

Technical Paper

Ethernet Network with OPC Server and Client

Foreword	3
Ethernet Network	4
Ethernet Hardware	4
Device Server.....	4
Router	5
Workgroup Switch	5
OPC Server and OPC Client Software	6
What is OPC?	6
OPC Server.....	6
OPC Client	7
ActiveX Control	7
Device Polling Strategies	7
Compatibility requirements on a TCP/IP LAN	8
Configuration example	8
Network Topology	9
LinkSys Router DHCP Function	10
Configuration of the iServer (Device Server).....	11
Connecting the HygroFlex 2 transmitter to the iServer.....	14
Preparing for a connection over the Internet	15
Communication protocol of the HygroFlex 2	16
Configuring TopServer (OPC Server).....	17
Using Microsoft Excel as an OPC client	38
Embedding OPC Data ActiveX Control in Excel / VBA.....	38
Code specific to the OPC Data ActiveX Control.....	42

Foreword

This technical paper provides some basic information about connecting ROTRONIC products with a serial output to an Ethernet network, and using an OPC server together with an OPC client to retrieve / process the measurement data.

Depending on the application, connecting and using our products in this manner may offer the following benefits:

- Integration of ROTRONIC products in a pre-existing Ethernet network is possible
- Devices that use different communications protocols can coexist on the same network
- A large number of devices can be connected to the network and the limit of 32 instruments applicable to a dedicated serial network using the ROTRONIC HW3 software can be exceeded.
- Large distances between instruments / transmitters are possible (for example by connecting over the Internet)
- A single software application (OPC Server) can be used to write to or read from all the devices on the network.
- Data from all devices on the network can be processed by a single application (OPC Client): data display, data logging, etc.

The paper presents an example based on the ROTRONIC HygroFlex 2 transmitter and on specific third party hardware / software. Compatibility of our products is not necessarily restricted to these third party products and the fact that we used any particular product as an example does not constitute a recommendation or an endorsement. The paper assumes that the reader is reasonably familiar with either Visual Basic or Visual Basic for Applications (VBA). Therefore, detailed "How To" instructions are not provided regarding these programming languages.

Ethernet Network

The term Ethernet refers to the communications protocol, wiring technology and family of local-area network (LAN) products covered by the IEEE 802.3 standard. Several data rates have been defined for operation over optical fiber and twisted-pair cables:

10 Mbps—10Base-T Ethernet (presently, the most commonly used)
100 Mbps—Fast Ethernet
1000 Mbps—Gigabit Ethernet

Other technologies and protocols have been touted as likely replacements, but the market has spoken. Ethernet has survived as the major LAN technology (it is currently used for approximately 85 percent of the world's LAN-connected PCs and workstations) because its protocol has the following characteristics:

- Is easy to understand, implement, manage, and maintain
- Allows low-cost network implementations
- Provides extensive topological flexibility for network installation
- Guarantees successful interconnection and operation of standards-compliant products, regardless of manufacturer

For additional information, you may want to visit any of the following sites:

<http://www.techtutorials.com/Networking/Architectures/index.shtml>
<http://www.lantronix.com/learning/tutorials/index.html>

Ethernet Hardware

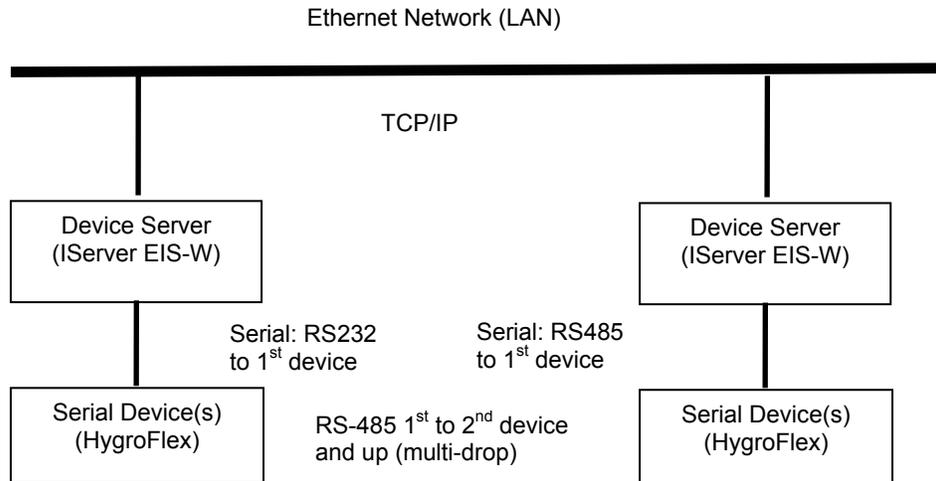
Device Server

A Device Server (or terminal server) allows the connection of one or several serial devices (RS-232 or RS-485) to an Ethernet network using TCP/IP or another protocol. The Device Server has the ability to package serial data into TCP/IP and vice-versa (a process called Ethernet Encapsulation). Each Device Server on a network must be given a unique IP address that is compatible with the network (see compatibility requirements on a TCP/IP network) and has to be configured to accept requests on a specific port (see example provided in this paper). The Device Server can be either an external piece of hardware or an internal module already integrated by the manufacturer of the serial device.

Many examples of such products can be found on the Internet. Here are a few links:

<http://www.newportus.com/Products/Network/index.htm>
<http://www.lantronix.com/products/ds/index.html>
<http://www.bb-elec.com/product.asp?sku=ESP901&src=home>

Note: throughout this technical paper, we are using the iServer External Microserver model EIS-W from Newport Electronics as an example.



Each individual ROTRONIC instrument / transmitter can be connected to a LAN or to the Internet by means of a Device Server. If so desired, it is possible to use small local networks of ROTRONIC devices, each with one Device Server and with dedicated wiring. In that case, each local network is limited to 32 devices but there is no limit to the number of networks. Local networks can be established using either RS-232 or RS-485 to connect the first instrument.

Router

Routers are used to provide an interface between multiple networks or between the Internet and a local network or LAN. One or more routers can be used to increase the maximum number of IP addresses on a LAN. For accessing the Internet, the router is typically used together with a cable or DSL modem (permanent Internet connection). Routers manage traffic and often provide a protection (firewall) against unauthorized access from the outside. Some routers can act as a DHCP server and can automatically assign an IP address to the devices connected to the network.

Typical router functions such as Gateway and Port Forwarding make it possible to send outgoing data to the Internet and to direct incoming data (from the Internet) to a specific IP address and port on a LAN (corresponding to a specific Device Server).

