

**User Manual**

**OPC Server  
ModbusTCP  
Configuration Manual**

# Table of Contents

<b>Chapter 1 Hardware Settings</b> .....	<b>3</b>
1-1 Data Type Description .....	4
1-2 Signal Type of Modbus PLC .....	4
1-3 Cable Diagram.....	5
<b>Chapter 2 Editing Configuration</b> .....	<b>7</b>
2-1 Modbus Communication Setting.....	8
2-1-1 Device Define.....	9
2-1-2 Folder Define.....	12
2-1-3 Data Item Define .....	14

# Chapter 1 Hardware Settings

---

## 1-1 Data Type Description

---

OPC Server Data Type	Description
<b>Bool</b>	Single bit
<b>Int</b>	Signed 16 bit value
<b>Dint</b>	Signed 32 bit value
<b>UINT</b>	Unsigned 16 bit value
<b>UDINT</b>	Unsigned 32 bit value
<b>Real</b>	32 bit floating point value
<b>String</b>	String type with "Null" character at the end
<b>BCD</b>	2 Byte (BCD)
<b>LBCD</b>	4 Byte (BCD)

## 1-2 Signal Type of Modbus PLC

---

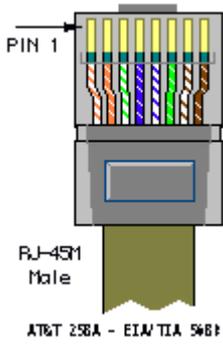
Signal types	Meaning	Access	Data type	Address
<b>DI</b>	Input Coil	Read Only	Bool	Start from 10001
<b>DO</b>	Output Coil	Read / Write	Bool	Start from 00001
<b>AI</b>	Input Register	Read Only	Bool , Real , Int , Uint , Dint , Udint , String , BCD , LBCD	Start from 30001
<b>AO</b>	Output Register	Read / Write	Bool , Real , Int , Uint , Dint , Udint , String , BCD , LBCD	Start from 40001

### 1-3 Cable Diagram

The devices which use Modbus TCP communication protocol based on Ethernet platform. General speaking, you can use the cross over Ethernet cable or HUB and easily let the PC connect to the Modbus devices. The hardware cable diagram is as follows:

 Lots of devices use Modbus TCP Protocol, but the cable diagram of each device may differ from another.

Straight Through Cable PLC/HMI	
RJ-45 PIN	RJ-45 PIN
1 Tx+	1 Rc+
2 Tx-	2 Rc-
3 Rc+	3 Tx+
6 Rc-	6 Tx-



RJ-45  
Male  
AT&T 258A - EIA/TIA 568B

Crossover Cable PLC/HMI	
RJ-45 PIN	RJ-45 PIN
1 Rx+	3 Tx+
2 Rc-	6 Tx-
3 Tx+	1 Rc+
6 Tx-	2 Rc-



