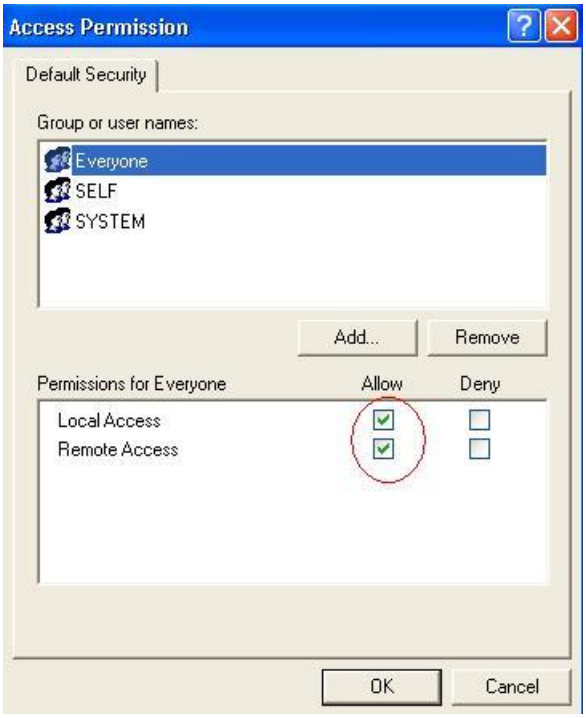


Step 5: Click on the "Edit Limits…" of "Access Permissions" button to set.

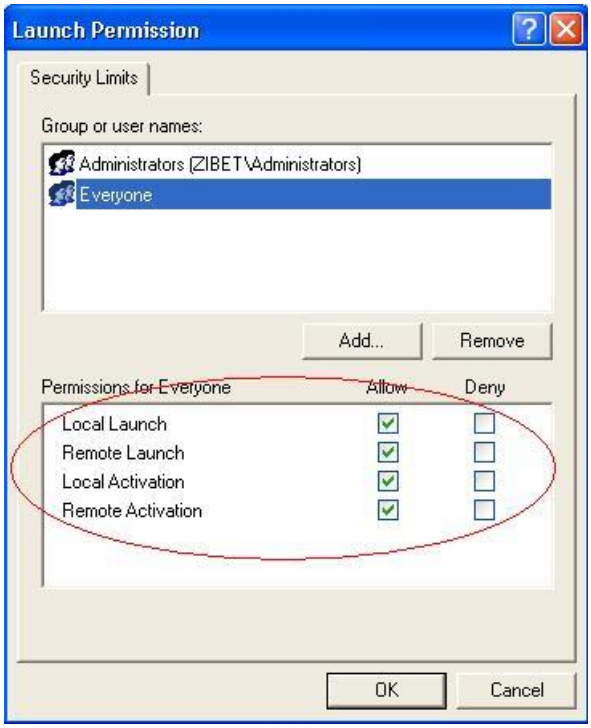




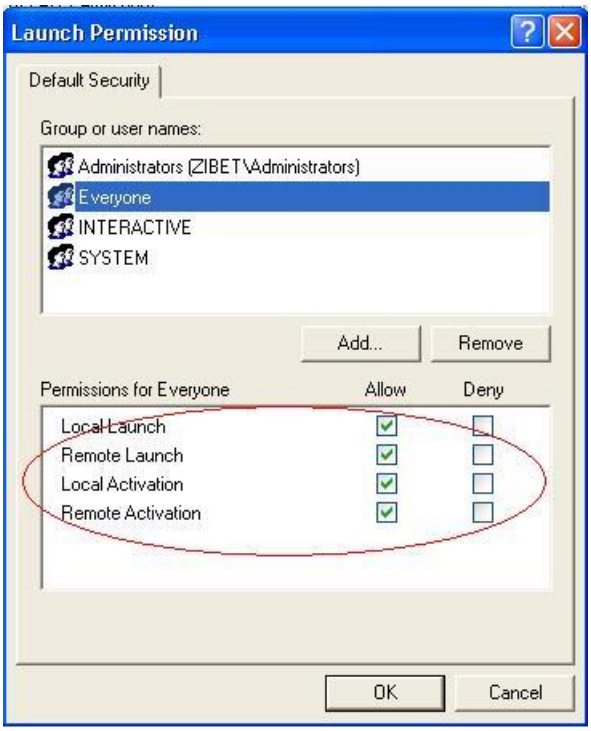
Step 6: Click on the "Edit Default…" of "Access Permissions" button to set.



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

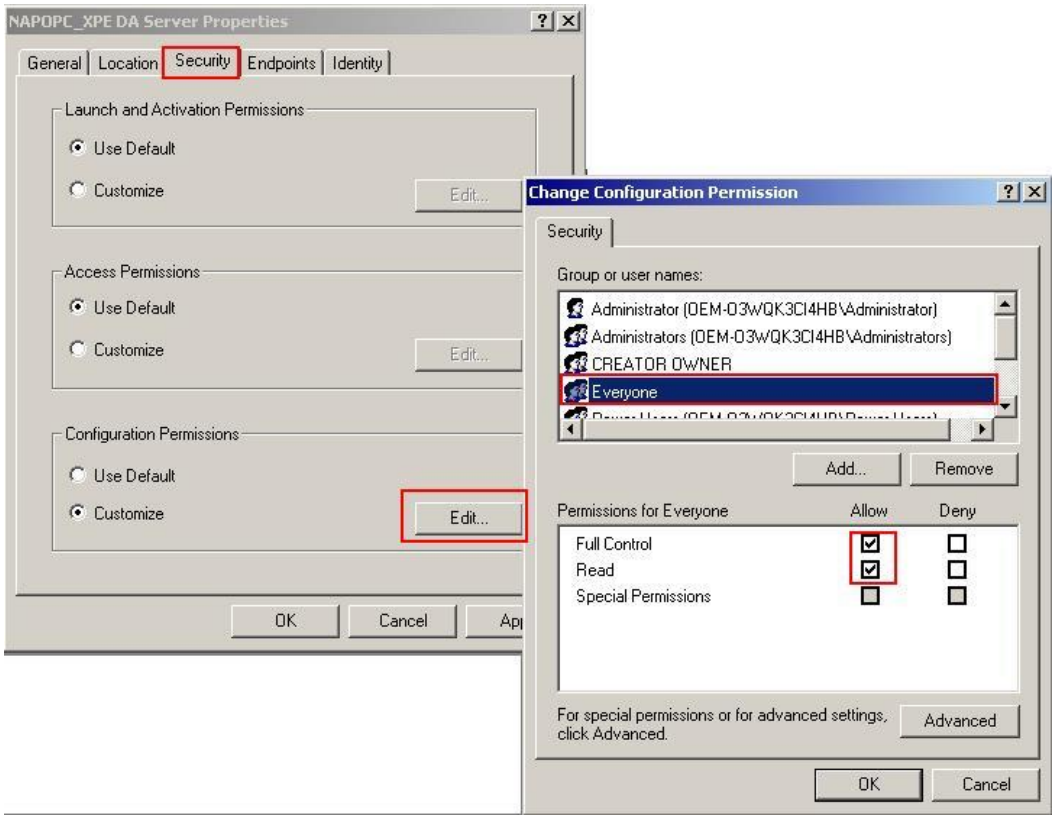


Step 8: Click on the "Edit Limits…" of "Launch and Activation Permissions" button to set.

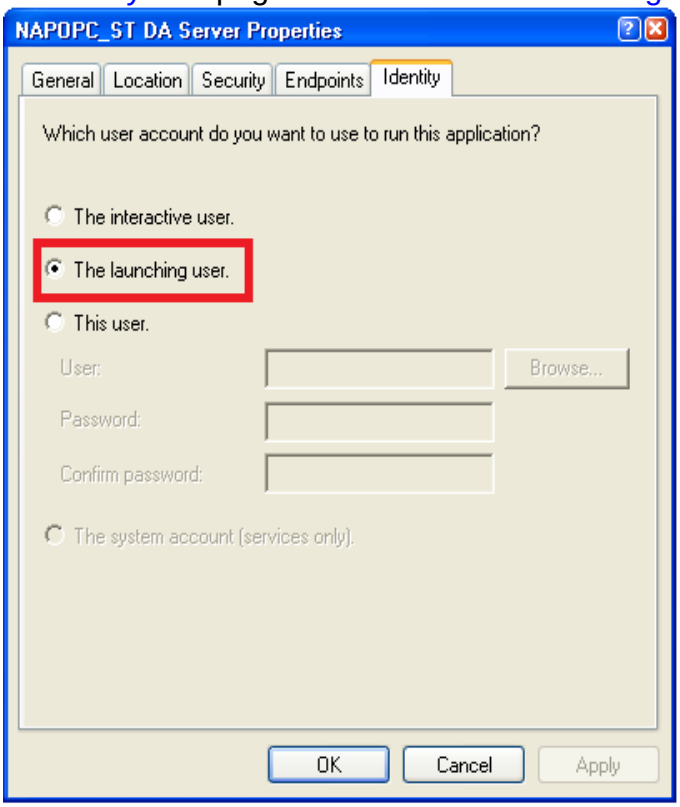


Step 9: Right click on the "NAPOPC_ST DA Server" of "DCOM Config" button and select "Properties".

Step 10: Select the "Security" tab page and click "Edit…" of "Configuration Permissions". To make sure there is "Everyone" in "Group or user names" and allow "Full Control" and "Read"



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



Step 12: Restart PC

4.1.3 Configuring On the Server Site (WinPAC)

System Requirement

OS version:

WinPAC OS 1.3.04 or later

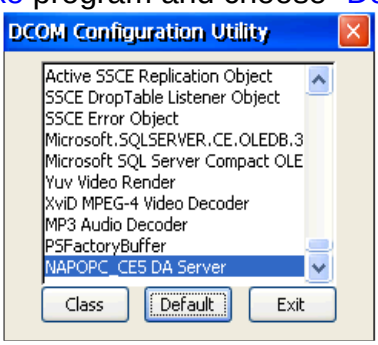
Program:

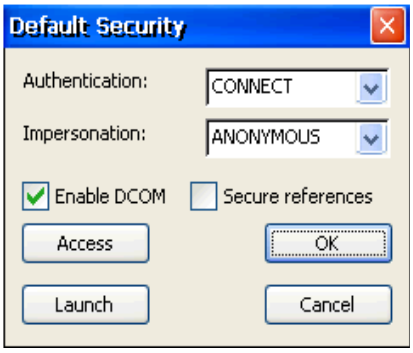
NAPOPC_CE5
DCOMCnfg.exe
WinPAC Utility 2.0.2.1 or later

Configuring DCOM

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

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

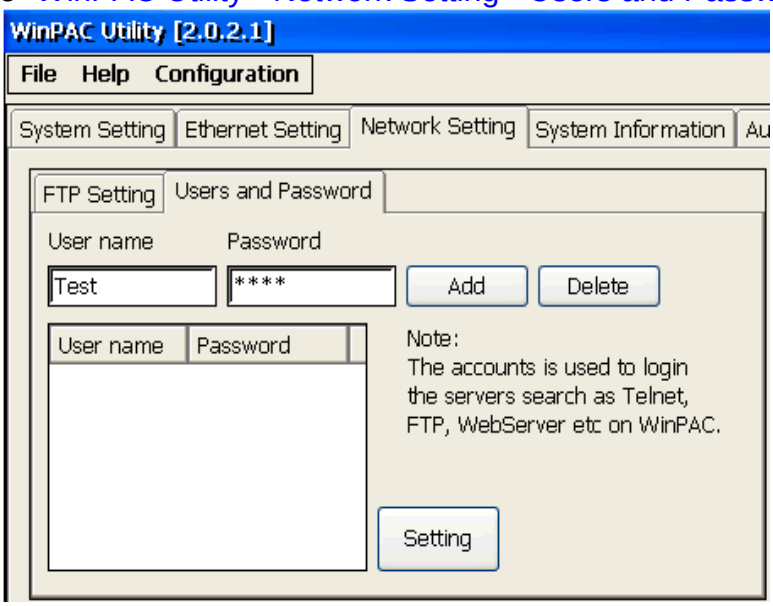




Step 3: Select the "Access" button to add an account which is current connection account from client site.

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"



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.

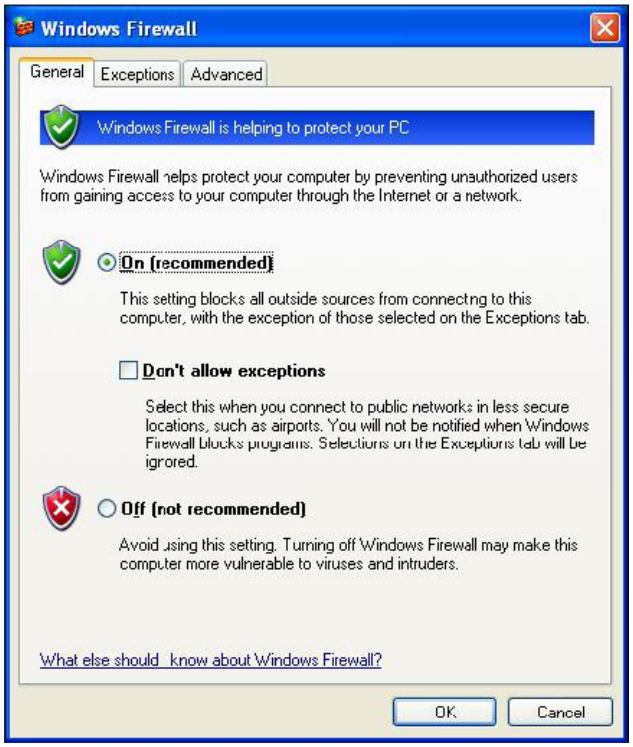
Step 7: Run WinPAC Utility to save and reboot.

4.1.4 Configuring On the Client Site (PC)

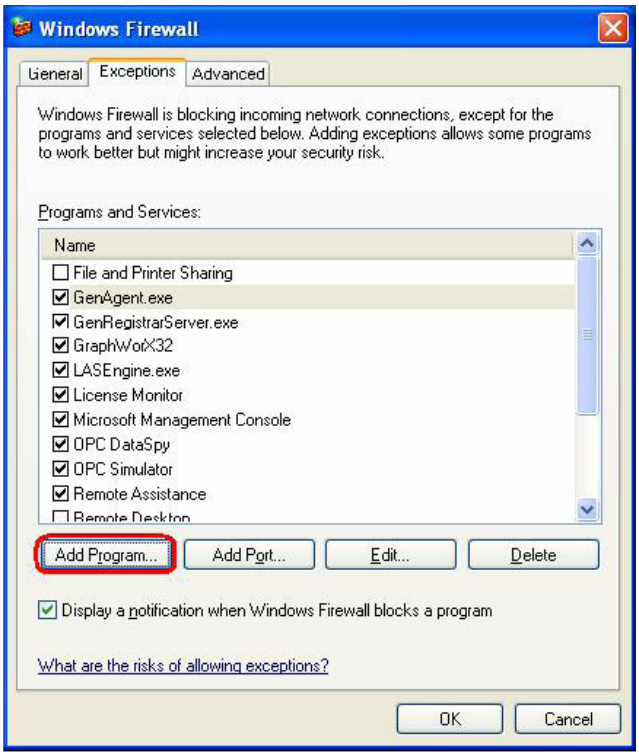
Configuring the Firewall

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.