The following is the web site address of a popular manufacturer of routers both for home and office applications: <http://www.linksys.com/products/>

Workgroup Switch

A workgroup switch (or hub) can be used to provide additional physical ports (or Ethernet connections) at a specific location. Use of a switch does not increase the number of available IP addresses on a LAN.

OPC Server and OPC Client Software

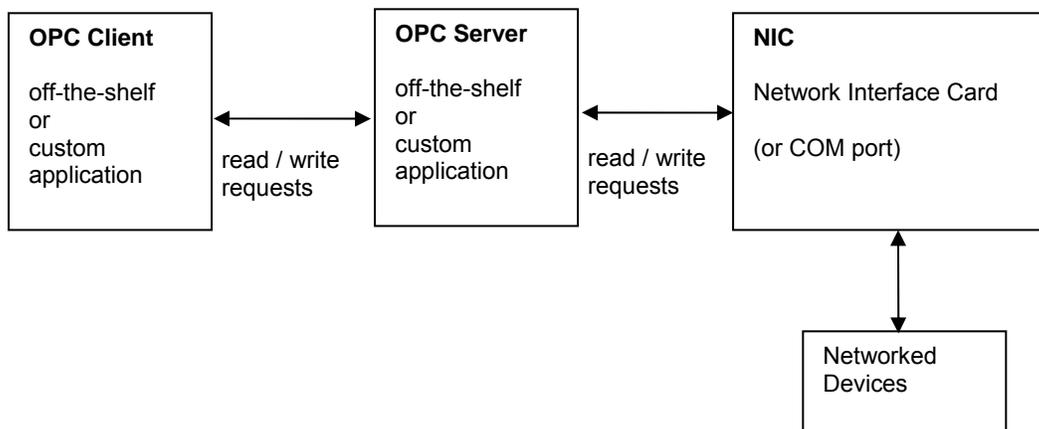
What is OPC?

OPC stands for OLE (Object Linking Embedding) for process control. OPC is a set of written rules and procedures for the way multiple software programs ("applications") can talk to each other. The OPC specification was created by users and software developers in the manufacturing / process control industry to serve the needs of the manufacturing automation / process control industry specifically. The specification is managed by volunteer effort and administered by the independent [OPC Foundation](#).

OPC came about because application software developers and users saw the need to begin to standardize how their applications worked together to reduce integration and total life cycle costs for software used in the manufacturing automation/process control industry.

OPC has become a standard common to many industrial process applications. In principle, any application written in Visual Basic (and a number of other programming languages) can be enabled to use OPC. In addition to allowing communication between different makes of software, a major benefit of OPC is the ability to integrate into a single application data from diverse sources.

Using OPC requires two software components: the OPC Server and the OPC Client.



OPC Server

The OPC server takes care of the back and forth communication with the devices on a network. Typically an OPC Server can connect to several network interface cards (NIC), and can in this manner handle networks that use different wiring technologies.

The communications protocol used by manufacturers of hardware tends to be unique to each manufacturer. Off-the-shelf OPC servers take care of this by offering a large choice of drivers matching the communications protocol of the most popular manufacturers of hardware. Some OPC servers also offer a user configurable driver handles to enable communication with devices for which there is no ready-made driver.

In this technical paper, we have used TOPServer from Software Toolbox Inc as an example of an OPC server. Additional information and an evaluation download are available at the following site: <http://www.toolboxopc.com>

In order to communicate with three ROTRONIC HygroFlex 2 transmitter connected to an Ethernet network, we have used the TOPServer User Configurable Driver (UCON). This driver is completely integrated into TOPServer.

OPC Client

The OPC client is where the data ends up for processing, visualization and / or storage. In the US, examples of popular off-the-shelf OPC clients are SuiteLink, Intellution NIO, Wonderware InTouch and Wonderware Factory Suite.

ActiveX Control

When the intended client application does not support OPC, it may be possible to enable the use of OPC by embedding an ActiveX control in the application. Typically this is the case of applications developed with Visual Basic. ActiveX controls are typically user interface components and do not run on their own like an EXE application.

OPC Servers and ActiveX controls are both based on the Microsoft Component Object Model (COM).

For example, the OPC Data ActiveX control from Software Toolbox Inc. can be used to add OPC client functionality to an Excel spreadsheet programmed with Visual Basic (VBA). For more information, use the following link: [OPC Data ActiveX control](#)

Device Polling Strategies

Polling the devices on the network (or scanning the data) can be done either from the OPC client or from the OPC server. Some OPC servers can also handle unsolicited communications from devices on the network.

Typically, device read requests are initiated by the OPC client. There are two types or Reads: synchronous reads and subscription reads.

Synchronous Read: the OPC client directs the OPC server to go read data from the device (i.e. the PLC). The OPC client will wait for the OPC server to return with the data. Unless it is running in a multi-threaded fashion (which many clients do, including the OPC ActiveX data control), the OPC client can't do anything else while waiting. Even if the OPC client is multi-threaded, it can still be wasteful of computing resources to have the client sit and wait. Generally, it is a bad idea to use synchronous reads in conjunction with a Visual Basic timer. Synchronous device reads may be used if (a) you only want to read an item one time based on operator input (i.e. they click a button) or some other event or (b) you need to know the value now and do not want to continue to do anything else until you get the result.

Subscription Read: the OPC client tells the OPC server the data items that it wants updates for and how often it wants the data to be checked. At that time the client returns to what it is doing. Whenever the OPC Server detects that a value has changed, it automatically makes a "callback" to the OPC client and delivers the data. The key is that the OPC client does not have to waste time polling the OPC server for changed data and waiting for data to come back from a request. The OPC server can manage it's priorities to provide optimal performance versus being constantly told what to do by the client, which is what happens with synchronous reads.

Compatibility requirements on a TCP/IP LAN

- IP address (Network ID and Host ID)

An IP address consists of 4 numbers (1 to 3 digits each) separated with dots. On any network, each IP address should be unique.

IP addresses are classified as follows:

Class A: **1**.x.x.x to **127**.x.x.x
Class B: **128.0**.x.x to **191.255**.x.x
Class C: 192.0.0.x to 223.255.255.x

In the above, x is a number between 0 and 255.

Within an IP address, the network ID is defined depending on the class of the IP address and is indicated in bold in the above table. The remainder of the address is the host ID.

For a class C address, the first 3 numbers define the network ID and therefore, the maximum number of hosts (or devices on the network) is limited to 256.

In order for two devices in a network to communicate together, their IP address must contain the same network ID.

- Gateway Address

This is the IP address of the device used to access other networks (in this case the router).

- Subnet Mask

The subnet mask is used to differentiate the network ID portion of an IP address from the host ID.

Class A: **255**.0.0.0
Class B: **255.255**.0.0
Class C: **255.255.255**.0

Note: if more network ID's are required by the network, the last number of the subnet mask can be used, but this reduces the maximum number of hosts available for each network ID.