## **Chapter 2** Editing Configuration

---

## 2-1 Modbus Communication Setting

### Sample

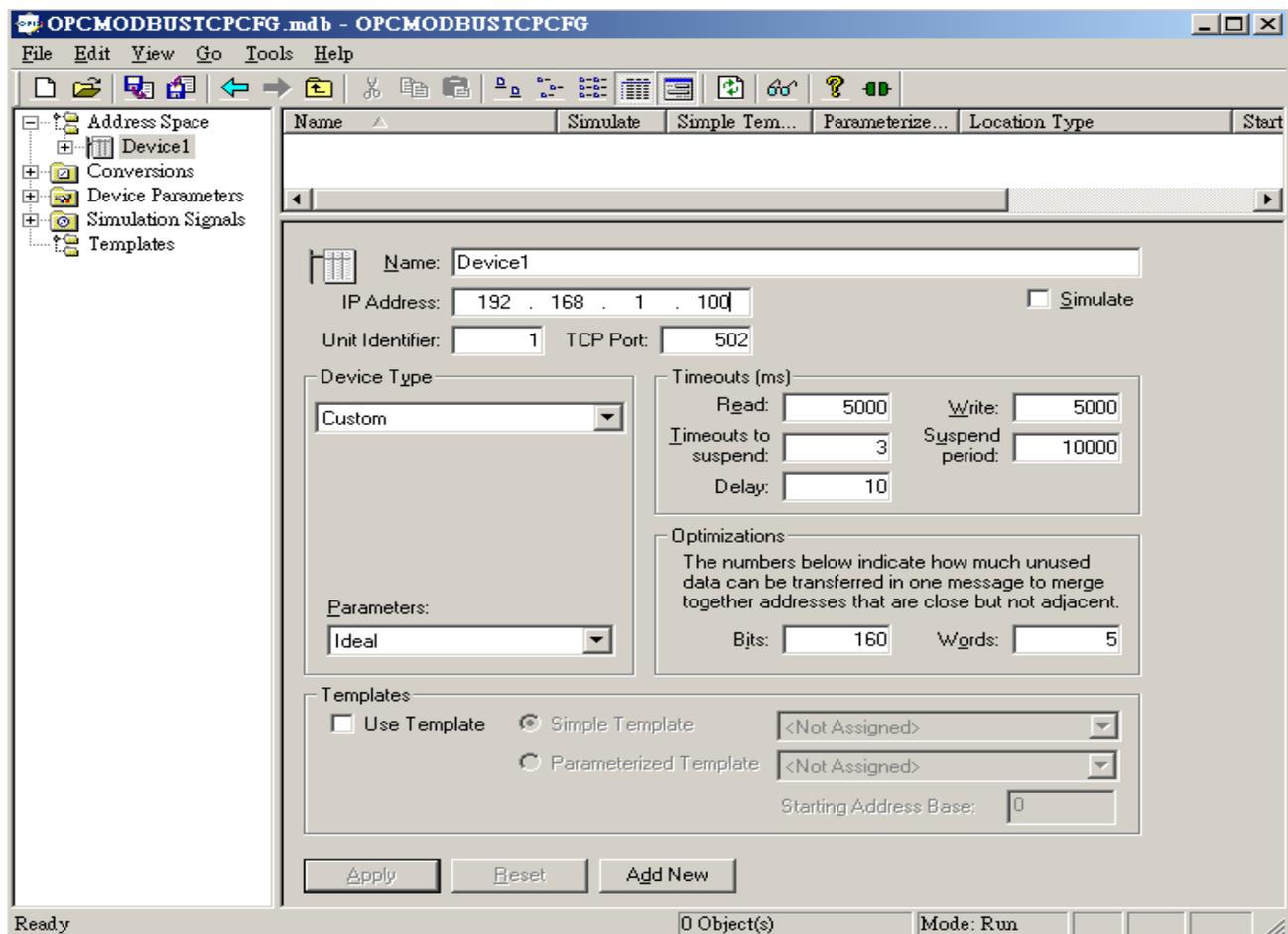
A device using Modbus TCP protocol and stands the slave side.

The Address Space tree control in the Configurator, shown in the figure below, sets the properties and connection parameters for the following items:

- **Devices**
- **Folders**
- **Data Items**

Data items can be organized hierarchically. It is similar to organizing directories and files on your computer's hard disk. The Modbus OPC Server offers several levels of hierarchy. The Configurator module uses the terms "Folder" and "Data Item". A folder can contain additional folders and also data items. The **data items** are always the branches in the tree control hierarchy.

The hierarchical structure of the **folders** and data items helps to organize the devices and tags, as shown below.