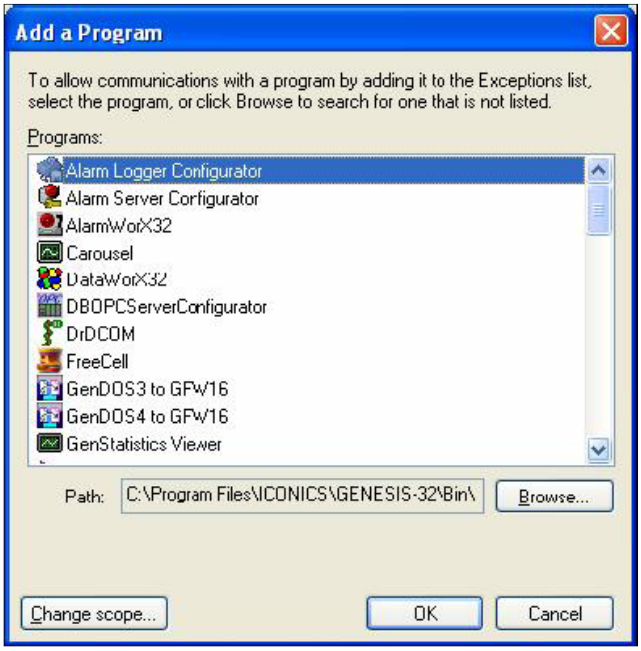


Step 2: Select the .Exceptions tab and add all OPC Clients and Servers to the exception list. Also add Microsoft Management Console (used by the DCOM configuration utility in the next section) and the OPC utility OPCEnum.exe found in the Windows\System32 directory.

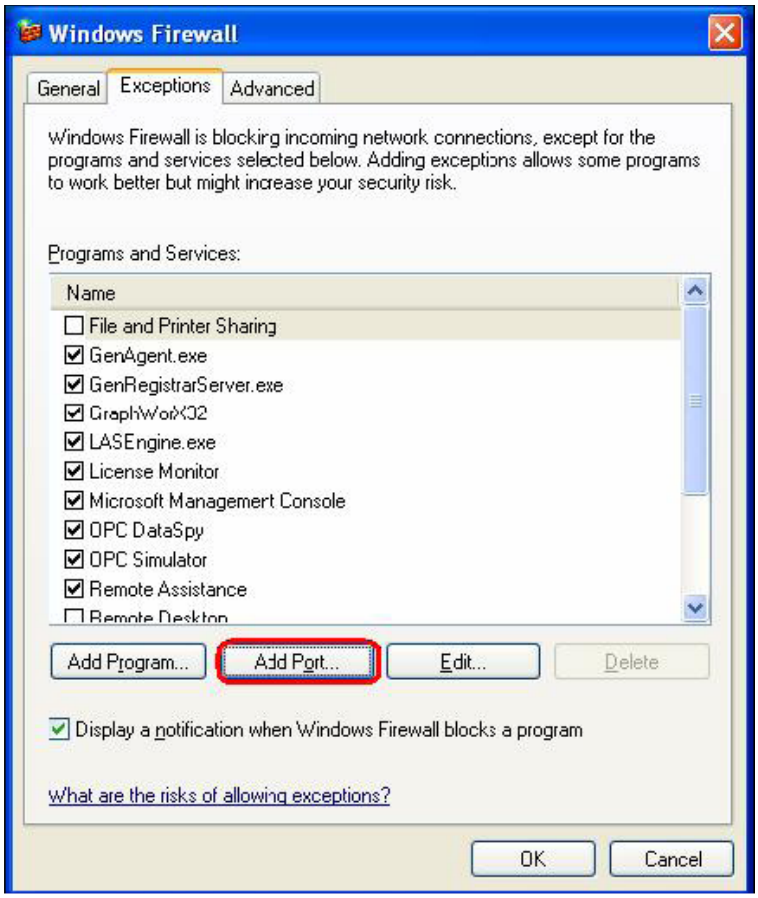


In the Add a Program dialog, there is a listing of most applications on the machine, but note that not all of them show up on this list. Use the "Browse" button to find other executables installed on the computer.

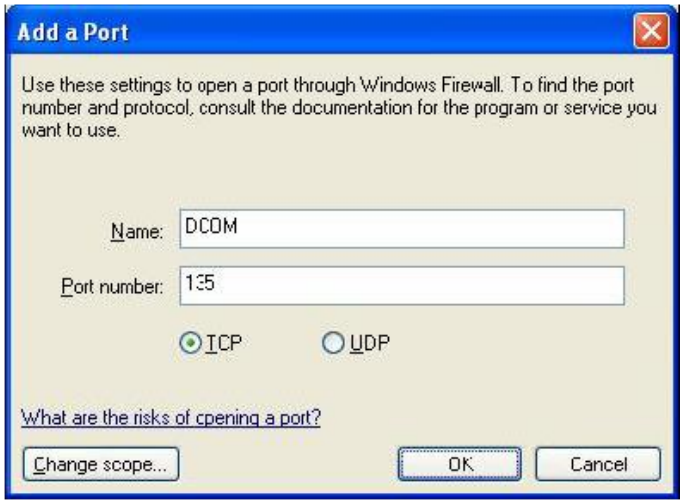
Note: Only EXE files are added to the exceptions list. For in-process OPC Servers and Clients (DLLs and OCXs) you will need to add the EXE applications that call them to the list instead.



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.

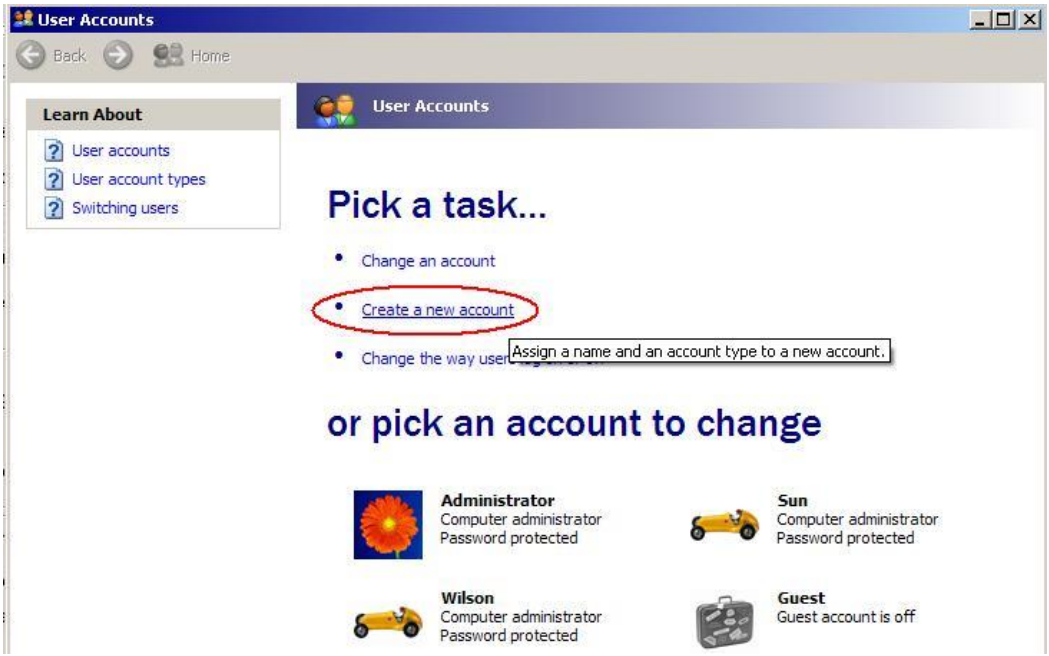


In the Add a Port dialog, fill out the fields as follows:
Name: DCOM
Port number: 135
Choose the TCP radio button



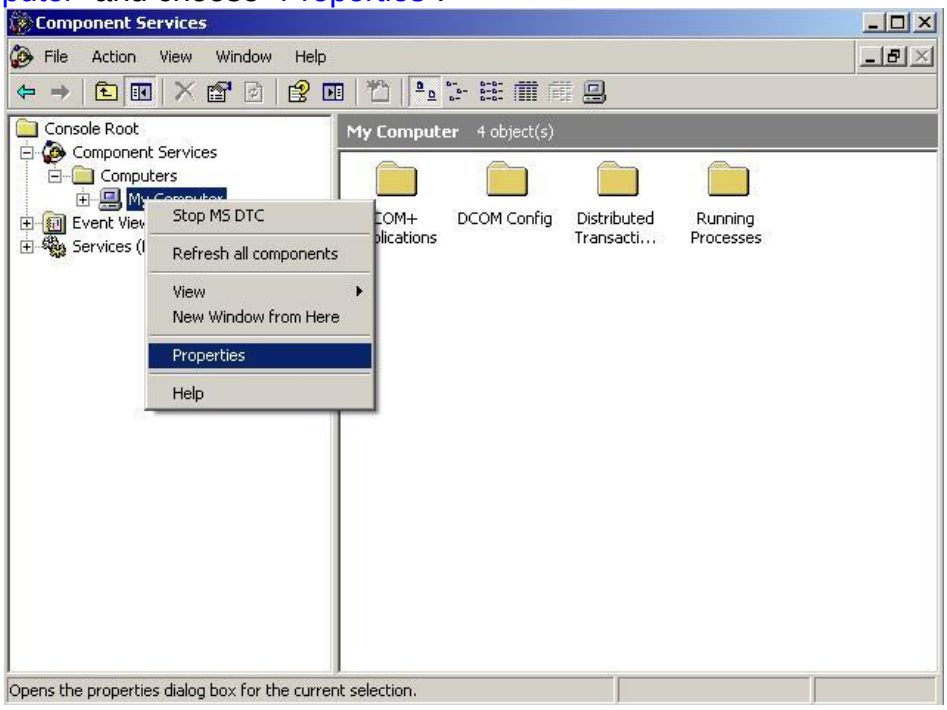
Creating the Account

Step 1: Create a account which must be the same with the account of server site.



Configuring DCOM

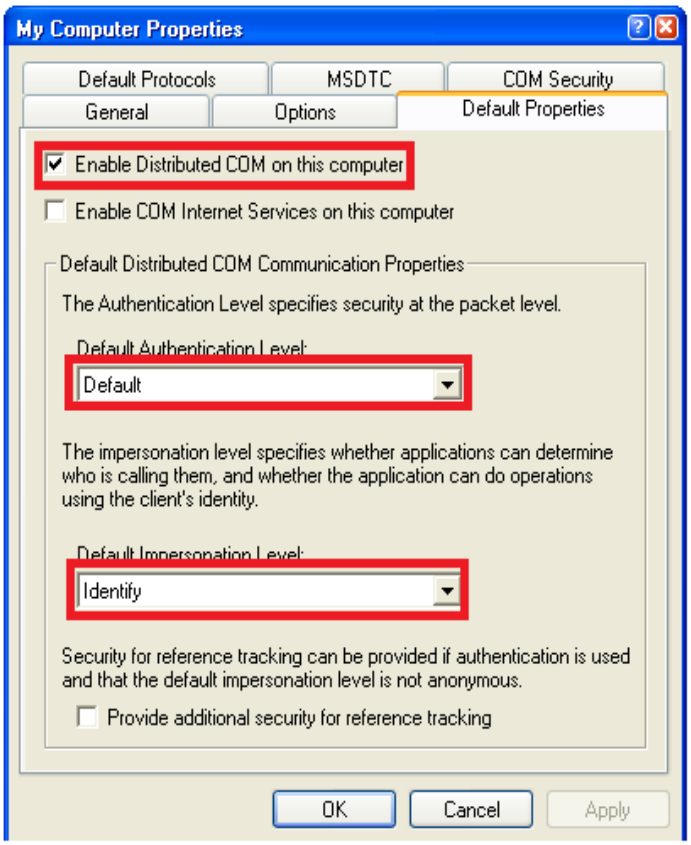
Step 1: Run the dcomcnfg.exe program to launch component services. Right clieck "My Computer" and choose "Properties".



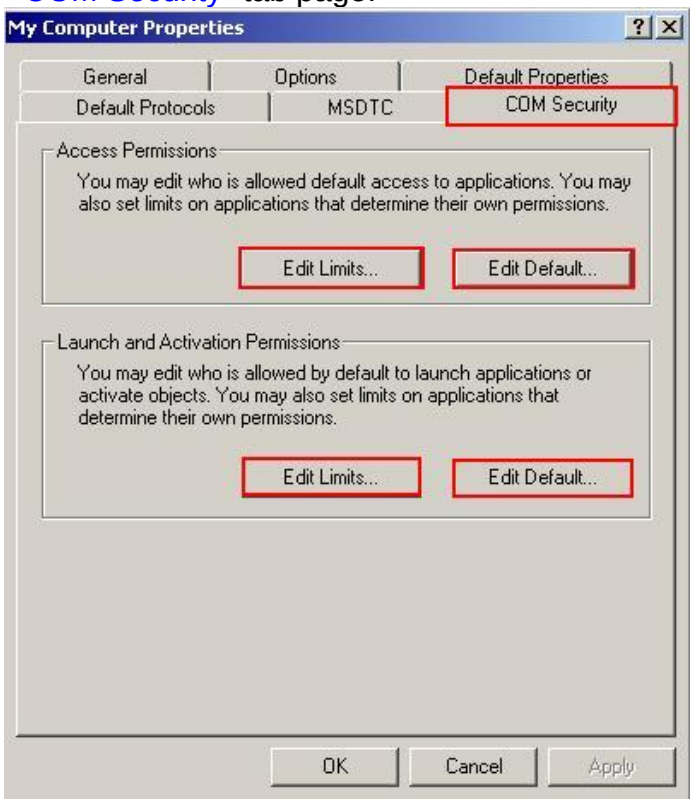
Step 2: Select the "Default Properties" tab page.