In order for two devices in a network to communicate together, their subnet mask must be the same.

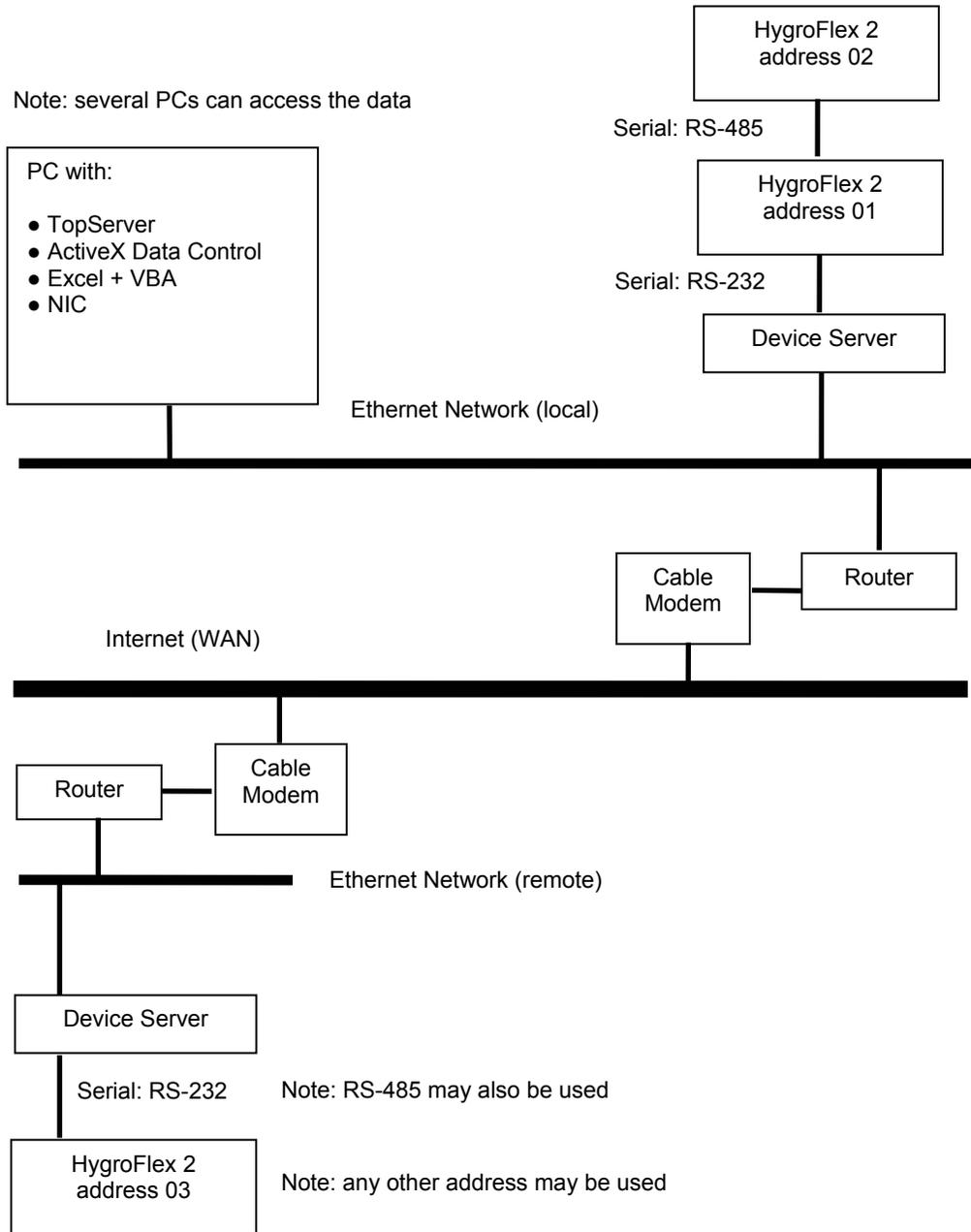
Configuration example

This example is based on the following hardware and software:

Device Server	: iServer EIS-W from Newport Electronics, Inc.
Router	: BEFSR81 from LinkSys (Cisco Systems, Inc.)
OPC Server	: TopServer from Software Toolbox, Inc.
OPC ActiveX Control	: OPC Data ActiveX Control from Software Toolbox, Inc.
OPC Client	: Microsoft Excel

Using other hardware and software would require similar steps.

Network Topology



Note: in this example, both LANs (remote and local) use the same the subnet mask 255.255.255.0. In order to be compatible, all devices connected to these two LANs (Routers, Device Servers, PCs) must share this subnet mask.

LinkSys Router DHCP Function

The LinkSys router features a web server (to serve web pages) and can be accessed / configured with Microsoft Internet Explorer.

The screenshot shows the LinkSys router's DHCP configuration interface. At the top, there is a navigation bar with buttons for 'Setup', 'Password', 'Status', 'DHCP', 'Log', 'Help', and 'Advanced'. The 'DHCP' button is highlighted. Below the navigation bar, there is a section titled 'DHCP' with a sub-header 'DHCP Server:'. The 'DHCP Server' is currently set to 'Enable' (indicated by a selected radio button). Below this, there are three input fields: 'Starting IP Address' with the value '192.168.57.1', 'Number of DHCP Users' with the value '50', and a 'DHCP Clients Table' button. At the bottom of the configuration area, there are three buttons: 'Apply', 'Cancel', and 'Help'.

The range of dynamic IP addresses

The range of dynamic IP addresses that can be assigned to other devices by the LinkSys Router can be set within a low and a high limit. In this example, we are leaving some room for static IP addresses, because the iServer will eventually be assigned a static IP address.

In this example, the available static IP addresses are:

192.168.57.0
192.168.57.52 to 192.168.57.89
192.168.57.91 to 192.168.57.255

In this example, the address 192.168.57.90 is used by the router itself on the LAN side and should not be used by any other device.

Configuration of the iServer (Device Server)

In order to ensure compatibility of the iServer with the other devices on our LAN, we have used a LinkSys router to perform the function of a DHCP server. DHCP stands for Dynamic Host Configuration Protocol. When DHCP is enabled, any new device on the network is automatically assigned a dynamic IP address, a Gateway address and a Subnet Mask. This feature saves a lot of work when devices are frequently attached to or removed from the LAN (laptops etc.).

- Enable DHCP on the iServer

The iServer has an external DIP switch which is used to enable or disable DHCP (Dynamic Host Configuration Protocol). Assuming that a DHCP server exists on the LAN (for example: the LinkSys Router), set the DIP switch so as to enable DHCP.