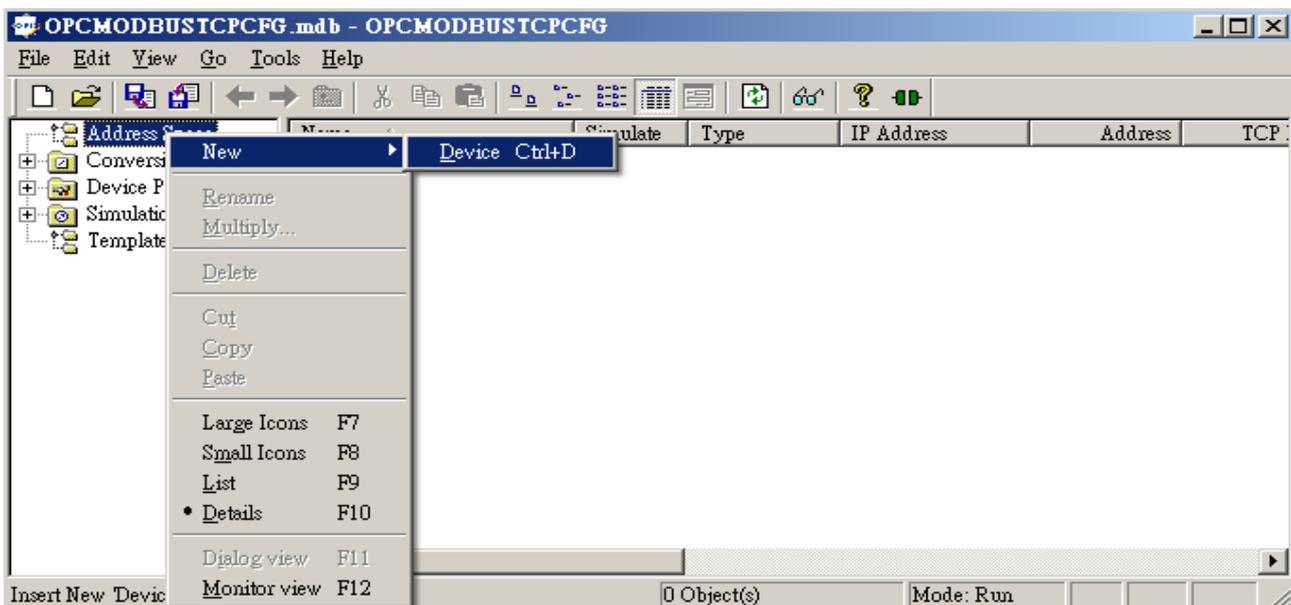
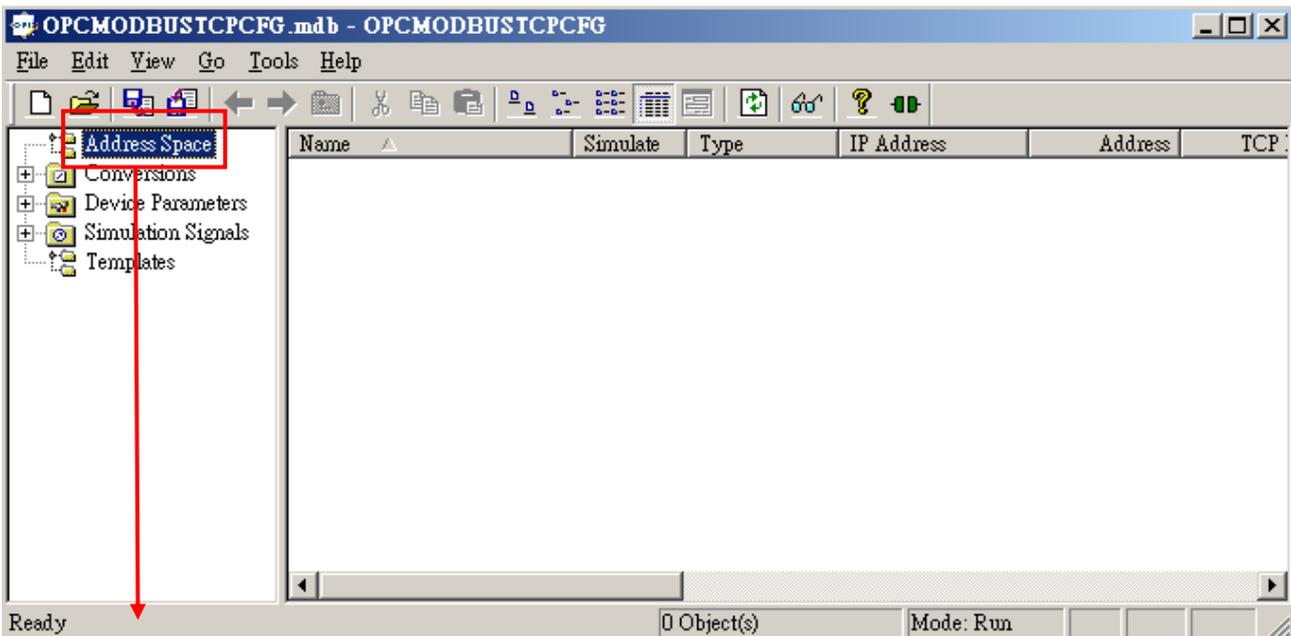


## 2-1-1 Device Define

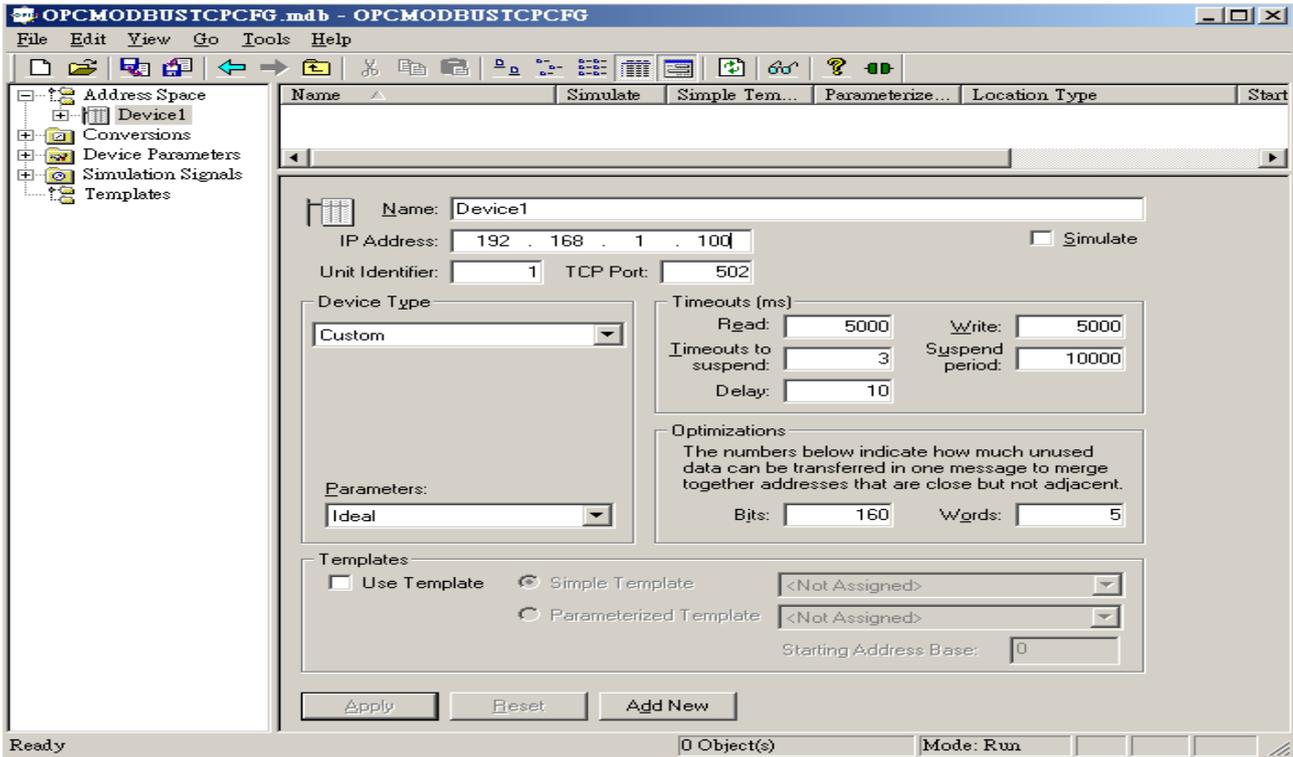
Every **device** is connected to particular port, so it logically creates the second level in the **Address Space** tree. Each device is represented by its symbolic name, and is uniquely identified by the Address value. It is impossible to have two devices with the same address connected to one port.