Step 3: Use the following settings:

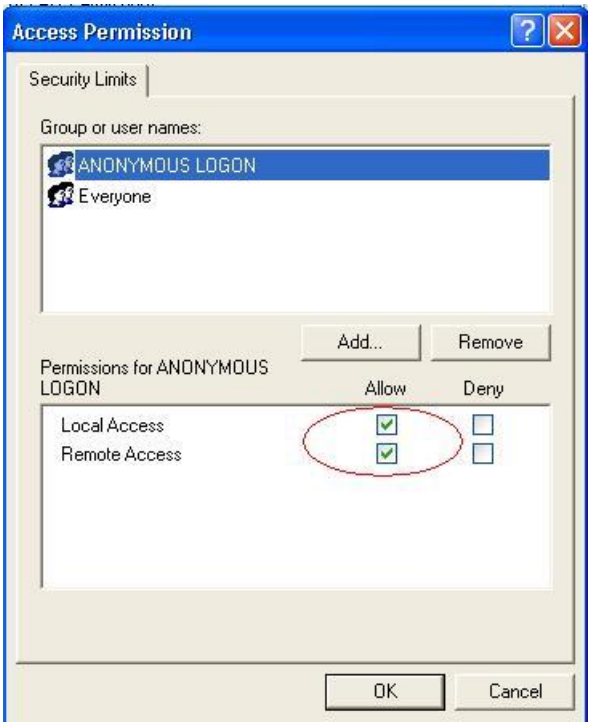
Field Name	Set to
Enable Distributed COM on this computer	Checked
Default Authentication Level:	Default
Default Impersonation Level:	Identify

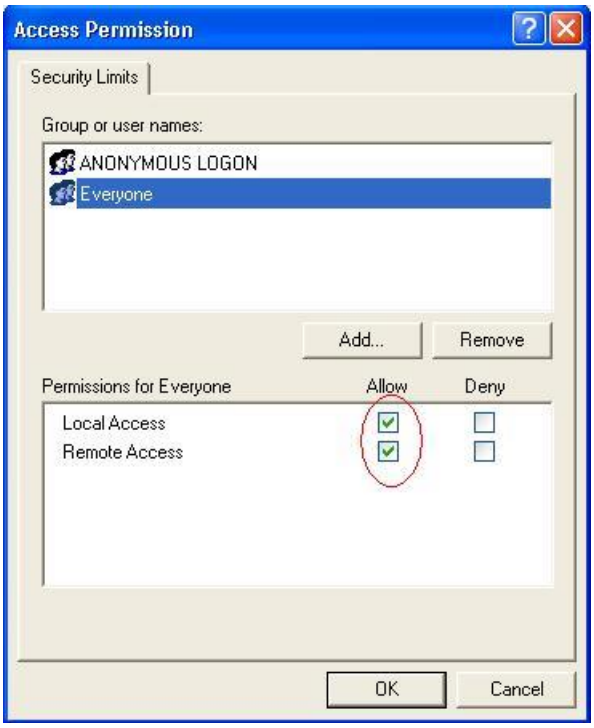


Step 4: Select the "COM Security" tab page.



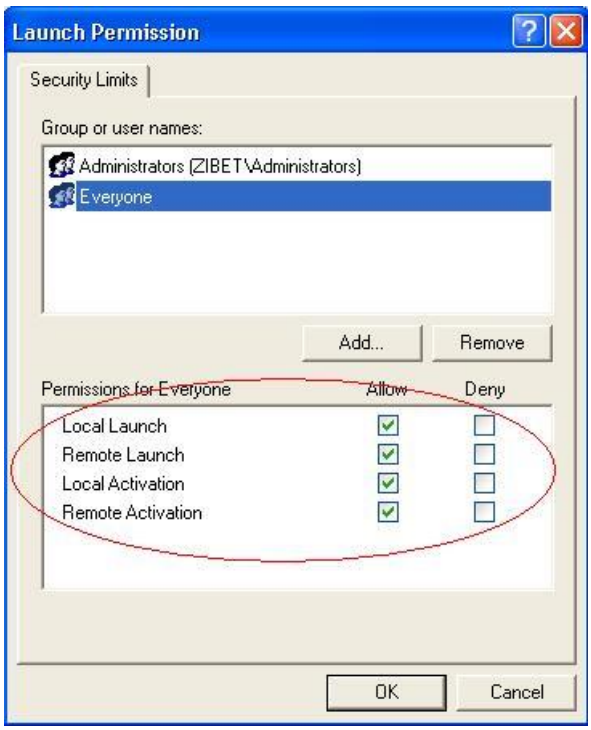
Step 5: Click on the "Edit Limits…" of "Access Permissions" button to set.



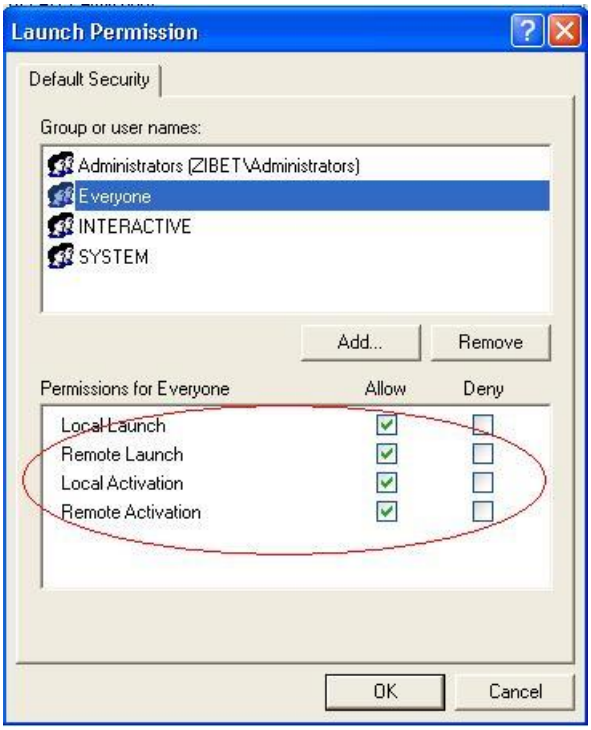


Step 6: Click on the "Edit Default…" of "Access Permissions" button to set.

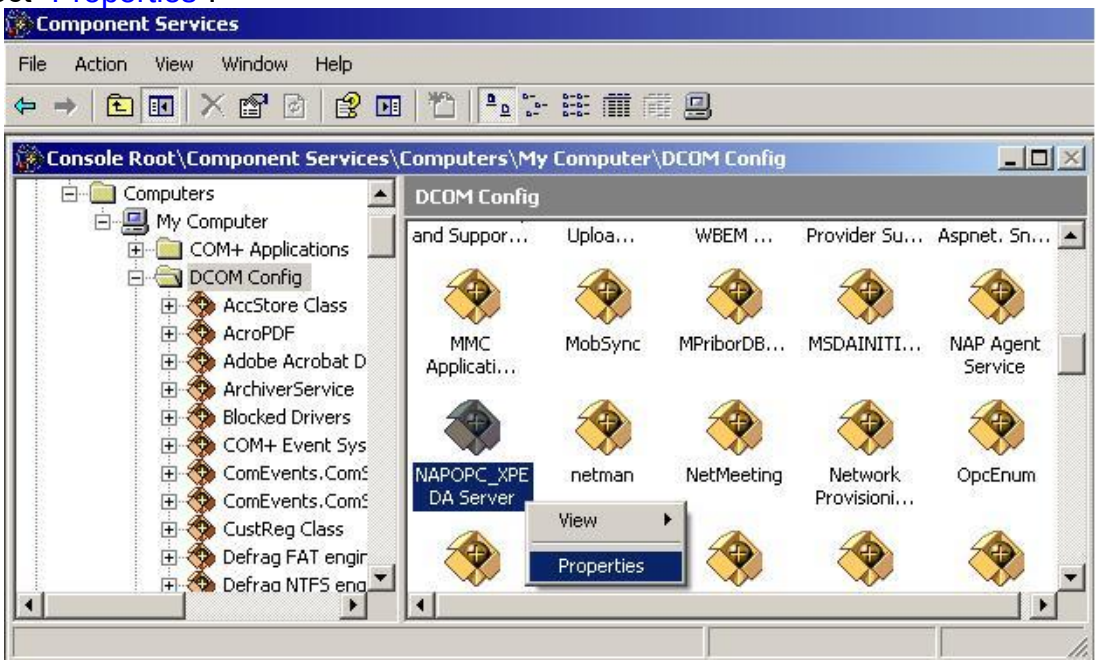
Step 7: Click on the "Edit Limits…" of "Launch and Activation Permissions" button to set.



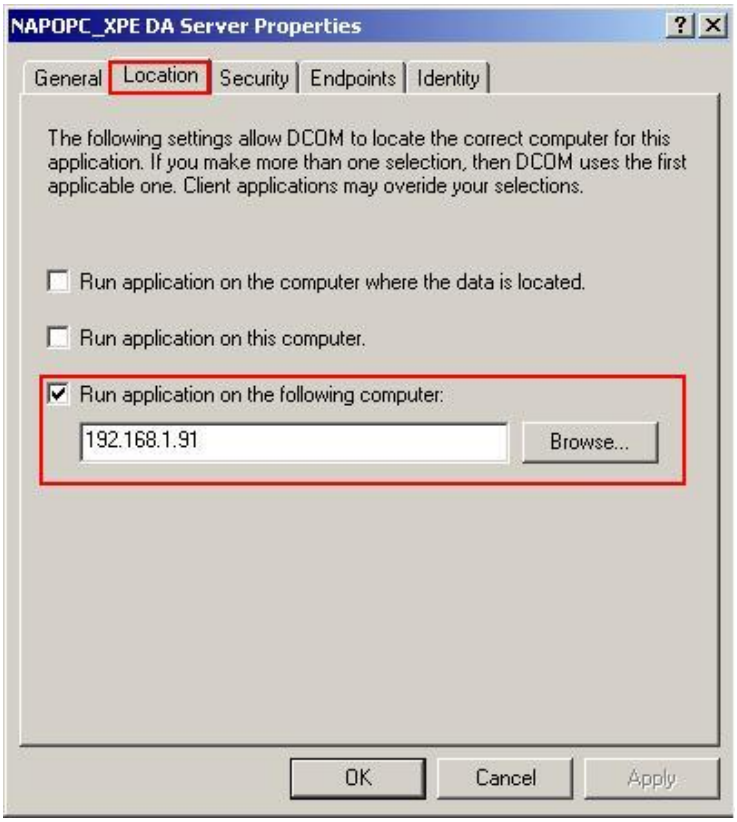
Step 8: Click on the "Edit Limits…" of "Launch and Activation Permissions" button to set.



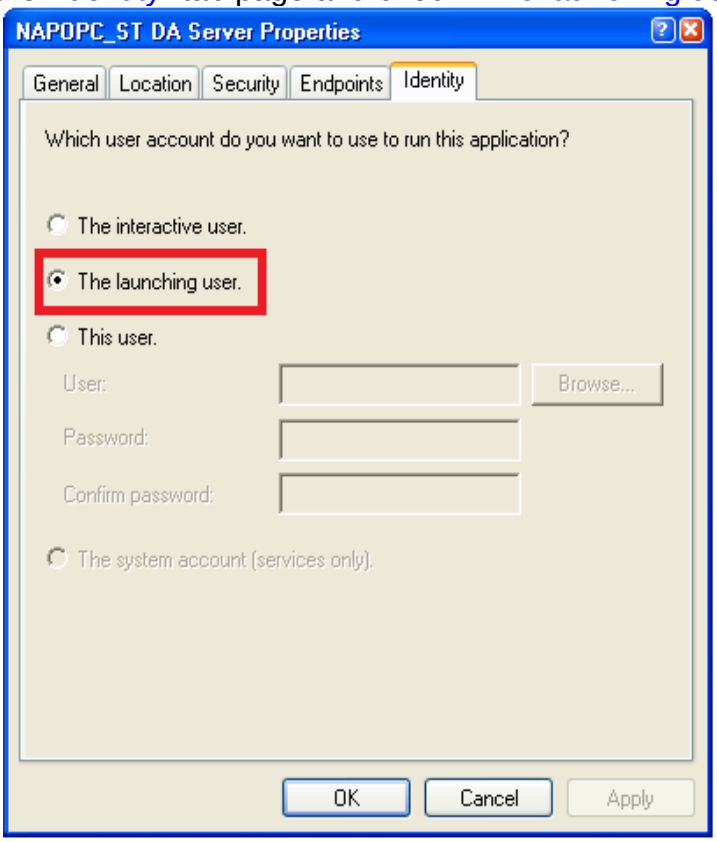
Step 9: Right click on the "NAPOPC_ST DA Server" of "DCOM Config" button and select "Properties".



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



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



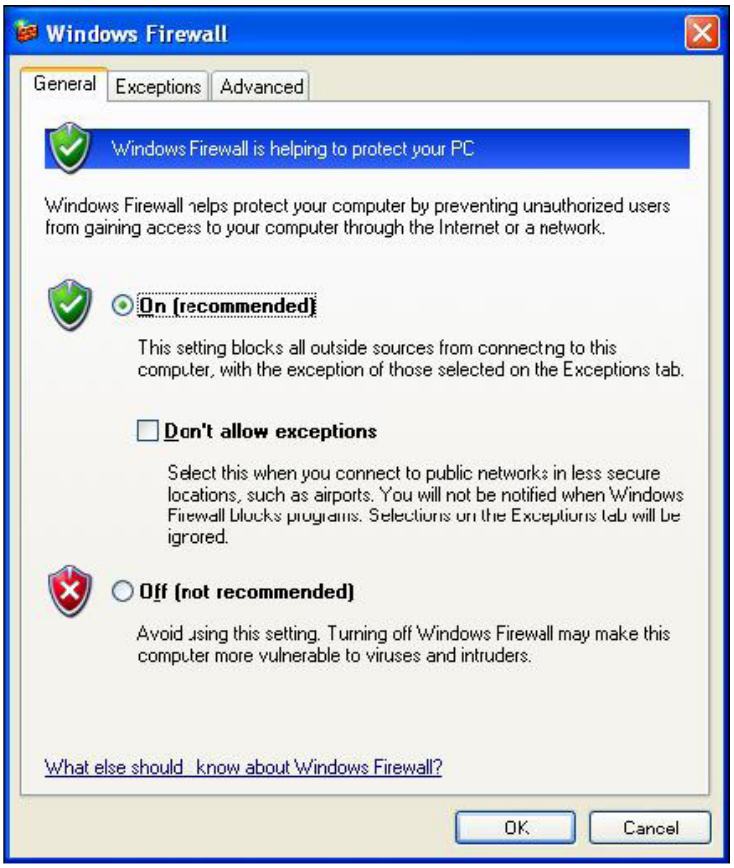
Step 12: Restart PC

4.1.5 Configuring On the Client Site (XPAC)

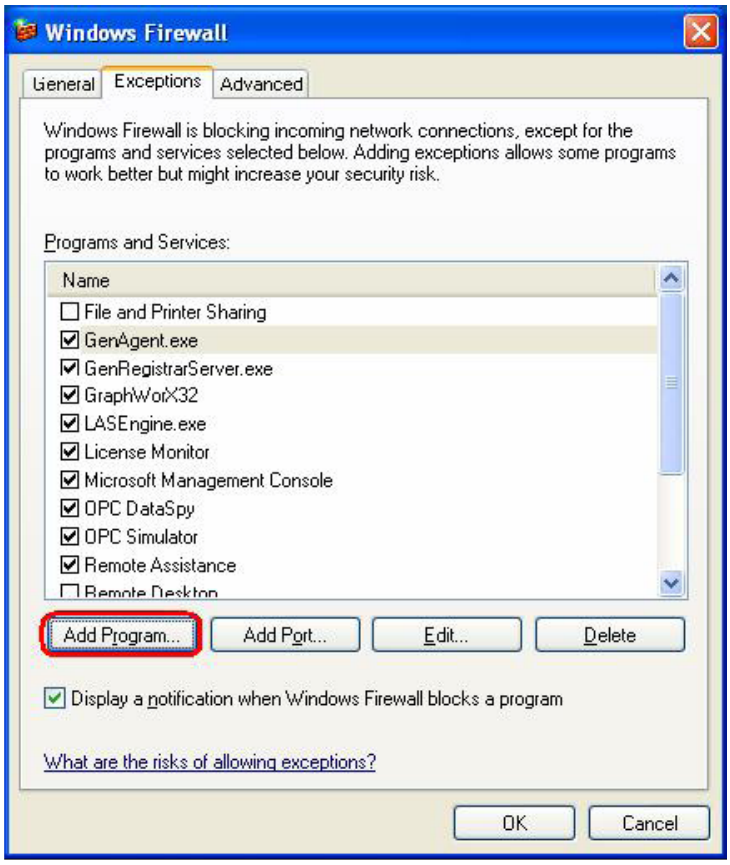
Configuring the Firewall

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.