- Initial IP address of the iServer (dynamic)

Upon connecting the iServer to the LAN (with DHCP enabled), the LinkSys Router automatically gives the iServer a dynamic address, a Gateway address and a subnet mask. The router DHCP Clients Table is used to find out the new IP address of the iServer (in the table, look for the host name and MAC address of the iServer).

Power off the iServer and set the dip switch on the iServer back to static IP address (DHCP disabled). The reason for this is that we will give a new (static) IP address to the iServer.

- Final IP address of the iServer (static)

The iServer features a web server and can be accessed / configured with Microsoft Internet Explorer. The iServer can also be configured from its serial port with an application such as Windows HyperTerminal. In this example, we are using Internet Explorer.

Knowing the initial IP address (dynamic) of the iServer, we can access the iServer configuration web page with Internet Explorer, from any PC on the LAN.

The iServer is protected both by a standard user password and by an administrator's password:



The login screen features the LinkSys logo (a vertical bar with red, yellow, and green segments and a red dot above it) to the left of the word "Login" in a bold, black, sans-serif font. Below the logo and text, there is a text label "Password" followed by a rectangular input field. To the right of the input field is a rectangular button with the text "Ok". Below the input field and button, the text "Main Menu" is displayed in a blue, underlined, sans-serif font.

Enter 12345678 as the standard user password.



The admin screen features the LinkSys logo (a vertical bar with red, yellow, and green segments and a red dot above it) to the left of the word "Admin" in a bold, black, sans-serif font. Below the logo and text, there is a text label "Password" followed by a rectangular input field. To the right of the input field is a rectangular button with the text "Ok". Below the input field and button, the text "Main Menu" is displayed in a purple, underlined, sans-serif font.

Enter 00000000 as the administrator's password.

The following page appears after entering the administrator's password.



ACCESS CONTROL

Login Password	<input type="text" value="12345678"/>
Admin Password	<input type="text" value="00000000"/>
Web Server	<input type="text" value="enable"/> <input type="button" value="v"/>
Host Name	<input type="text" value="eis2187"/>
MAC Address	<input type="text" value="00:03:34:00:21:87"/>
IP Address	<input type="text" value="192.168.57.200"/>
Gateway Address	<input type="text" value="192.168.57.90"/>
Subnet Mask	<input type="text" value="255.255.255.0"/>

[Main Menu](#)

In this example, the iServer is given the static IP address: 192.168.57.200. Both the gateway address and subnet mask are left to the values assigned by the LinkSys Router.

Host Name: if so desired, it is possible to register the iServer host name (eis2187) on the PC. To do this, both the host name and the IP address of the iServer should be entered in the following file:

C:\Windows\System32\Drivers\Etc\HOSTS (the file HOSTS does not have any extension). This file can be opened with Notepad. Instructions are provided within the file.

After registering the Host Name, the iServer can be accessed as eis2187 from Internet Explorer or pinged as eis2187.

- Serial Communication and other Settings

Go to Configuration web page and use the following settings:

 **CONFIGURATION****Serial Communication**

Baud Rate Data Bits Parity Stop Bits

Flow Control Transceiver Modbus/TCP

End Character(Hex) Forward End Char Timeout msec

Serial Port Password

The serial communication settings are in accordance with the requirements of the HygroFlex. Note the following settings: (a) End Character set to 00 (HEX), (b) forward End Character set to disabled and (c) time-out set to zero.

Terminal Server

TCP/UDP Server Type Number of Connections Local Port

Connection Control Connection Timeout X 10 msec

Remote Access(Tunneling)

Remote Access Remote IP Address Remote Port

[Main Menu](#)

Terminal Server should be enabled as shown above. The number of connections (sockets) should be set to at least 1.

Important: be sure to write down both the IP address and the local port number of the i-Server (2060 in this example). This information will be useful later on.

Local port may be set to any value, except for values used by other services such as 21 (ftp), 25 (SMTP), 80 (http), 110 (pop3), etc. Do not use port 2002 as this is used for configuring the iServer itself.

If you plan on communicating over the Internet, a unique port number should be assigned to each iServer connected to the same LAN. The value assigned to Local port will be used by a Router with a registered public IP address to direct data requests to the proper iServer (non-registered private address).

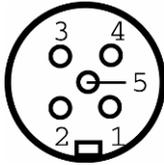
Remote Access (Tunneling) is disabled. This is used only if two iServers are communicating one with the other.

Connecting the HygroFlex 2 transmitter to the iServer

Note: throughout this technical paper, we have used the ROTRONIC HygroFlex 2 transmitter as an example of a serial product. For connection details regarding other ROTRONIC serial products please consult the appropriate instruction manual.

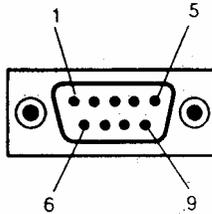
In principle, it is possible to use either RS-232 or RS-485 to connect the first HygroFlex transmitter to the Device Server. When using a relatively fast data refresh rate (2 seconds or less), we have had more reliable results when the first HygroFlex transmitter was connected to the Device Server by means of RS-232.

Additional transmitters are connected to the first and then to one another in a multi-drop arrangement using RS-485. Please note that depending on the type of serial connection used for the transmitter that is directly connected to the Device Server, there is a small difference in the syntax of the data request to be sent to the HygroFlex transmitters,



HygroFlex 2 RS232 / RS485 connector
(5-pin shown from contact side - female)

Pin #	Function	Pos. on Terminal Block (PCB)
1	TXD	5
2	RXD	4
3	GND	3
4	RI +	2
5	RI -	1



RS232 or RS-485 connection (iServer)

Pin #	Function (RS-232)	Function (RS-485)
2	TX	RI +
3	RX	RI -
5	GND	Shield (optional)

Note: the iServer can be configured for either RS-232 or RS-485

IMPORTANT: be sure to verify that there is no error in the wiring between the HygroFlex 2 transmitter and the Device Server. Doing this may save a lot of trouble shooting time later.

Preparing for a connection over the Internet

Using the port forwarding capability of a router provides the means of establishing a communication over the Internet with the Device Server connected to the remote LAN.