Setting up a device requires configuration of its unique address, type, timeouts and optimization parameters, as shown in the figure below.

1. Select **Port** → **Edit** → **New** → **Device**
2. You can also press [Ctrl+D] as the shortcut to create a Device.



Device define operation platform



Item	Description
Name:	Specify a name of the new device. The special symbols such as '+', '/', '*', '-' are not allowed.
IP Address: <input type="text" value="192 . 168 . 1 . 100"/>	The number must match the network IP address setting of the device.
TCP Port: <input type="text" value="502"/>	The port connected with the device. (Usually, the number is 502 in Modbus TCP)
Unit Identifier: <input type="text" value="1"/>	The Unit Identifier must match the station number.
Device Type <input type="text" value="Custom"/>	There is a group of six predefined standard device types enhanced with Other (Any) and Custom options. A device with the most limited parameters and the lowest performance is called the Other (Any) alternative.
Timeouts (ms) Read: <input type="text" value="3000"/> Write: <input type="text" value="3000"/> Timeouts to suspend: <input type="text" value="3"/> Suspend period: <input type="text" value="5000"/> Delay: <input type="text" value="10"/>	Timeout parameters (separately for reading and writing data) specify the period of time (in milliseconds) the server will wait for a response from the device: <b>Read/Write:</b> Amount of time (in milliseconds) the OPC server will wait for a response (read/write) from the device. <b>Timeouts to Suspend:</b> The number of