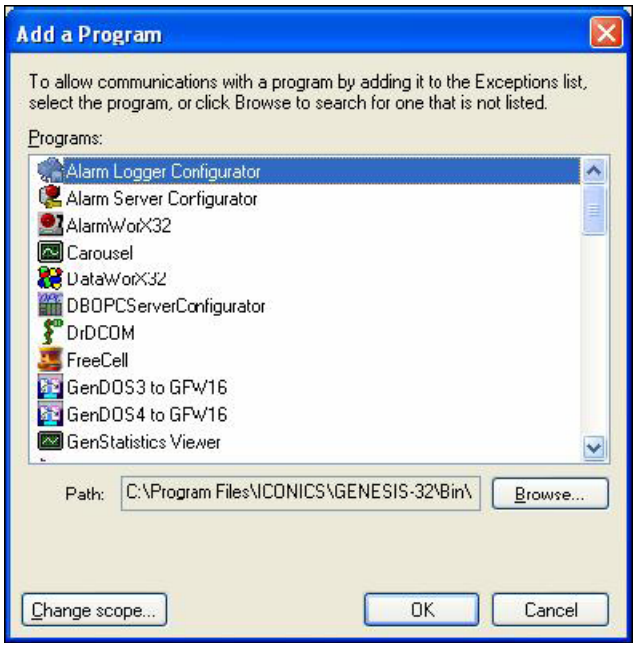


Step 2: Select the .Exceptions tab and add all OPC Clients and Servers to the exception list. Also add Microsoft Management Console (used by the DCOM configuration utility in the next section) and the OPC utility OPCEnum.exe found in the Windows\System32 directory.



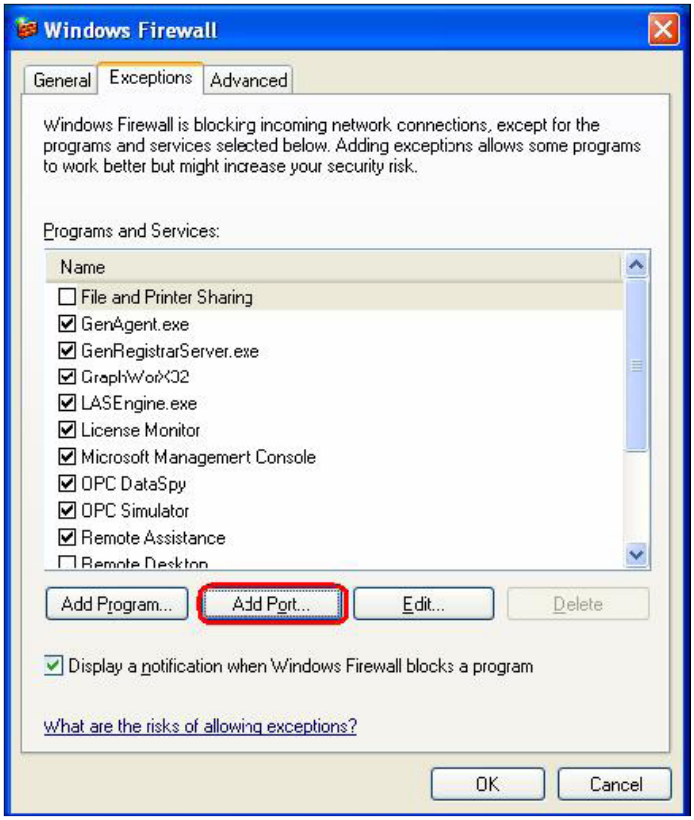
In the Add a Program dialog, there is a listing of most applications on the machine, but note that not all of them show up on this list. Use the "Browse" button to find other executables installed on the computer.

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.

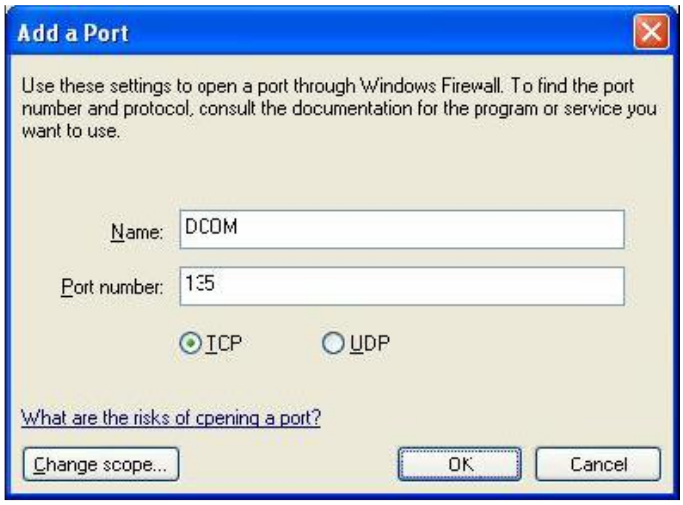


Step 3: Add TCP port 135 as it is needed to initiate DCOM communications, and allow for incoming echo requests. In the Exceptions tab of the Windows Firewall,

click on Add Port.

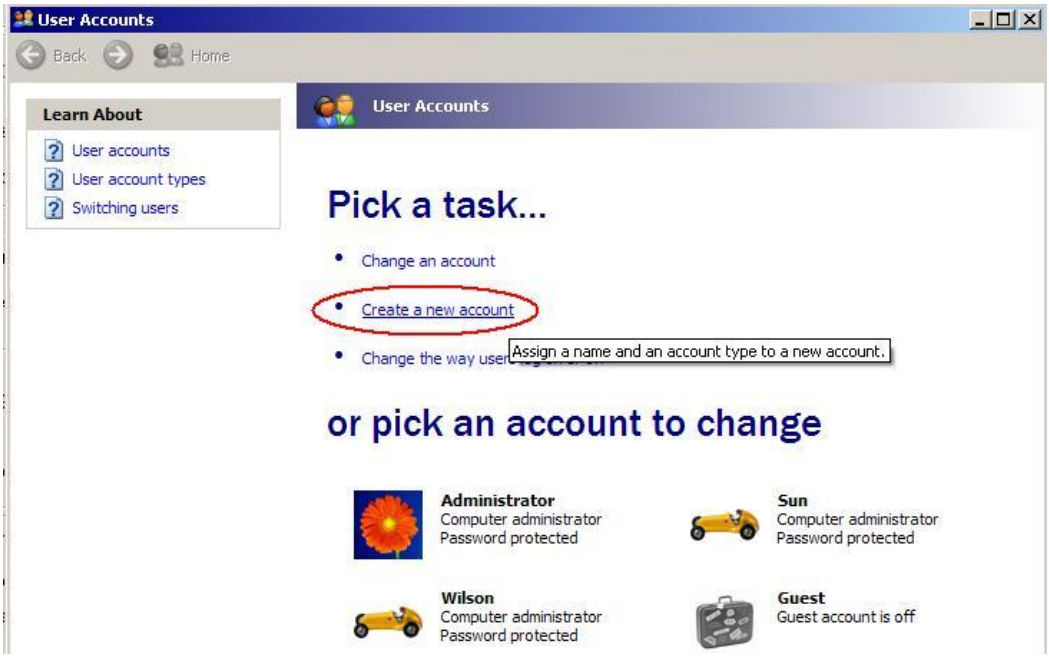


In the Add a Port dialog, fill out the fields as follows:
Name: DCOM
Port number: 135
Choose the TCP radio button



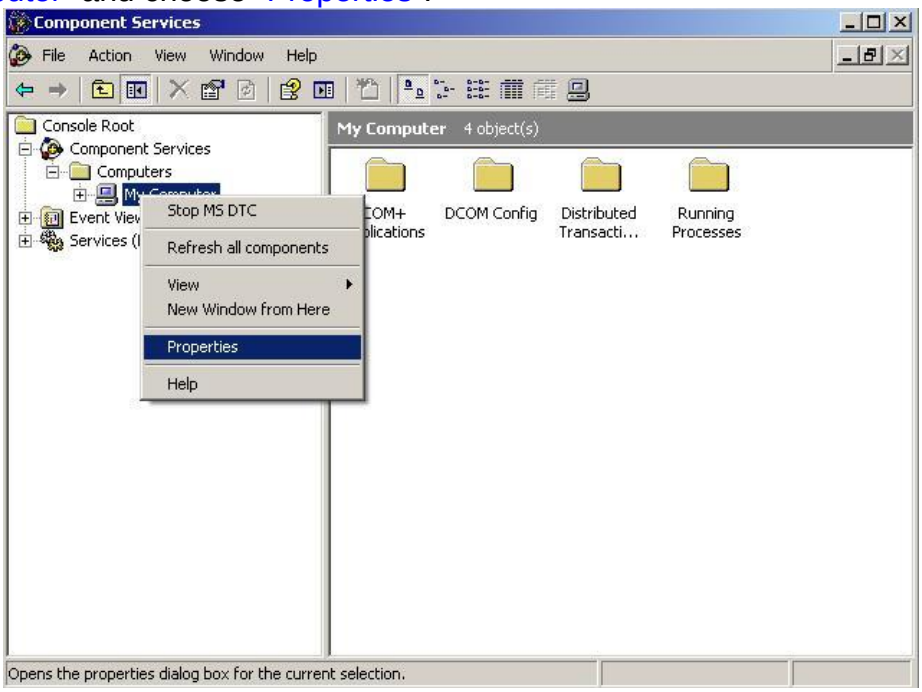
Creating the Account

Step 1: Create a account which must be the same with the account of server site.



Configuring DCOM

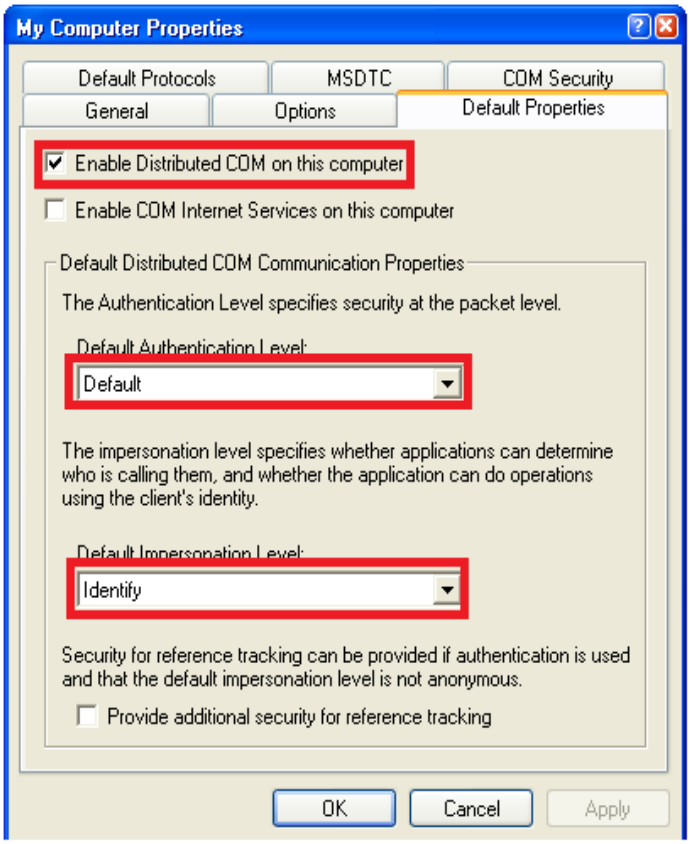
Step 1: Run the dcomcnfg.exe program to launch component services. Right clieck "My Computer" and choose "Properties".



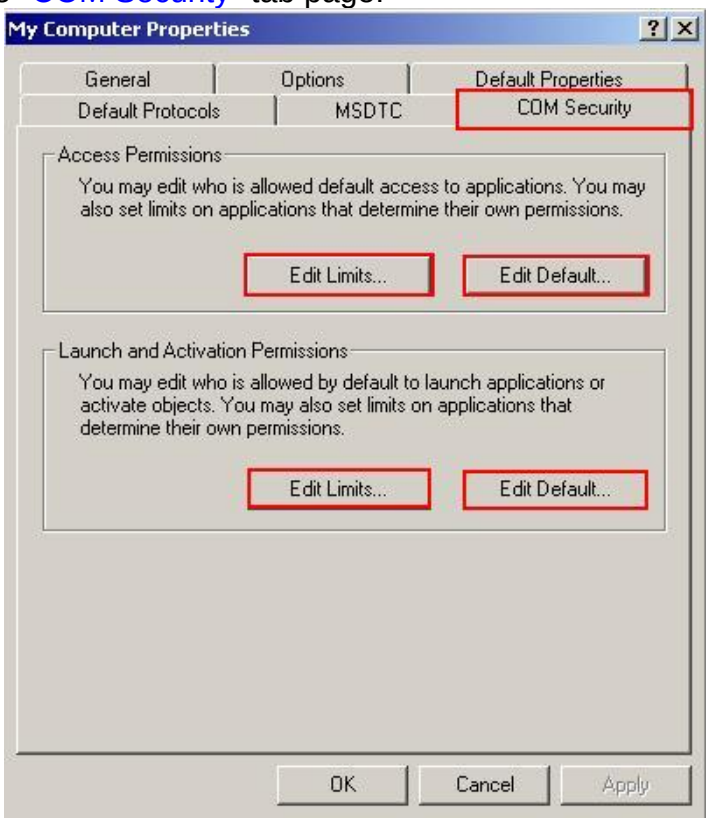
Step 2: Select the "Default Properties" tab page.

Step 3: Use the following settings:

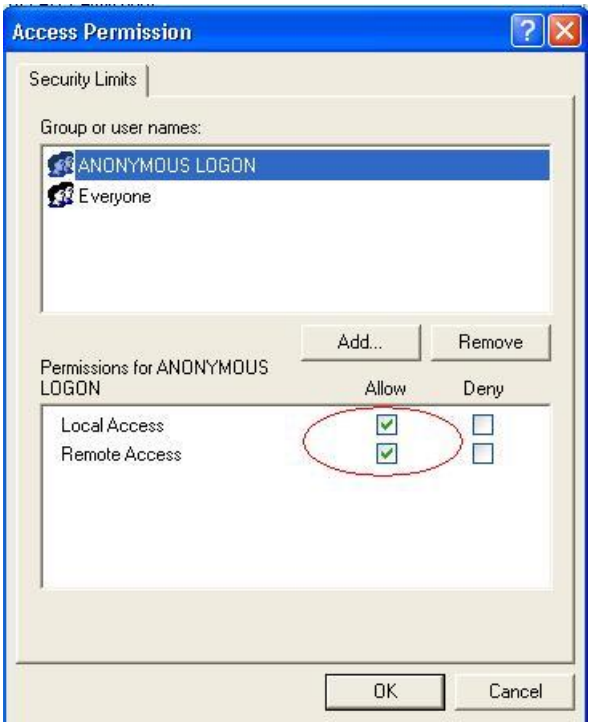
Field Name	Set to
Enable Distributed COM on this computer	Checked
Default Authentication Level:	Default
Default Impersonation Level:	Identify

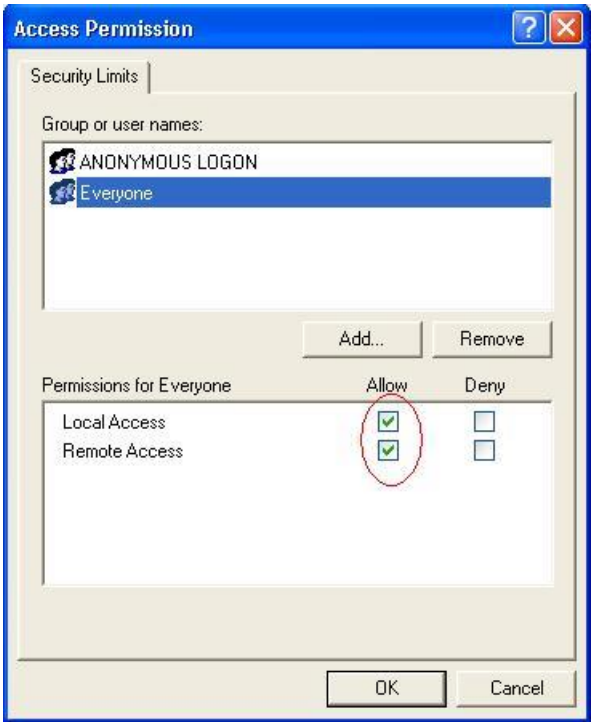


Step 4: Select the "COM Security" tab page.