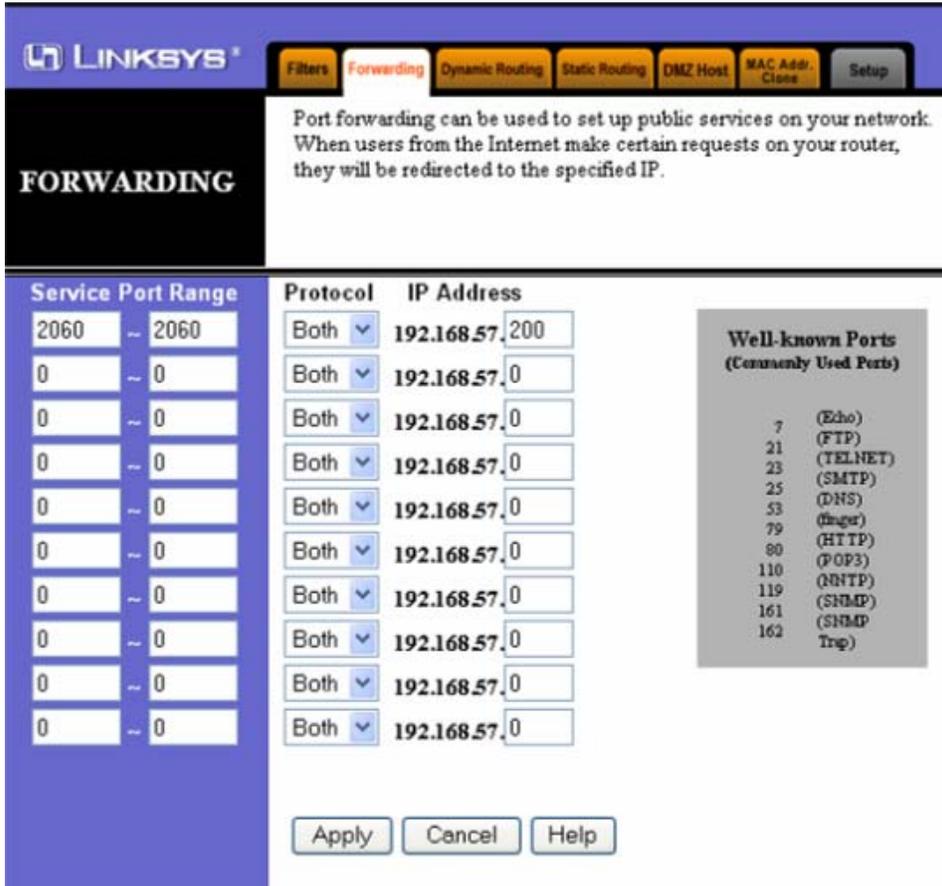
Using the Internet requires a **registered public IP address**. Preferably, this should be a static address and getting such an address can be arranged for a fee with the ISP (Internet Service Provider) at the remote location. For test purposes, we have used the dynamic IP address assigned by the ISP to the remote LinkSys router. This address (WAN) was read from the router: 24.48.19.167.

The screenshot shows the LinkSys router's Status page. At the top, there is a navigation bar with buttons for Setup, Password, Status (highlighted), DHCP, Log, Help, and an Advanced button. Below the navigation bar, a black box contains the word "STATUS" in white. To the right of this box, a message states: "This screen displays the router's current status and settings. This information is read-only." Below this, the status information is organized into sections: Host Name, Firmware Version, Login, LAN, and WAN. Each section lists various parameters and their values. At the bottom, there are buttons for DHCP Release, DHCP Renew, DHCP Clients Table, and Help.

Section	Parameter	Value
Host Name:		
Firmware Version:		2.37, Feb 11 2001
Login:		Disable
LAN:	(MAC Address: 00-04-5A-22-21-B5)	
	IP Address:	192.168.57.90
	Subnet Mask:	255.255.255.0
	DHCP server:	Enabled
WAN:	(MAC Address: 00-04-5A-22-21-B6)	
	IP Address:	24.48.19.167
	Subnet Mask:	255.255.240.0
	Default Gateway:	24.47.16.1
	DNS:	167.206.3.203 167.206.3.137 167.206.3.204
	DHCP Remaining Time:	2 days 6:11:21

Buttons: DHCP Release, DHCP Renew, DHCP Clients Table, Help

Next the router was configured to forward to the LAN IP address 192.168.57.200 (iServer) any outside request directed to port 2060 on the router. Note that this should be the same port that assigned to the iServer, under Terminal Server.



Communication protocol of the HygroFlex 2

Details about the structure of the data request string and response string can be found in each ROTRONIC product instruction manual.

Regarding the data request string, it is important to note the following:

- If RS-232 is used to connect the first ROTONIC instrument or transmitter of a multi-drop to the Ethernet Server (iServer), you should use the vertical bar | (ASCII decimal code 124) as the first character of the data request string just before the left curly bracket. This should be used for all instruments or transmitters in the multi-drop. Example: `{|m01RDD0;}`
- If RS-485 is used to connect the first ROTONIC instrument or transmitter of a multi-drop to the Ethernet Server (iServer), do not use the vertical bar | to begin the data request string. Example: `{m01RDD0;}`

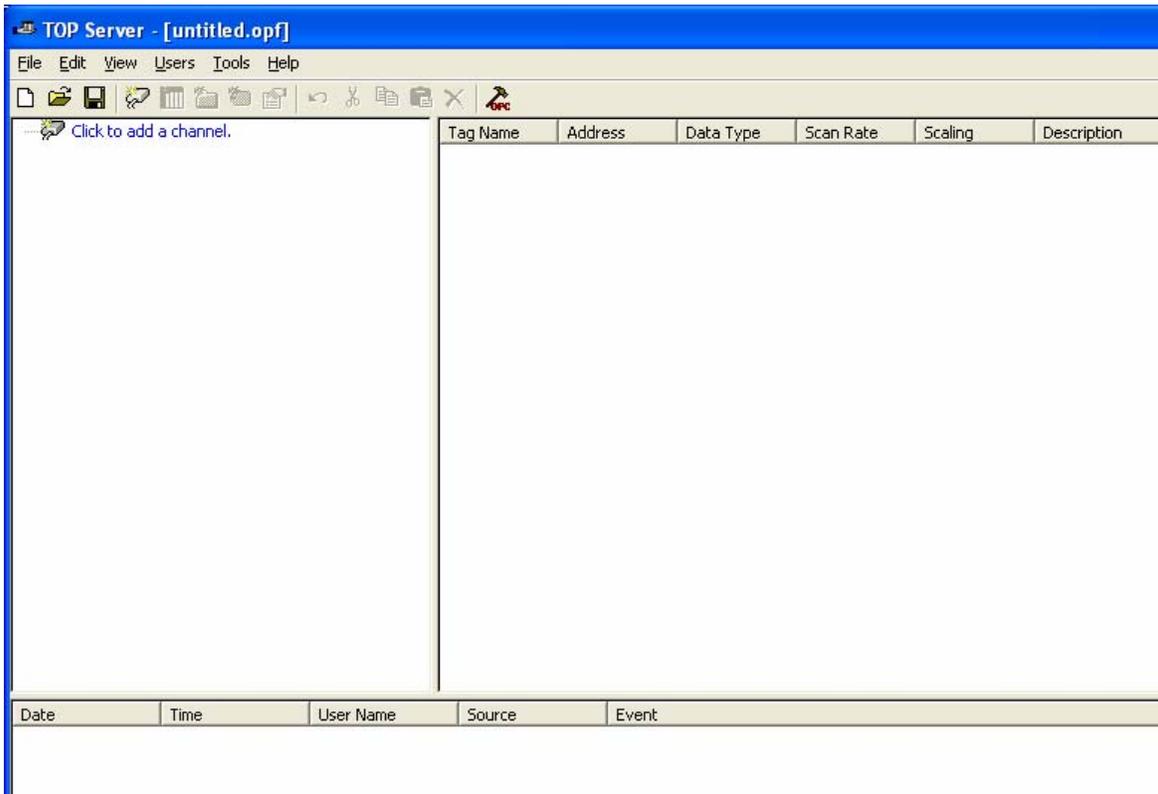
See pages 30 and 31 for a detailed example.