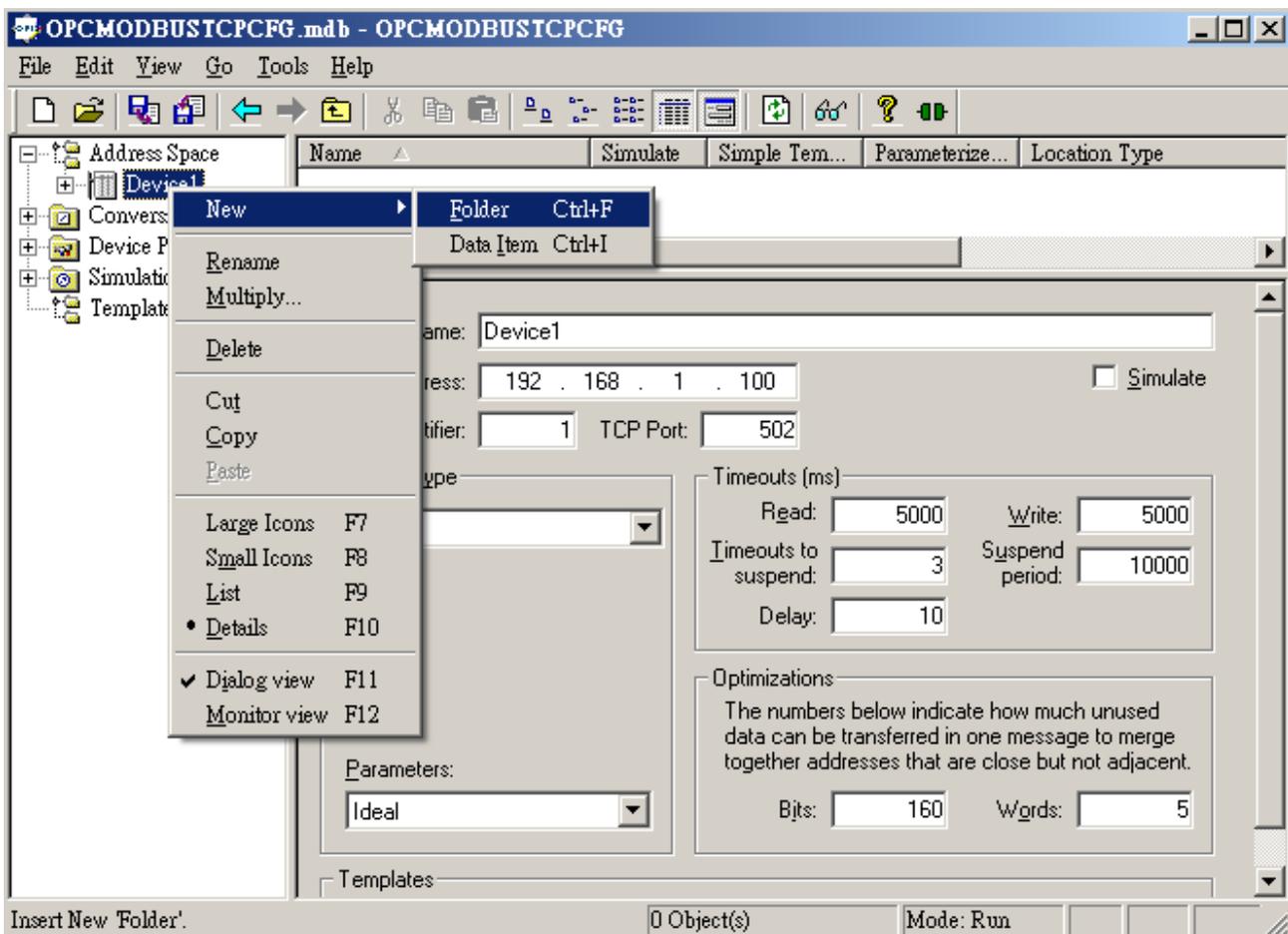
	<p>consecutive read/write attempts that timeout before the OPC server will suspend communication with the device.</p> <p> <b>Suspend Period:</b> Amount of time the OPC server will wait before attempting to reconnect to the device. <b>Note:</b> The Suspend Period setting can be decreased to reconnect faster with a device that has been restarted, but this will cause the OPC server to generate more network traffic while a device is down.</p> <p> <b>Delay:</b> Amount of time (in milliseconds) between read attempts.</p>
<p>Optimizations</p> <p>The numbers below indicate how much unused data can be transferred in one message to merge together addresses that are close but not adjacent.</p> <p>Bits: <input type="text" value="64"/> Words: <input type="text" value="5"/></p>	<p>The server tries to optimize the communication with the devices by requesting as much data as possible in one message. Consecutive registers are merged together into one request for efficiency. The server also can read registers that are not really requested, if this allows it to join two blocks of requested registers. The numbers entered under Optimizations specify the maximum block length of adjacent unused data.</p>
<p>Parameters:</p> <p><input type="text" value="Ideal"/></p>	<p>The Device Parameters directory contains the list of custom device types. Device parameters influence the behavior and performance of the server for the device.</p>
<p><input type="checkbox"/> Simulate</p>	<p>All of the groups and data items under this device are in the simulation mode. Simulation function will simulate the real communication situation between devices and OPC Server.</p>
<p>Templates</p> <p><input type="checkbox"/> Use Template <input checked="" type="radio"/> Simple Template <input type="text" value="&lt;Not Assigned &gt;"/></p> <p><input type="radio"/> Parameterized Template <input type="text" value="&lt;Not Assigned &gt;"/></p> <p>Starting Address Base: <input type="text" value="0"/></p>	<p>Please see "UMHMIOPCEA.pdf".</p>
<p><input type="button" value="Apply"/> <input type="button" value="Reset"/> <input type="button" value="Add New"/></p>	<p><b>Apply:</b> Saves all changes specified in the properties dialog box.</p> <p><b>Reset:</b> Undo the setting configuration</p> <p><b>Add New:</b> Add the new Device</p>