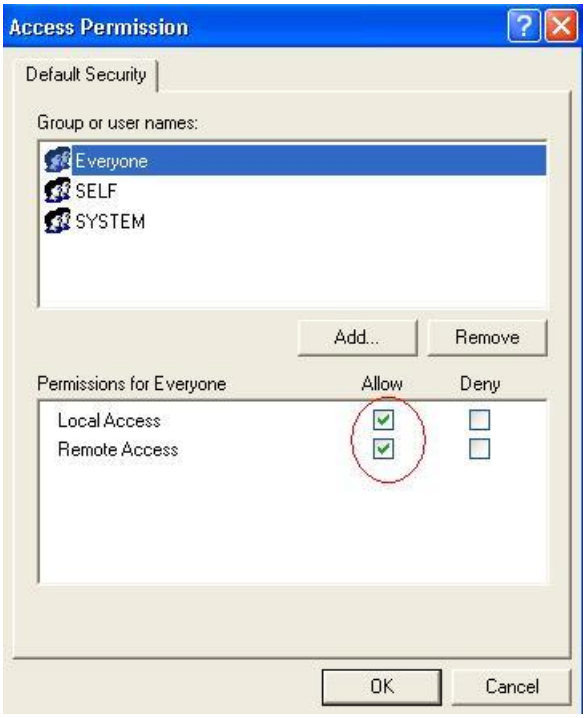


Step 5: Click on the "Edit Limits…" of "Access Permissions" button to set.

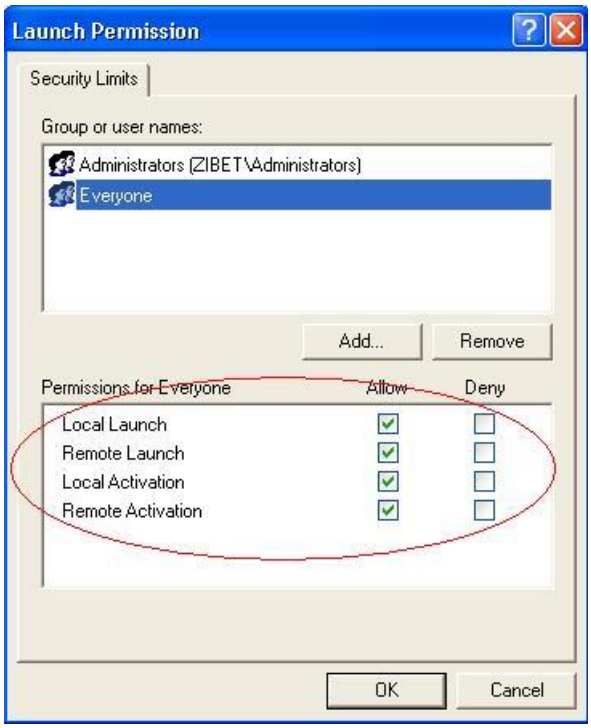




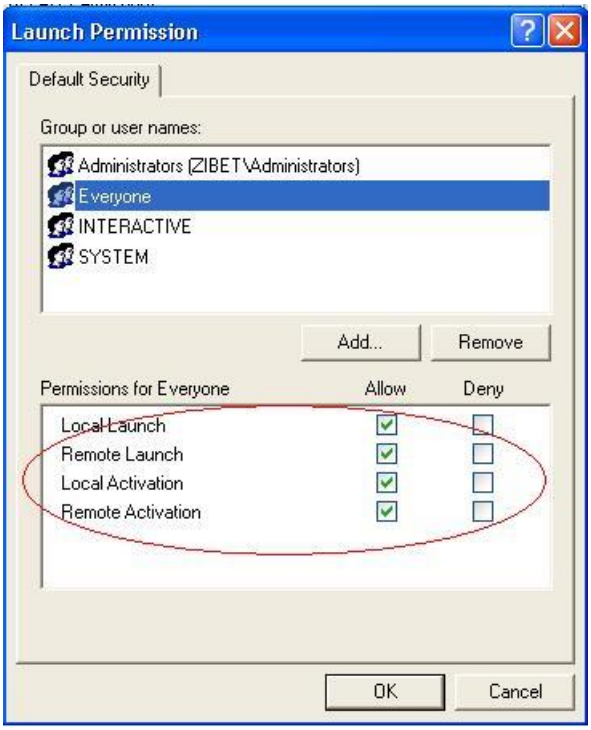
Step 6: Click on the "Edit Default…" of "Access Permissions" button to set.



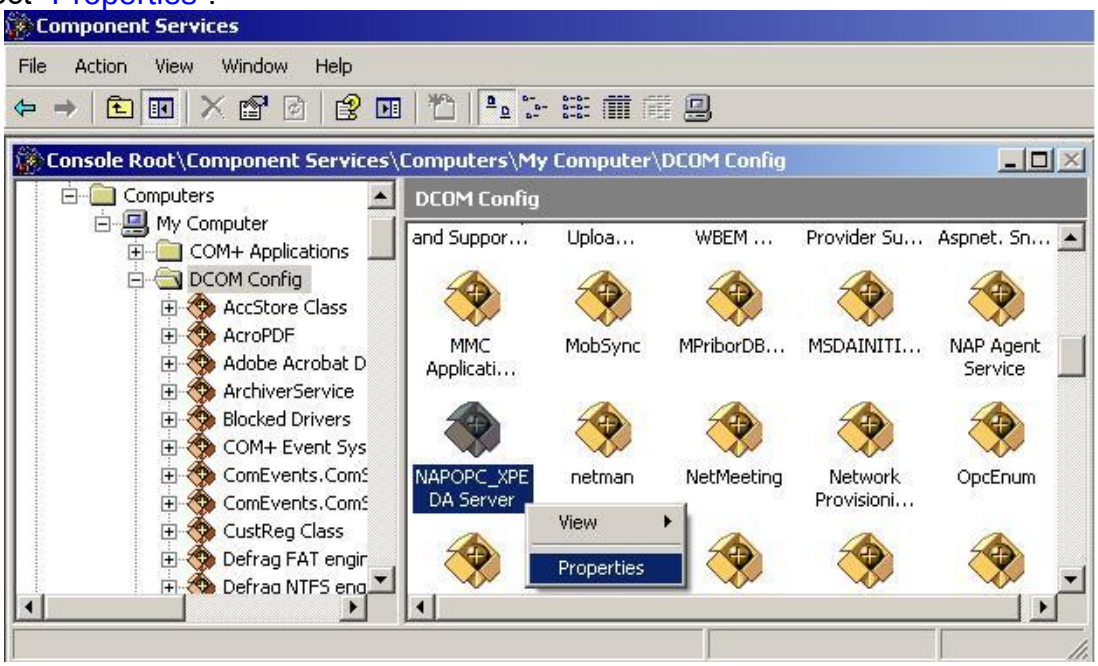
Step 7: Click on the "Edit Limits…" of "Launch and Activation Permissions" button to set.



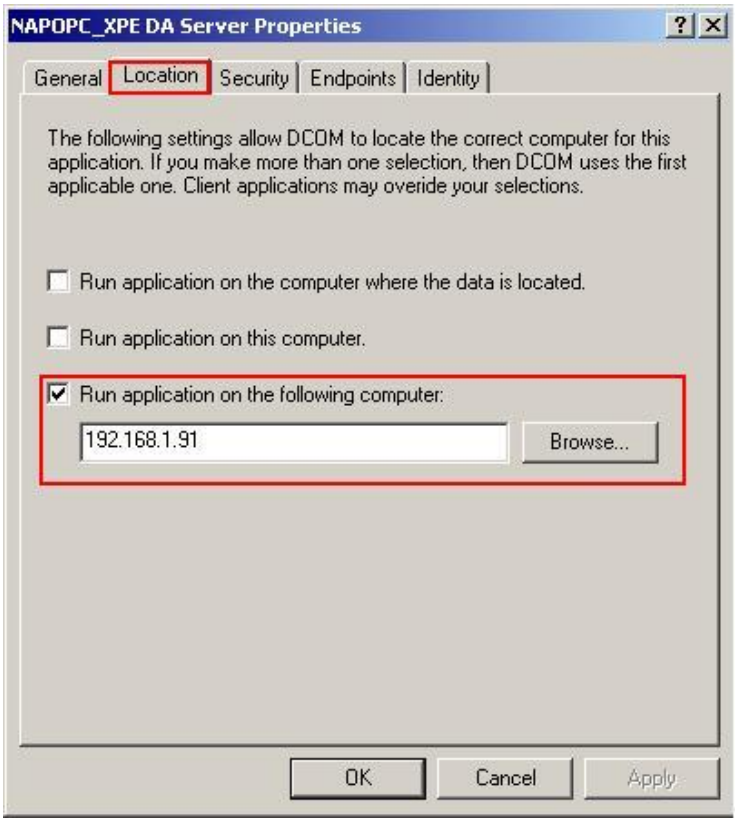
Step 8: Click on the "Edit Limits…" of "Launch and Activation Permissions" button to set.



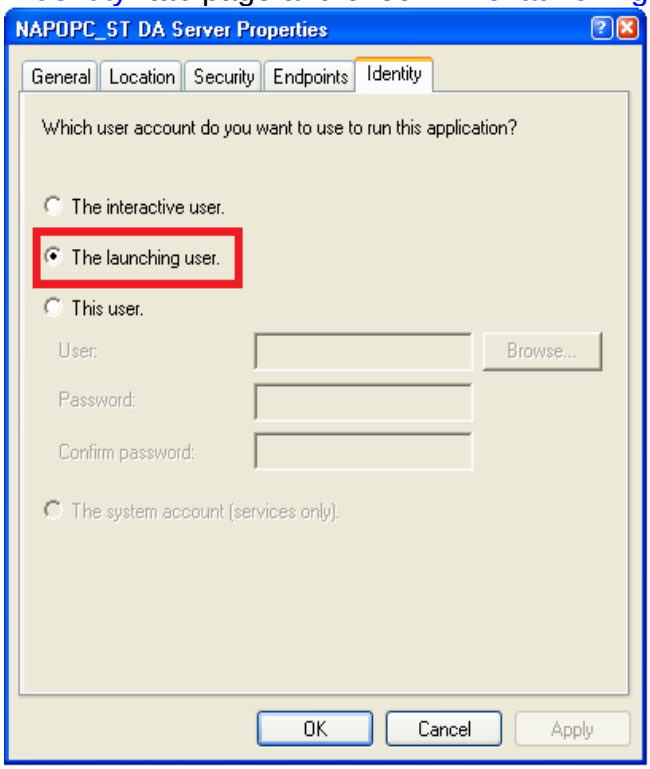
Step 9: Right click on the "NAPOPC_XPE DA Server" of "DCOM Config" button and select "Properties".



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



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



Step 12: Restart XPC

Shut Down Windows
Windows Embedded
Standard
Copyright © 1985-2008
Microsoft Corporation
Microsoft
What do you want the computer to do?
Log off Administrator
Log off Administrator
Shut down
Restart
Stand by
OK
Cancel
Help

5 Writing Client Program with VB

5.1 Programming with VB5

5.1.1 Overview of OPC & VB

Visual Basic language supports COM(Component Object Model). COM implementation from Visual Basic use what is called an "Automation" interface. The OPC Foundation supplies the source code of "Automation Wrapper" DLL which lets VB access OPC Servers and their underlying Groups and items. That's why we have to install the OPC DAC (Data Access Component.) software(see 2 Quick Start). After we install the OPC DAC software, we can use Automation Wrapper connects VB to OPC.

The following figure shows the architecture of object model for the automation wrapper. Because the OPC Server Object contains Group Objects and Items Objects by using Collection, OPC Browse Object can access the item data through the pointer of OPC Server Object.

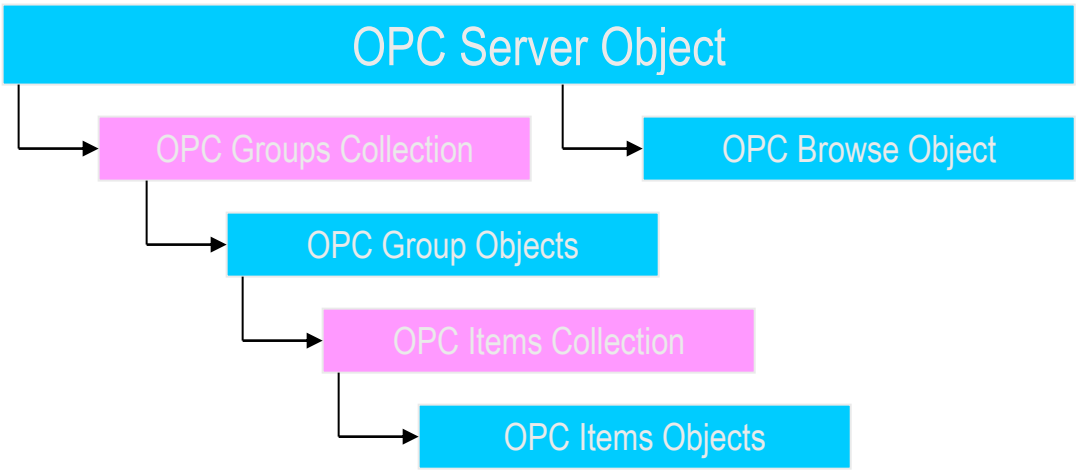


Figure 5-1-1. Object model for the Automation Wrapper

The following figure shows the architecture of OPC Server Object through the Automation Wrapper under COM/DCOM mechanism. The VB program wakes up the remote OPC Server Object through the automation wrapper object by DCOM mechanism.