Configuring TopServer (OPC Server)

IMPORTANT: when installing TopServer, be sure to select **User Configurable Driver** from the list of available drivers.

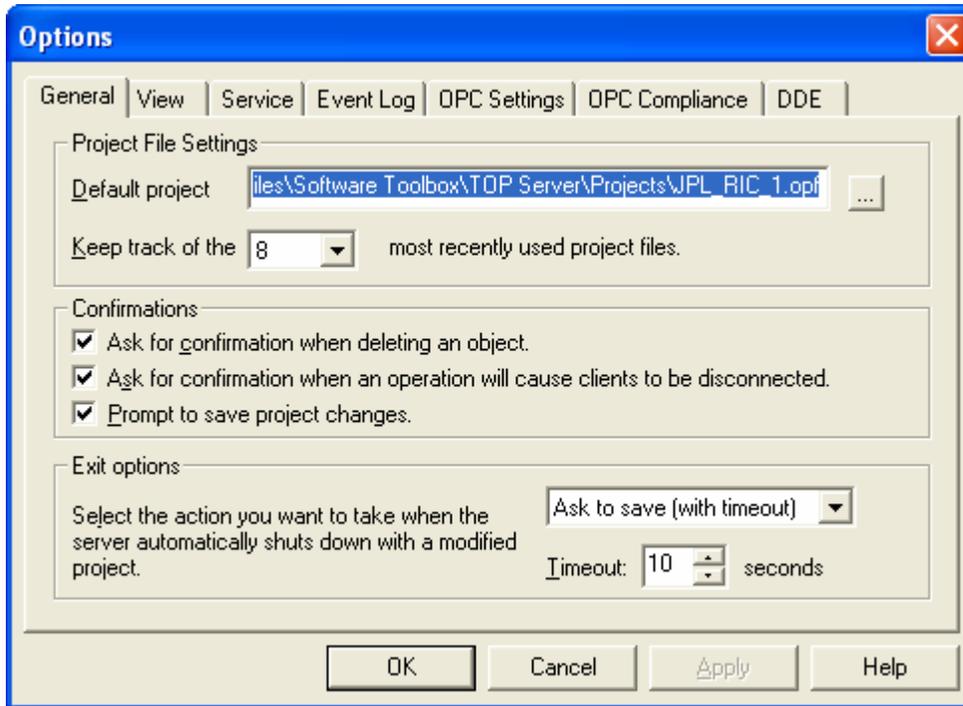
1) Set the TopServer Options

Start TopServer and select New (Project) from the File Menu. The TopServer Main screen has 3 panes. The left pane displays Channels, Devices and Tags. The right pane displays Tags in more detail. The bottom pane displays the main events logged by TopServer.

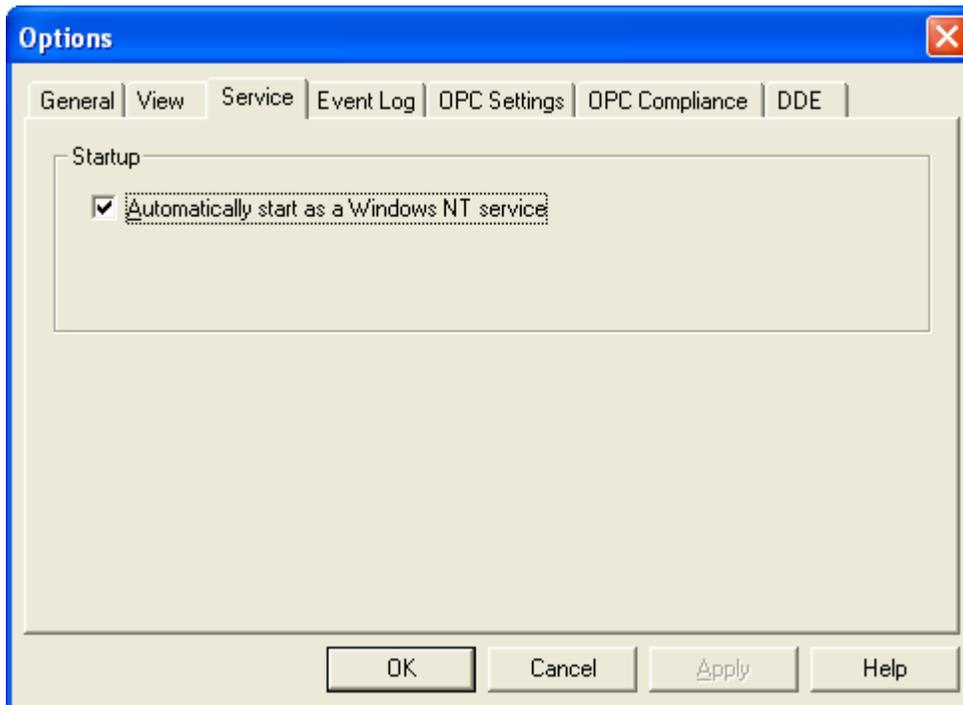


Click on Tools, Options to display the following box:

General Tab: this will be used when the project is completed and saved to disk. At this time, you may want to set the default project to be your project.

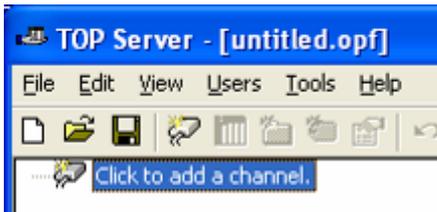


Service Tab: enable TopServer to automatically start as a Service. This will cause TopServer to appear as a small icon on the right hand side of the task bar.