## 2-1-2 Folder Define

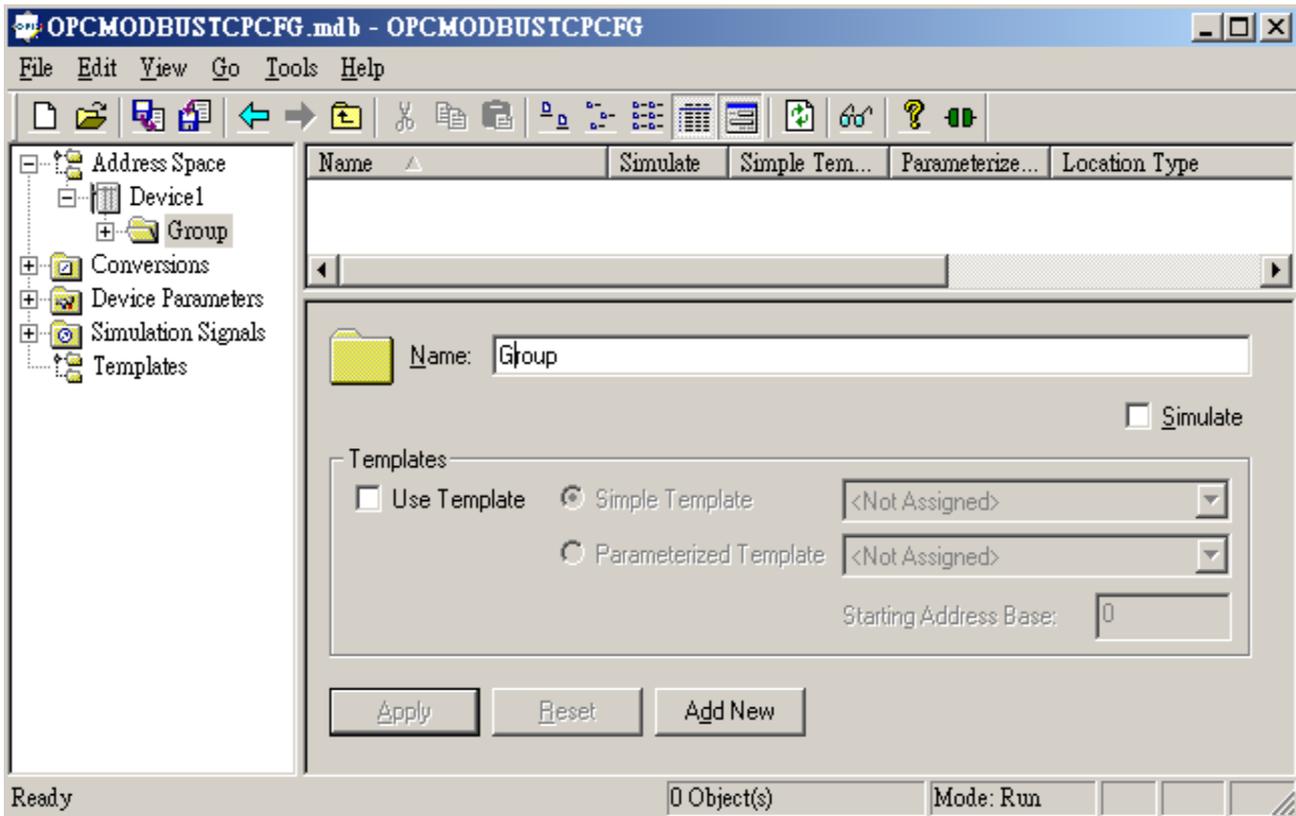
Folders can be used to group items logically. You can configure as many folders as required. Each folder can even have subfolders. The Modbus OPC Server supports up to three folder levels. The use of folders is not required.

If desirable, the configuration could just contain data items without any folders. But most likely this will only be useful if the application does not demand too many persistent tags.

Select [ **Device** ] or [ **Folder** ] on the Tree View. Press the right key of the mouse and select [ **New** ]→ [ **Fold** ]



Folder define Operation Platform



Item	Description
 Name:	Specify a name of the new folder. The special symbols such as '+', '/', '*', '-' are not allowed.
<input type="checkbox"/> Simulate	All of the data items under this group are in the simulation mode. Simulation function will simulate the real communication situation between devices and OPC Server.
Templates <input type="checkbox"/> Use Template <input checked="" type="radio"/> Simple Template    <Not Assigned> <input type="radio"/> Parameterized Template    <Not Assigned> Starting Address Base: 0	Please see "UMHMIOPEA.pdf".
<input type="button" value="Apply"/> <input type="button" value="Reset"/> <input type="button" value="Add New"/>	<b>Apply:</b> Saves all changes specified in the properties dialog box. <b>Reset:</b> Undo the setting configuration <b>Add New:</b> Add the new Folder