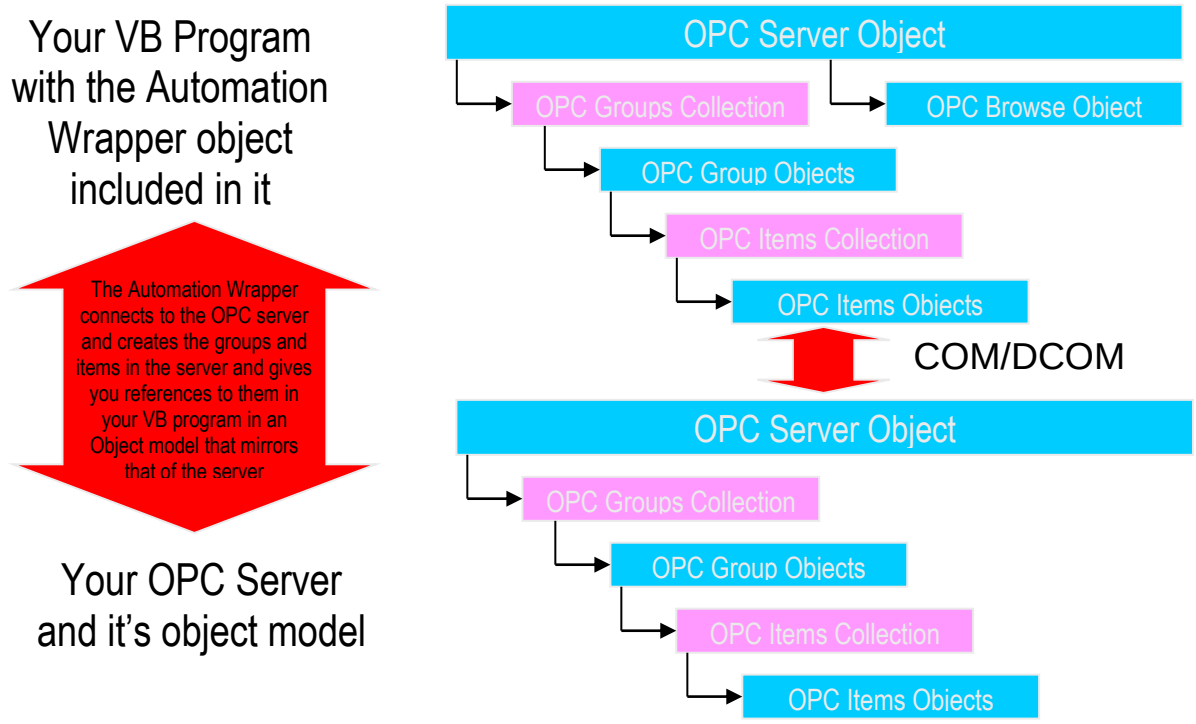


Figure 5-1-2. Architecture of OPC Server Object under COM/DCOM mechanism

5.1.2 Tools You Will Need to Build Your VB Client

If you want to build an OPC client in VB and test it, you will need the following tools.

- **Visual Basic 5 or 6** running on Windows XP/7/10

- **An OPC Server**
  At this manual, we use ICP DAS NAPOPC_ST Server as the demo. You can find it at your CD:\\Napdos\Napopcsvr\ or you can download it from the http://www.icpdas.com/download/7000/napopcsvr.htm

- **The OPC Automation Wrapper**
  You can find it from http://www.icpdas.com/download/7000/napopcsvr.htm

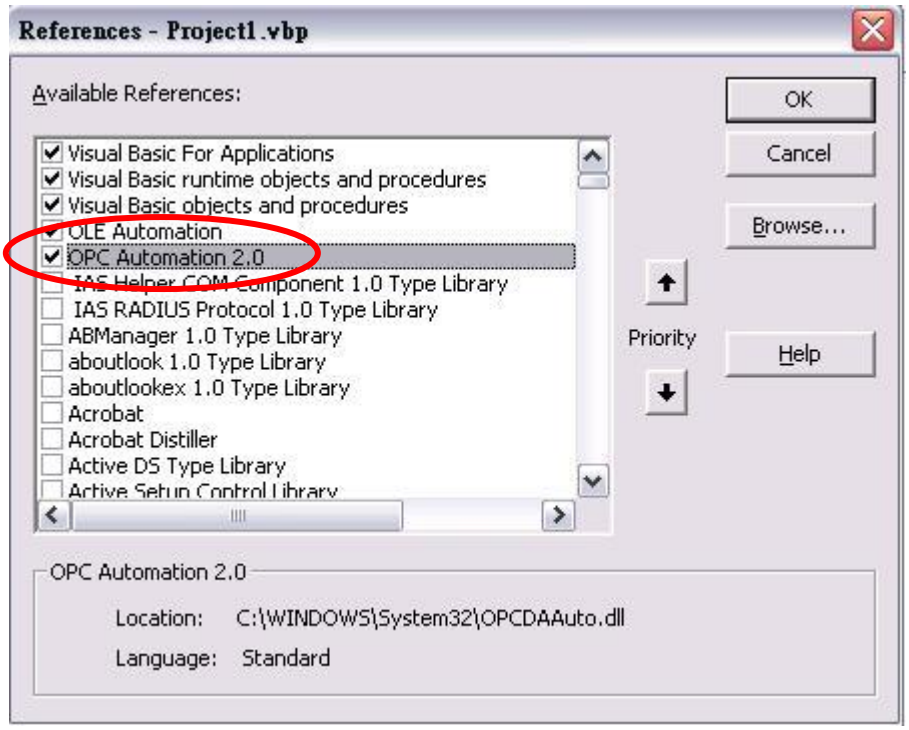
5.1.3 Building Your VB Client – Step By Step

At this section, we just focus on the key steps of building the VB client. If you want to know more information about OPC Automation 2.0 and the VB demo, please refer to the opcda20_auto.pdf in the C:\ICPDAS\NAPOPC_ST\Manual and the VB demo source code in the C:\ICPDAS\NAPOPC_ST\Client\VB5.

Step 1:

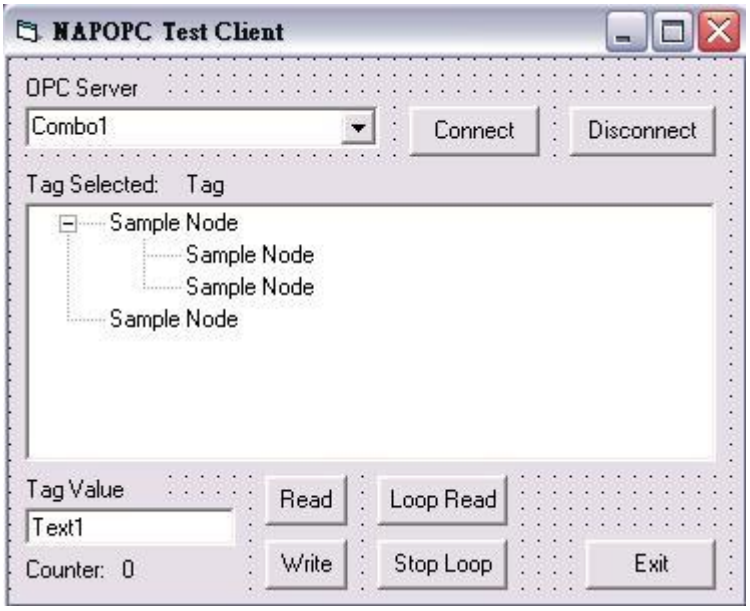
- Install OPCDAC on your PC
- Start a new VB project

- In VB, click on Project -> References on the VB menu bar
- The OPC Automation Wrapper appears on the dialog as "OPC Automation 2.0" – select it as shown here



Step 2:

First, you have to design your UI(User Interface). You can refer to the UI demo of VB program shown as below. Next, you need to declare some variables at the General Declarations area of VB code window. The most important types of variables are OPCServer, OPCGroup, and OPCBrowser. As the declaration, we can use several functions to read/write item values through Server and Group variables.




```
'Declare a new OPC Server object
Public Server As OPCServer
'Declare a new OPC Group object
```

```
Public Group As OPCGroup
'Declare a new Browser object
Public browser As OPCBrowser
```

Step 3:

You can call GetOPCServers() to scan the OPC Servers at your PC as the following codes.

```
'Declare a Variant Variable
Dim Servers As Variant
'Create a new OPC Server object
Set Server = New OPCServer
'Call GetOPCServers to scan the OPC Servers on your PC
Servers = Server.GetOPCServers("")
'Show the servers on the Combo box
Dim lastIndex As Integer
lastIndex = 0
For I = LBound(Servers) To UBound(Servers)
   cbServerList.AddItem Servers(I)
   If Servers(I) = lastServer Then lastIndex = I - 1
Next I
cbServerList.ListIndex = lastIndex
MousePointer = vbDefault
```

Step 4:

Next, you'll go ahead and add the code rights after you get your connection to the NAPOPC_ST Server. Please refer to the FillItems and Branch subroutine of VB demo program.

```
'Generate the tree of tags
Private Sub FillItems()
'Populate the sub branches in the browser tree
Public Sub Branch(Count As Integer, node1 As node)
```

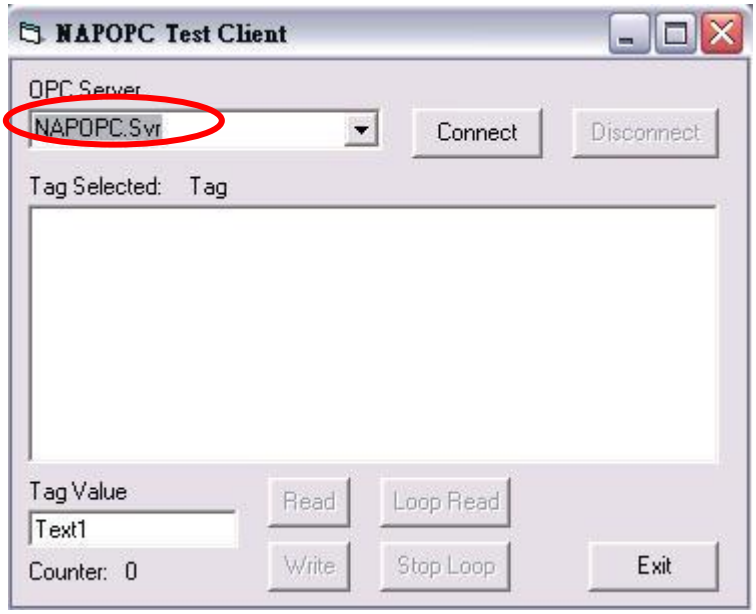
Step 5:

Now, you can add the code for the "Read" button and "Write" button. Please refer to the btnRead_Click and btnWrite_Click subroutine of VB demo program. In these two functions, the *anItem.Read* and the *anItem.Write* are two key methods.

```
'Read the OPCItem value after the read button press
Private Sub btnRead_Click()
'Write the value in the text box after the write button press
Private Sub btnWrite_Click()
```

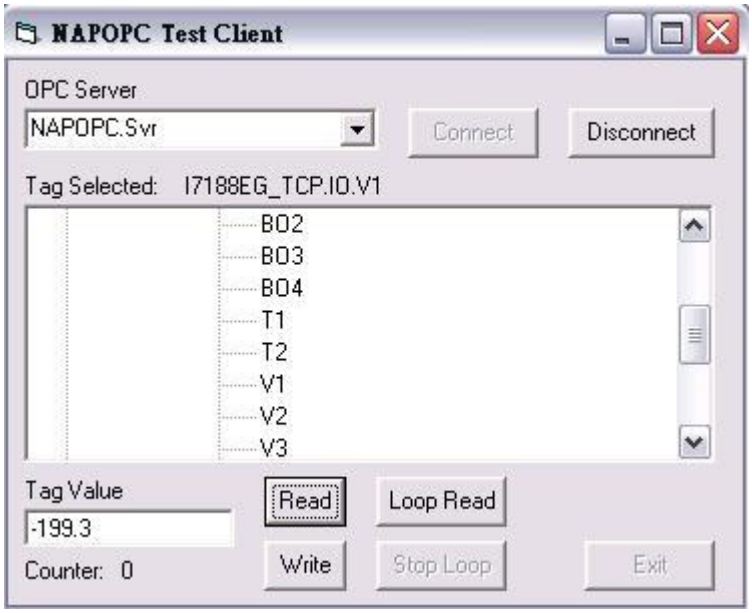
Step 6:

You can build the project and you will see the UI as below.



Step 7:

After you click on the "Connect" button, you will see the OPC Server tree list. You can choose one of them and click on the "Read" button. You will see the item value at the "Tag Value" field as below. You can also type the value you want to write in the text box and click on the "Write" button. (Refer to 4.1 Client Demo Program)



5.2 Programming with .Net

5.2.1 Limitations about .Net client programming

1. OPC DA component 2.0 must be installed. (In this case, OPC DA Component 1.0 can't be used)

2. Development was done on a Windows XP/7/10 system using Microsoft .NET Development with .Net Framework playform and any new release version of .Net will need to rebuild these codes.

3. Testing was done on following operation system, any others might not work
   - Windows XP professional
   - Windows 7 professional
   - Windows 10 professional

5.2.2 Tools – You Need to Build Your .Net Client

If you want to build an OPC VB .Net or VC# client and test it, you will need following tools.

- **Microsoft .NET Development** running on Windows XP/7/10

- **OPC Automation 2.0**
  You can install NAPOPC_ST DA Server(From CD:\\Napdos\napopcsvr or download it from http://opc.icpdas.com/download.htm). The NAPOPC_ST DA Server installation will install OPC Automation 2.0 automatically.