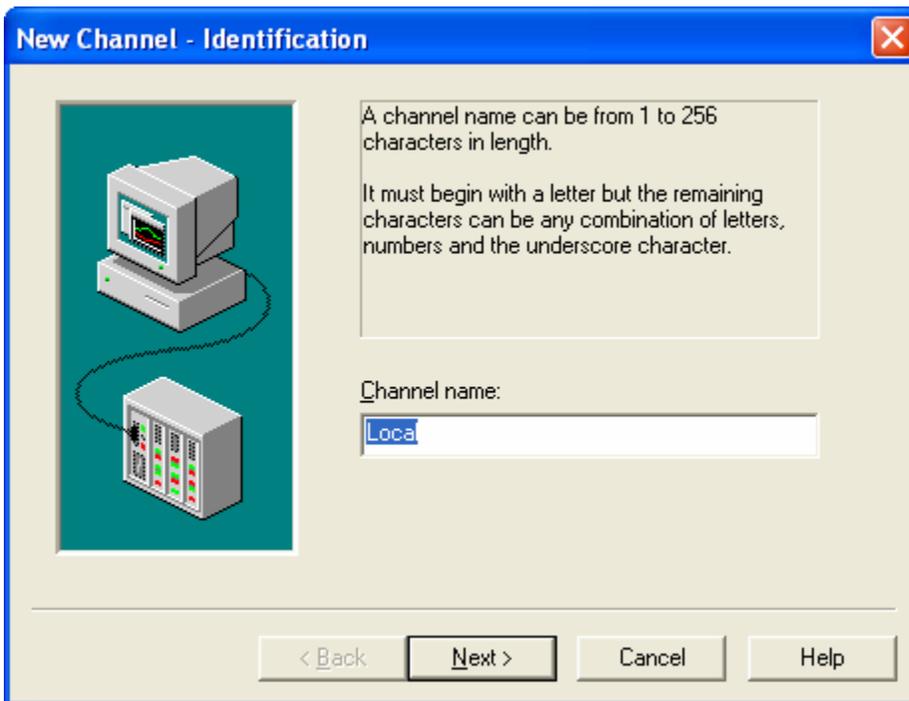
In this example, the other tabs can be ignored.

2) Create a Channel



Left click with the mouse to open the Channel dialog box:

In this example, we use Local as the name of the first channel



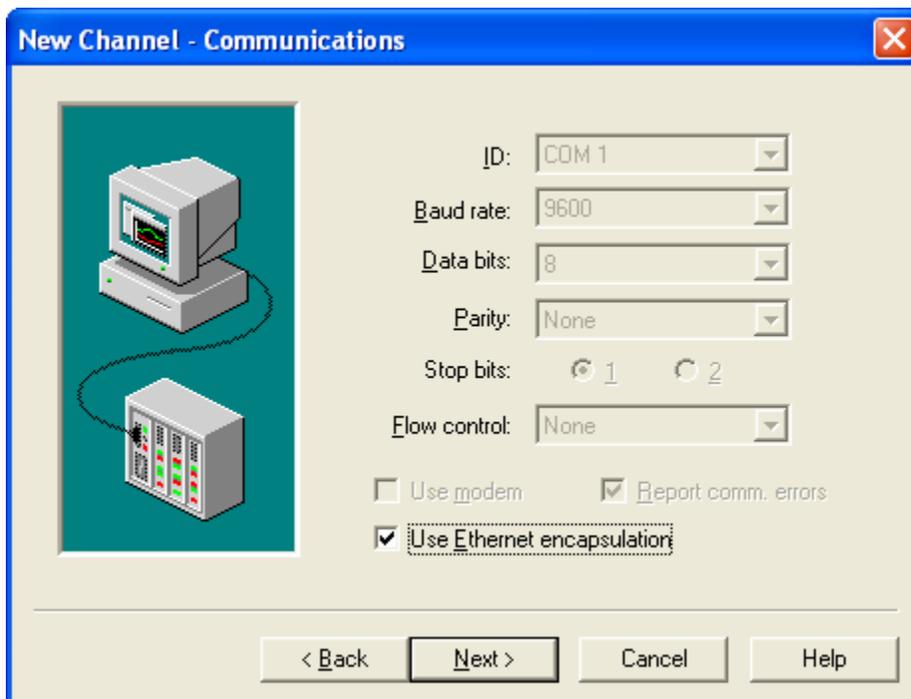
Click on next.

Select User Configurable Driver and enable diagnostics.



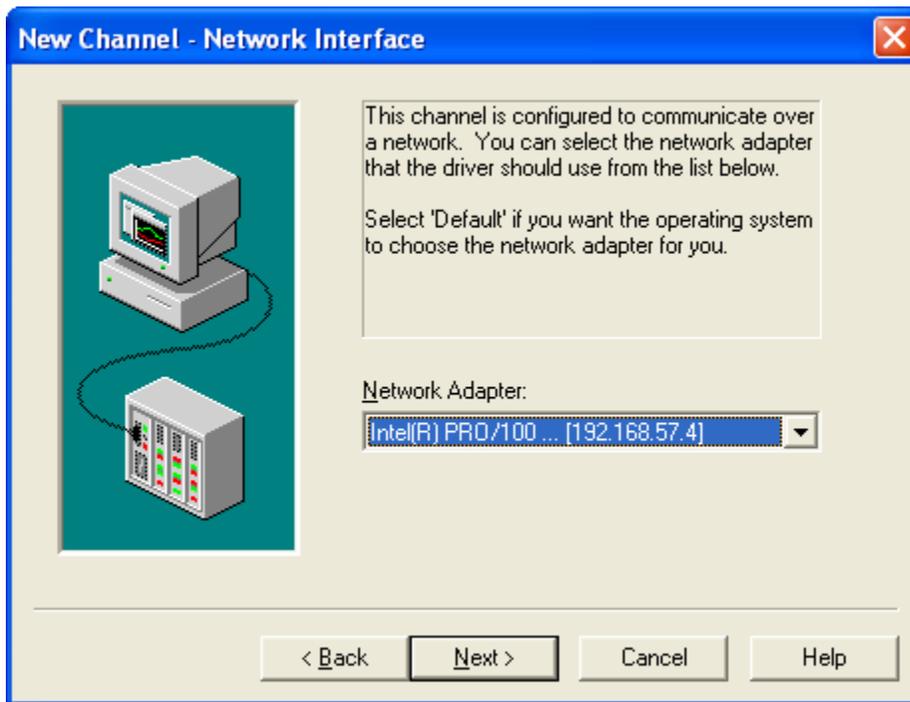
Click on Next.

Because the HygroFlex transmitters are connected to the LAN by means of an Ethernet server (Newport iServer), select Use Ethernet Encapsulation.