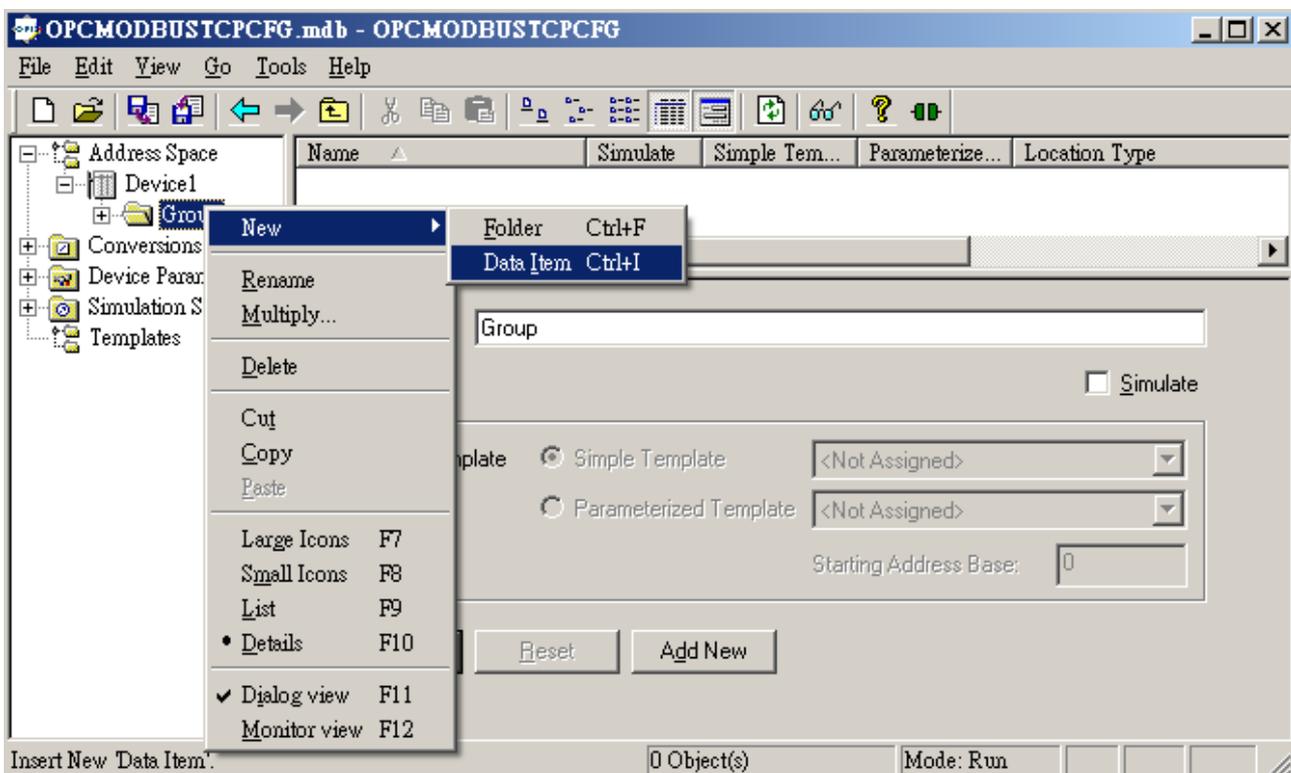
## 2-1-3 Data Item Define

A Data Item represents a register in the device or a range of registers. A symbolic name and description is associated with the data item. An OPC client can obtain the data item description.

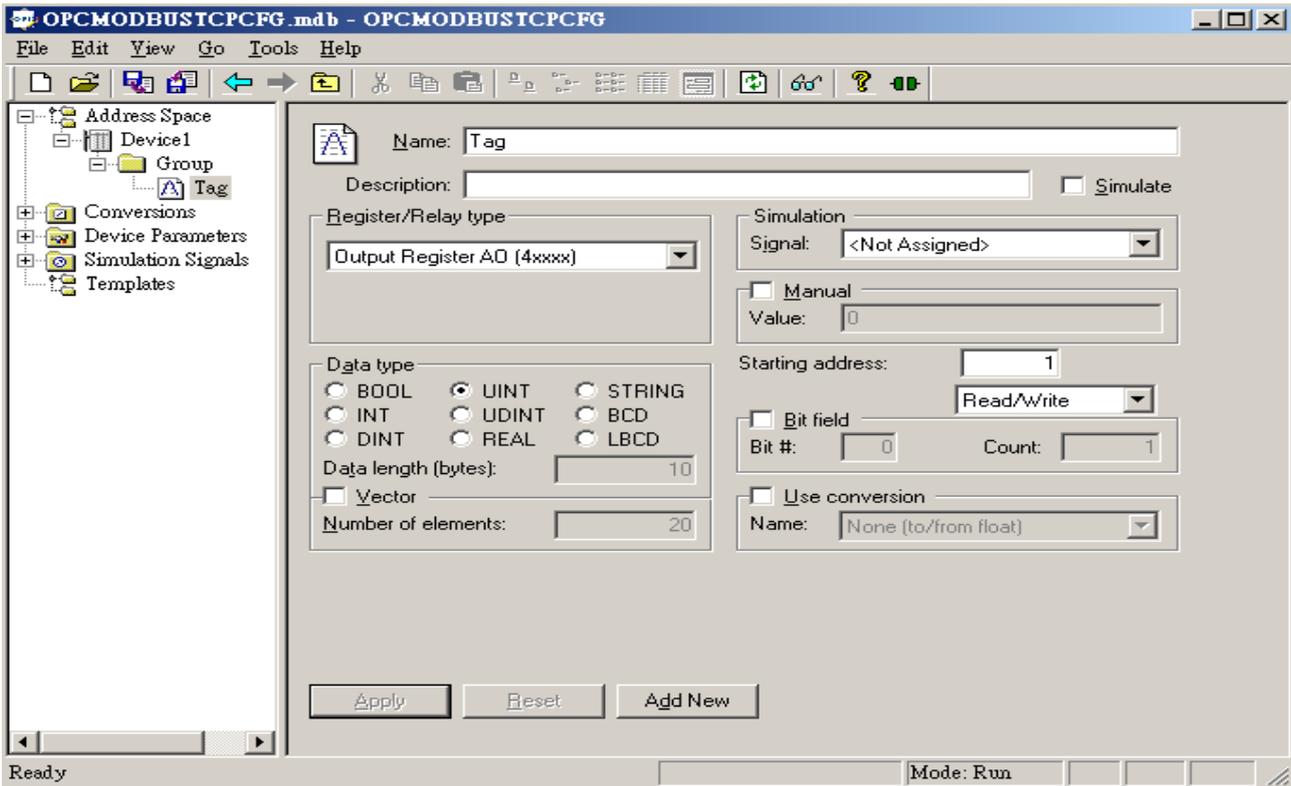
The actual OPC item name (tag) is compounded from the **Address Space** root, the names of the folder and its subfolders, and the name of the data item. Data items can be located in any folder, even in the root of the address space.