- **An OPC Server**
  At this manual, we use ICPDAS NAPOPC_ST Server as the demo. You can find it at your CD:\\Napdos\napopcsvr\ or download it from the http://opc.icpdas.com/download.htm

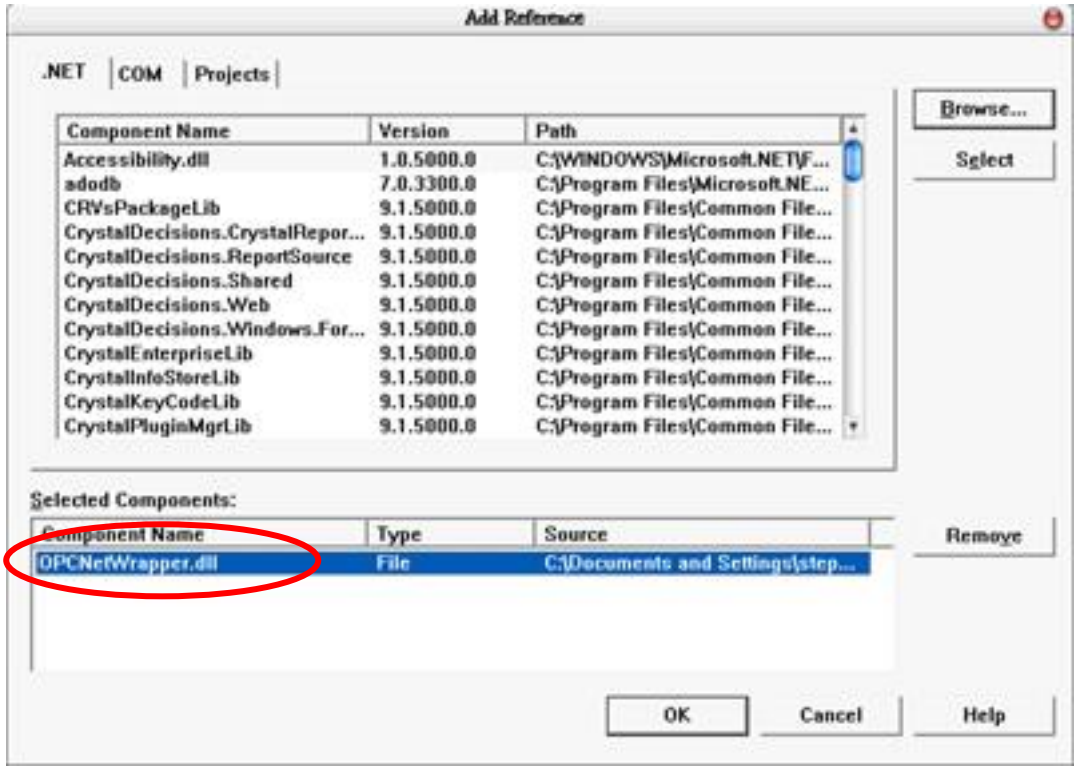
- OPC .Net wrapper named "**OPCNetWrapper.dll**".
  After you install NAPOPC_ST DA Server (From http://opc.icpdas.com/download.htm), you can find it in Root\\ ICPDAS\ NAPOPC_ST\ Client\ OPC_NetClientDemo\VBOPCClient_Demo

- **OPCNETWrapper.pdf**
  After you install NAPOPC_ST DA Server (From http://opc.icpdas.com/download.htm), you can find it in Root\\ICPDAS\NAPOPC_ST\Manual

5.2.3 Building Your VB.Net Client – Step By Step

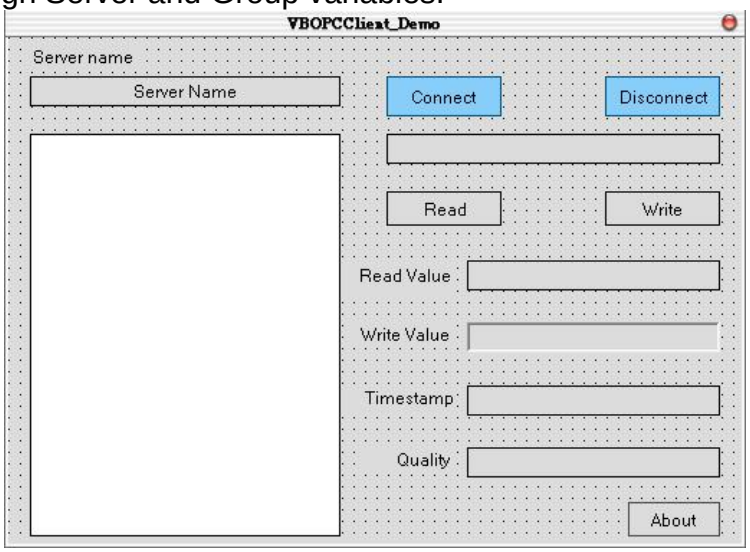
Step 1:

- Install OPC DA Component 2.0 on your PC
- Start a new VB.Net project with "Windows Application Program"
- In VB.Net, click Project -> Add References on the VB.Net menu bar
- Click Browse… to select "OPCNetWrapper.dll" as shown below



Step 2:

First, you must Import following declarations "ICPDAS.OPC","ICPDAS.OPCDA" and "ICPDAS.OPC.NET". Second, you could design your UI (User Interface). You can refer to the UI of VB.Net program shown as below. Third, you need to declare some variables at the General Declarations area of VB.Net code window. The most important types of variables are ICPDAS_OPCServer, ICPDAS_SyncIOGroup, and ServerTreeBrowser. As the declaration, we can use several functions to read/write tag values through Server and Group variables.




```
'Declare a new ICPDAS_OPCServer object
Dim Svr As ICPDAS_OPCServer
'Declare two new ICPDAS_SyncIOGroup objects
Dim ReadWriteGroup, Grp As ICPDAS_SyncIOGroup
'Declare a new ServerTreeBrowser object
```

```
Dim TagTree As ServerTreeBrowser
'Declare two new RefreshGroup objects
Dim AsyncRefrGroup, rGrp As RefreshGroup
'Declare a new TagDef object
Dim TagData As TagDef
'Declare a new String variable for OPC Server ProgID
Dim ServerName As String
```

Step 3:

You can set ServerName as "NAPOPC.Svr.1" and pass it to ICPDAS_OPCServer for connecting.

```
'Set a ProgID to ServerName
ServerName = "NAPOPC.Svr.1"
'Create a new OPC Server object
Svr = New ICPDAS_OPCServer
'Connect to NAPOPC DA Server
Svr.Connect (ServerName)
```

Step 4:

Now, you'll go ahead and add the code rights after you get your connection to the NAPOPC_ST Server. Please refer to following TagReadWrite and DataChangeHandler subroutine of VB demo program.

```
'Handles of data change callbacks
Public Sub DataChangeHandler (ByVal sender As Object, ByVal e As
DataChangeEventArgs)

Private Sub TagReadWrite (ByVal lgrp As ICPDAS_SyncIOGroup, ByVal lrgrp As
RefreshGroup, ByVal iTree As ServerTreeBrowser, ByVal tagId As String)
```

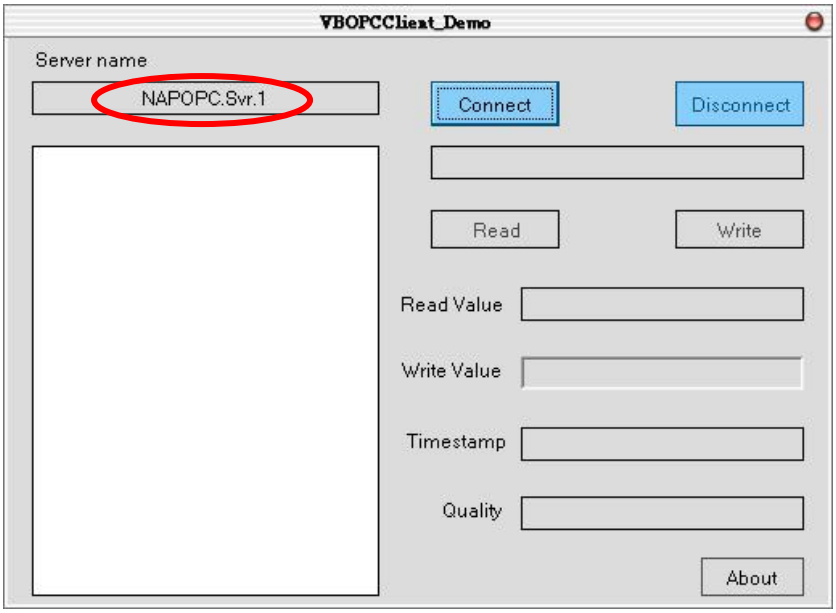
Step 5:

Now, you can add codes for the "Read" button, "Write" button and "TreeView" AfterSelect function. Please refer to the btnRead_Click, btnWrite_Click and tvTags_AfterSelect subroutine of VB.Net demo program. In these three functions, the *Grp.Read*, the *Grp.Write and the* TagTree.TagName are three key methods.

```
'Read the OPCTag value after the read button press
Private Sub btnRead_Click ()
'Write the value in the text box after the write button press
Private Sub btnWrite_Click ()
'The action after selecting the tag
Private Sub tvTags_AfterSelect ()
```

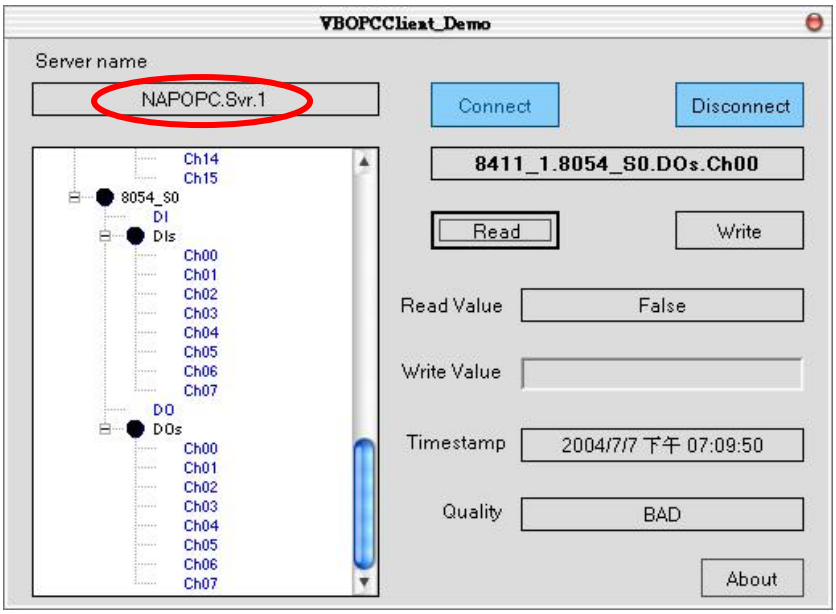
Step 6:

You can build the project and you will see the UI as below.



Step 7:

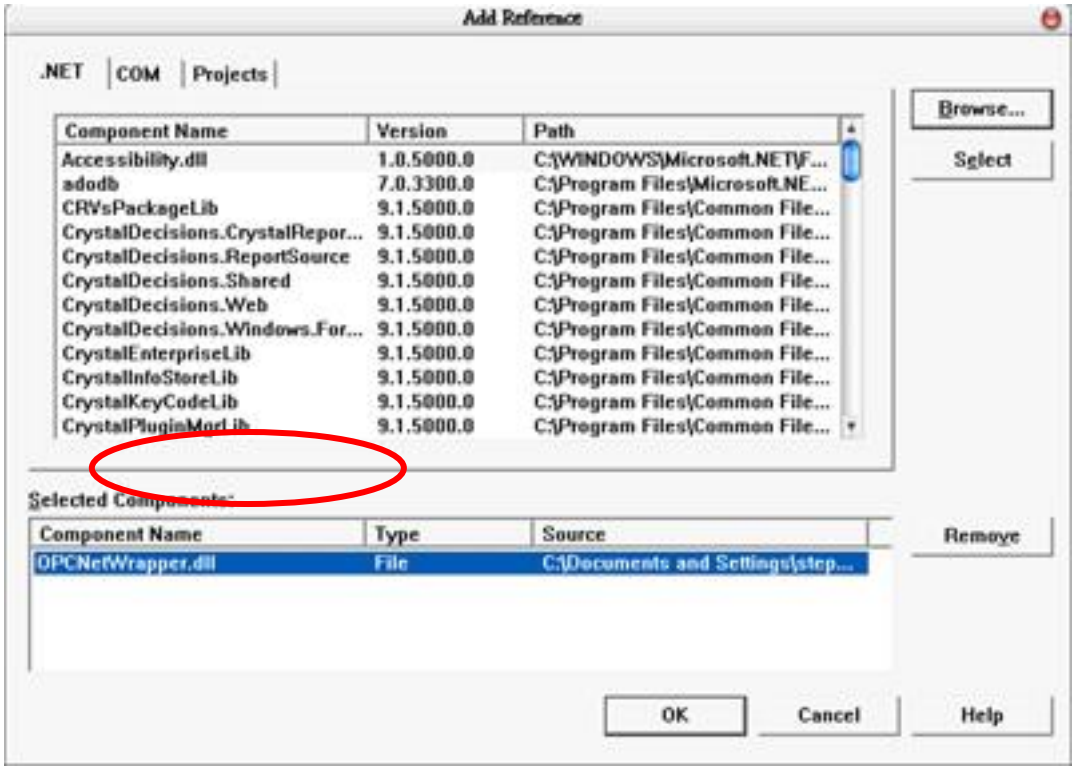
After you click on the "Connect" button, you will see the OPC Server tree list. You can choose one of them and click on the "Read" button. You will see the tag value at the "Read Value" field as below. You can also type the value you want to write in the "Write Value" and click on the "Write" button. (Refer to 4.2 .Net Client Demo Program)



5.2.4 Building Your VC#.Net Client – Step By Step

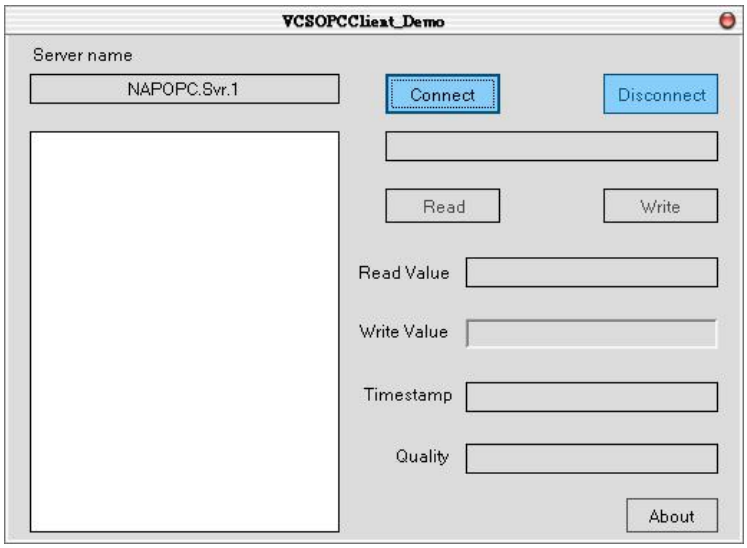
Step 1:

- Install OPC DA Component 2.0 on your PC
- Start a new VC#.Net project with "Windows Application Program"
- In VC#.Net, click Project -> Add References on the VC#.Net menu bar
- Click Browse… to select "OPCNetWrapper.dll" as shown below



Step 2:

First, you must using directive as following declarations: "ICPDAS.OPC","ICPDAS.OPCDA" and "ICPDAS.OPC.NET". Second, you could design your UI (User Interface). You can refer to the UI of VC#.Net program shown as below. Third, you need to declare some variables at the General Declarations area of VC#.Net code window. The most important types of variables are ICPDAS_OPCServer, ICPDAS_SyncIOGroup, and ServerTreeBrowser. As the declaration, we can use several functions to read/write tag values through Server and Group variables.




```
'Declare a new ICPDAS_OPCServer object
ICPDAS_OPCServer                    Svr = null;
```

```
'Declare two new ICPDAS_SyncIOGroup objects
ICPDAS_SyncIOGroup                    ReadWriteGroup, Grp;
'Declare a new ServerTreeBrowser object
ServerTreeBrowser                     TagTree;
'Declare two new RefreshGroup objects
RefreshGroup                          AsyncRefrGroup, rGrp;
'Declare a new TagDef object
TagDef                                TagData;
'Declare a new String variable for OPC Server ProgID
string                                ServerName;
```

Step 3:

You can set ServerName as "NAPOPC.Svr.1" and pass it to ICPDAS_OPCServer for connecting.

```
'Set a ProgID to ServerName
ServerName = "NAPOPC.Svr.1;"
'Create a new OPC Server object
Svr = new ICPDAS_OPCServer ();
'Connect to NAPOPC DA Server
Svr.Connect (ServerName);
```

Step 4:

Now, you'll go ahead and add the code rights after you get your connection to the NAPOPC_ST Server. Please refer to following TagReadWrite and DataChangeHandler subroutine of VC# demo program.

```
'Handles of data change callbacks
public void DataChangeHandler ( object sender, DataChangeEventArgs e )

public void TagReadWrite ( ICPDAS_SyncIOGroup grp, RefreshGroup rgrp,
ServerTreeBrowser iTree, string tagId)
```

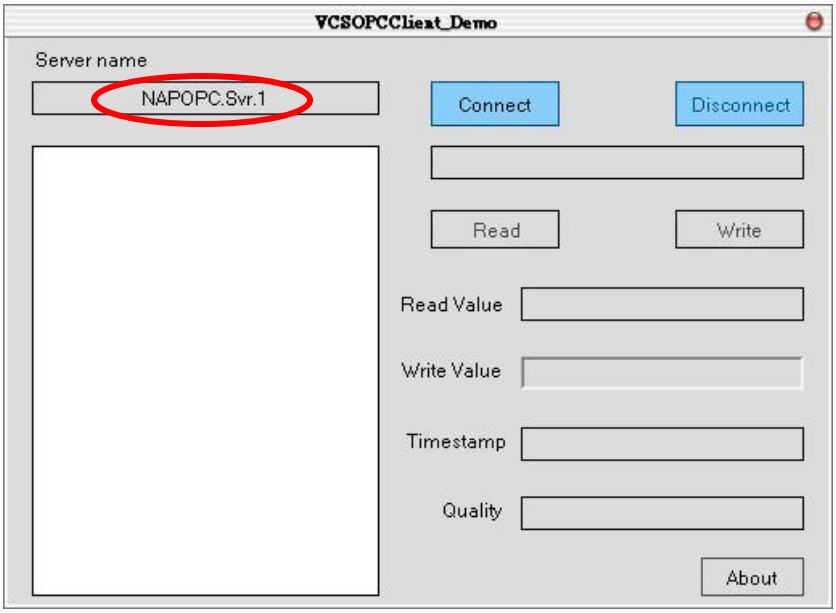
Step 5:

Now, you can add codes for the "Read" button, "Write" button and "TreeView" AfterSelect function. Please refer to the btnRead_Click, btnWrite_Click and tvTags_AfterSelect subroutine of VC#.Net demo program. In these three functions, the *Grp.Read*, the *Grp.Write and the* TagTree.TagName are three key methods.

```
'Read the OPCTag value after the read button press
private void btnRead_Click (object sender, System.EventArgs e)
'Write the value in the text box after the write button press
private void btnWrite_Click(object sender, System.EventArgs e)
'The action after selecting the tag
private void tvTags_AfterSelect(object sender,
                                System.Windows.Forms.TreeViewEventArgs e)
```

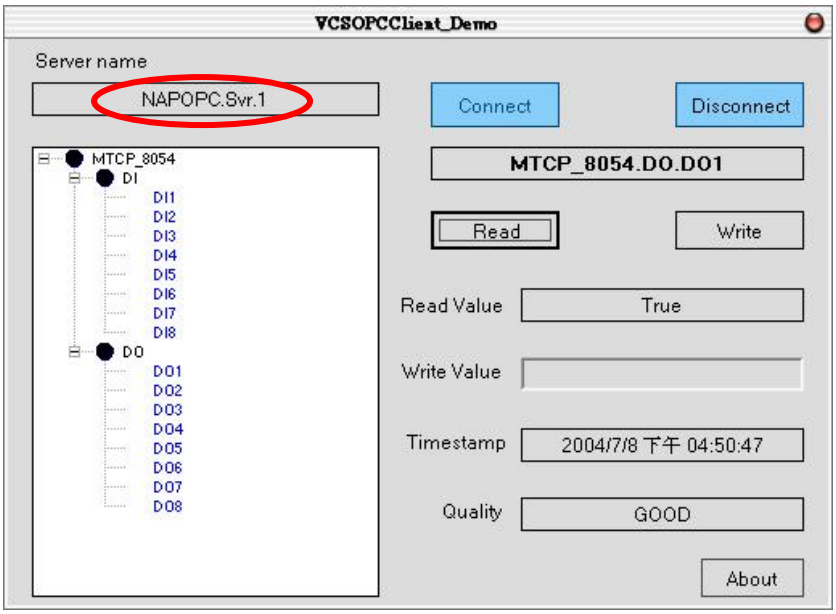
Step 6:

You can build the project and you will see the UI as below.



Step 7:

After you click on the "Connect" button, you will see the OPC Server tree list. You can choose one of them and click on the "Read" button. You will see the tag value at the "Read Value" field as below. You can also type the value you want to write in the "Write Value" field and click on the "Write" button. (Refer to 4.2 .Net Client Demo Program)



6 Changes List

6.1 New features of NAPOPC version 3.0

The new features of NAPOPC_ST version 3.0 are briefly described in this chapter.

6.1.1 New IO Kernel

NAPOPC_ST version 3.0 uses new IO kernel "IOCtrl.DLL" and "DCON_PC.DLL" to elevate IO communication performance. Also, the new IO kernel architecture allows customers only to update "module.ini" for getting new module support.

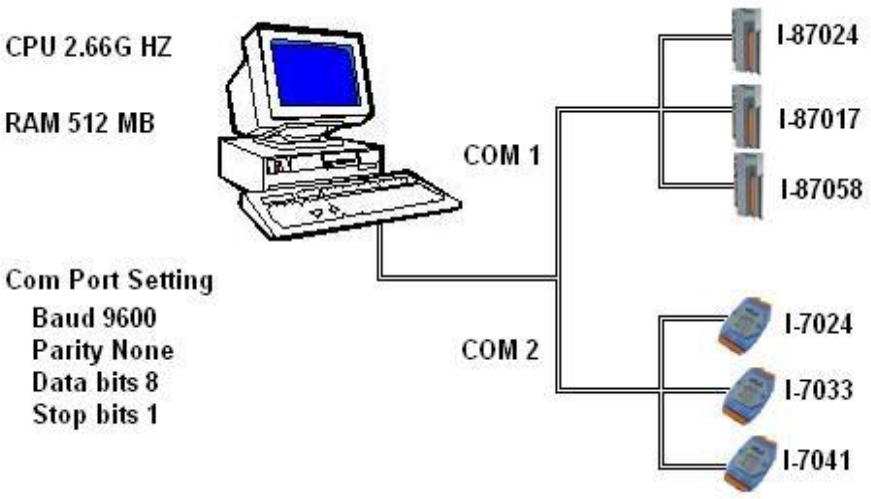
6.1.2 Customized Module/Device Polling Time

NAPOPC version 3.0 allows customers to optimize their communication by setting "Pending Time" in "Device Dialog ". Customers have chance to spread more time resource to other modules which are connected with each other. Please refer to "3.1 Optimize Your Communication" in detail.

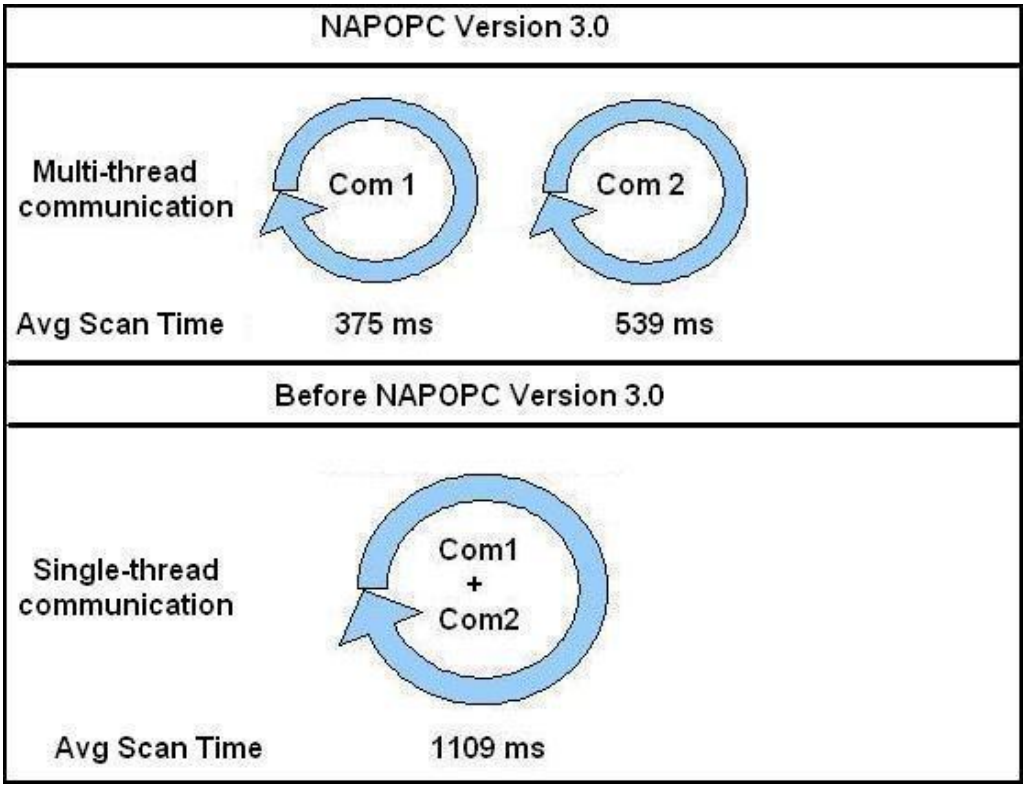
6.1.3 Multi-Thread Communication

NAPOPC version 3.0 uses multi-thread architecture to organize module and device communication. Before NAPOPC version 3.0, NAPOPC uses single-thread to deal with module and device communication. Therefore, if NAPOPC connects many modules and devices, accessing time will increase observably. By NAPOPC version 3.0, customers have chances to group modules and devices into several parts, and connect via different COM port. One COM port works in one thread. All TCP communications belong to one thread.

Here is an example to show the difference of average scan time between NAPOPC version 3.0 and before NAPOPC version 3.0.



Under the same architecture, the average scan time in NAPOPC version 3.0 is 375 ms and 539 ms separately. Before NAPOPC version 3.0, the average scan time is 1109 ms.



6.1.4 Miscellaneous

- Remove "import/export CSV file" function
- Remove "debug logging" function

6.2 New features of NAPOPC version 3.09

6.2.1 Support Remote Procedure Call with Quicker/UPC

NAPOPC version 3.09 supports RPC(Remote Procedure Call) communication with Quicker and UPC(ICP DAS Universal Protocol Converter). Based on RPC service, NAPOPC can synchronize OLE data with Quicker and UPC automatically. The behavior of changing data between "Quicker/UPC" and NAPOPC is not polling from NAPOPC but sending from "Quicker/UPC". This mechanism can transfer data effectively and shorten response time.

6.2.2 Switch of Single-Thread and Multi-Thread

NAPOPC version 3.09 supports "Communication Mechanism" option at "Options/ Configurate Initial Status". This option lets user define the communication behavior of NAPOPC. Generally, "Multi-Thread" is the best choice for high performace. However, for some particular OPC clients which can not work smoothly under "Multi-Thread" communication, user can choose "Single-Thread" instead.

6.3 New features of NAPOPC_ST version 3.11

6.3.1 Rename NAPOPC to be NAPOPC_ST

For better integration, we rename NAPOPC DA Server to be NAPOPC_ST DA Server. We change some UI display such as icon, logo, and denomination.

6.3.2 Bug Fix

- Fix "Unsupported_Device" bug when searching RU87PN
- Fix modbus tags showing error when tag property length is 4 bytes

6.4 Features of NAPOPC_ST version 3.20

The new features of NAPOPC_ST version 3.20 are briefly described in this section.

6.4.1 Support ET-7000 Search

NAPOPC_ST version 3.20 supports "ET-7000 Search" option at "Add/Search Modules". This option lets NAPOPC_ ST can search not only the modules communicating via COM port but also ET-7000 modules via Ethernet automatically.

6.4.2 Support ZB-2K I/O

NAPOPC_ST version 3.20 supports "ZB-2K" I/O modules at "Select Device" dialog. User can easily choose ZB-2K module ID to add ZB-2K module. And then use "Generate Tags" function to generate ZB-2K properties automatically.

6.4.3 Support FRnet Module

NAPOPC_ST version 3.20 supports "FRnet" I/O modules at "Select Device" dialog. User can easily choose FRnet module ID to add FRnet module. And then use "Generate Tags" function to generate FRnet properties automatically.

Note: If you want to use FRnet modules in NAPOPC_ST, please download FRB driver and install it first.
Download link: http://www.icpdas.com/download/frnet/index.htm

6.4.4 Support Account Selection

NAPOPC_ST version 3.20 supports "Recent File Source" selection at "Options/Configurate Initial Status" dialog. User can choose "Administrator Account" or "Current User Account" to decide registry location for some DCOM application scenario.

6.4.5 UI Modification

For more protocol support, NAPOPC_ST version 3.20 modifies "Select Device" user interface. NAPOPC_ST version 3.20 separates three groups of "DCON", "FRnet", and "Modbus" from one property sheet to clarify the user interface of device settings. Also, it limits the dialog size to an allowable boundary.

6.5 Features of NAPOPC_ST version 3.30

The new features of NAPOPC_ST version 3.30 are briefly described in this section.

6.5.1 Support FRnet module via RPC communication

NAPOPC_ST version 3.30 supports "FRnet" I/O modules via RPC communication. User can easily create RPC device to synchonize "FRnet" I/O modules plugged in WinPAC(See 1.6.2 Adding A New RPC Controller).

6.5.2 Support host watchdog setting for I-7000 module

NAPOPC_ST version 3.30 supports "WatchDog" settings for I-7000 module when it enables watchdog function. If user enables I-7000 watchdog function by "DCON Utility", NAPOPC_ST can enable and give an appropriate timeout value for I-7000 module to support this function(See 1.6.1 Adding A New I-7K/I-8K/I-87K/ZB-2K I/O Module).

Note: The "Communication Mechanism" must be "Multi-Thread".

6.5.3 Support usage of USB hardkey to enhance functionality

NAPOPC_ST version 3.30 supports usage of USB hardkey to enhance functionality. User can purchase USB hardkey to enhance the functionality of NAPOPC_ST DA Server by "License Manager" (See 1.16 License Manager).

6.6 Features of NAPOPC_ST version 3.54

The new features of NAPOPC_ST version 3.54 are briefly described in this section.

6.6.1 Support CSV file to create a NAPOPC_ST project based on Modbus TCP

NAPOPC_ST version 3.54 supports to import a CSV file to create a NAPOPC_ST project based on the Modbus TCP. The new contents introduce the new function to import a CSV file and the format of the CSV file. (See 1.3 File > Import CSV File > Modbus TCP).

7 Reference

FAQ.txt

The frequently asked questions and answers.
http://opc.icpdas.com/faq_st.htm

GetStart.PDF

This manual can be downloaded from below link.
http://www.icpdas.com/download/7000/manual.htm
It describes the following topics:

1. Connecting modules
2. The 7000 Utility user's manual.
3. Introduction to NAP7000P
4. Introduction to NAP7000X
5. Dual Watchdog
6. FAQ for 7000

OPCNetWrapper.PDF

This document describes how to use OPC .NET wrapper for .NET programming.

OPCOvw.PDF

The OPC overview - written by the OPC Foundation.

OPCCommn.PDF

The OPC common interface - specifications by the OPC Foundation.

OPCDA20_Cust.PDF

The OPC DA v2.0 custom interface - specifications by the OPC Foundation.

OPCDA20_Auto.PDF

The OPC DA v2.0 automation interface - specifications by the OPC Foundation.

OPC Foundation Web Site

http://www.opcfoundation.org/