**The steps to create the data item (tag) are as below.**

Select [ Device ] or [ Folder ], Press the right button of the mouse, it will show you the Popup menu. Then select **[New] → [Data Item]**



Then define the data Items, the picture is as below :



Item	Description
 Name:	Specify a name of the new data item. The special symbols such as '+', '/', '*', '-' are not allowed.
Description:	Description of Data Item
Register/Relay Type Output Register AO (4xxxx)	Register/ Relay Type: Output Coil, Input Coil, Input Register and Output Register.  Look for more information in Chapter 1
Data Type <input type="radio"/> BOOL <input type="radio"/> UDINT <input type="radio"/> STRING <input checked="" type="radio"/> INT <input type="radio"/> REAL Data length (bytes): 10	Data Type : Output Coil 、 Input Coil → Bool Input Register 、 Output Register → Int 、 UINT String → When selecting String type, you must specify the data length
<input type="checkbox"/> Vector Number of elements: 20	From the starting address to access the data by block, not point to point to access the data. It could speed up the communication speed. You do not have to define a lot of items to get the values. Using [ Vector ] function will let OPC server put the values into the arrays.
Starting address: 1	This value specifies the data item address (register number) in the device data space. Note: The starting address of Modbus OPC Server

	<p>starts with 1. If the starting address of the equipment is 0, then add 1 to be the correct address</p>
<p><input type="checkbox"/> Bit field</p> <p>Bit #: <input type="text" value="0"/> Count: <input type="text" value="1"/></p>	<p>If the data type is UINT, you can choose the number of bits and analyze the value and put into this item. Bit# means the starting Bit No. Count means the constant Bits number.</p>
<p><input type="checkbox"/> Use conversion</p> <p>Name: <input type="text" value="None (to/from float)"/></p>	<p>To get the data value converted according to a prescribed form, choose one of the predefined or user-defined conversions.</p>
<p><input type="checkbox"/> Simulate</p>	<p>Simulation function will simulate the real communication situation between devices and OPC Server. Executing “Monitor View” or press F12 as the shortcut, the “Tag Value” will keep changing for testing the connecting situation.</p>
<p>Simulation</p> <p>Signal: <input type="text" value="&lt;Not Assigned&gt;"/></p>	<p>To test the client functionality, choose a Simulation Signal from the Signal drop-down list and check the Simulate check box. All levels in the Address Space (port, device, folder, data item) support the process of simulation (Simulate check box). The parent list in the tree is superior; it has a higher priority when deciding to simulate the data item or not. In other words, a data item is simulated, if it itself has a simulation selected, or if any of its parents has the Simulate check box checked. (It may be simulated even if its Simulate check box stays unchecked)</p>
<p><input type="checkbox"/> Manual</p> <p>Value: <input type="text" value="0"/></p>	<p>If checked, the data item will offer constant parameter value, because Manual setting is of the highest priority. The changes in the configuration take effect only when the server reloads the configuration (on startup).</p>
<p><input type="button" value="Apply"/> <input type="button" value="Reset"/> <input type="button" value="Add New"/></p>	<p><b>Apply:</b> Saves all changes specified in the properties dialog box.  <b>Reset:</b> Undo the setting configuration  <b>Add New:</b> Add the new Data Item</p>