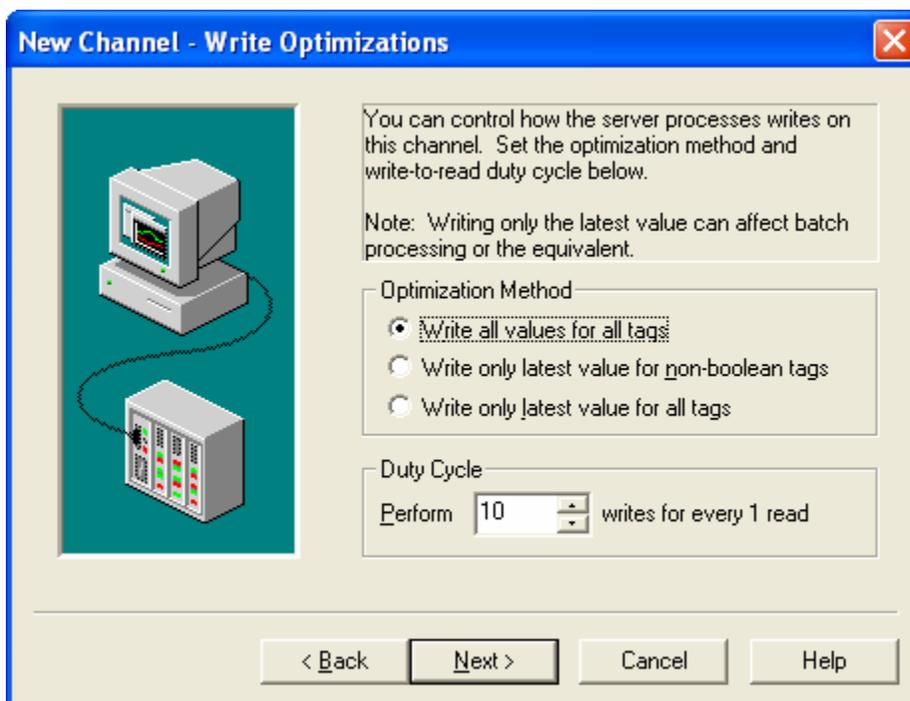
Click on Next.

Select the Network Adapter corresponding to the LAN to which the HygroFlex transmitters are connected (you may have more than one LAN and adapter).



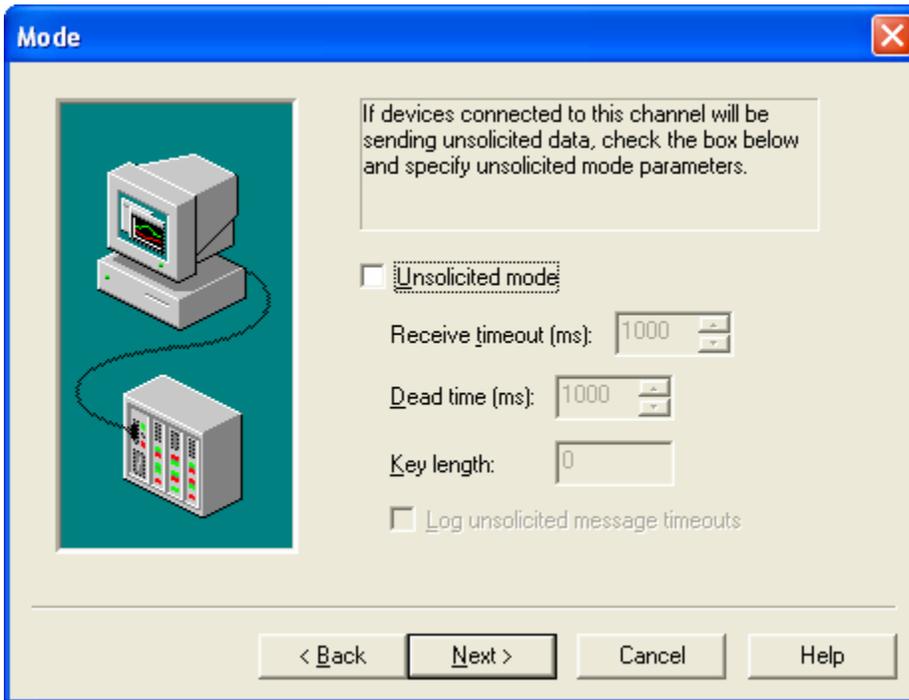
Click on Next.

In the following box, leave the TopServer default settings

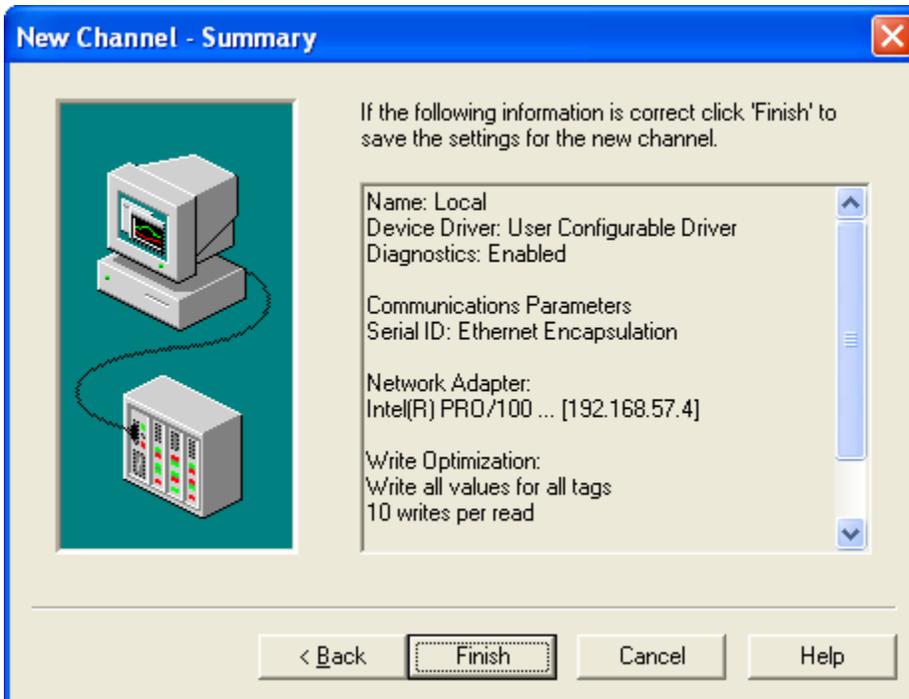


Click on Next.

In the following box, do not enable Unsolicited mode (the HygroFlex transmitter sends data only upon receiving a data request).



Click on Next. The following box provides a summary of the selections made in the previous steps.



Click on Finish.

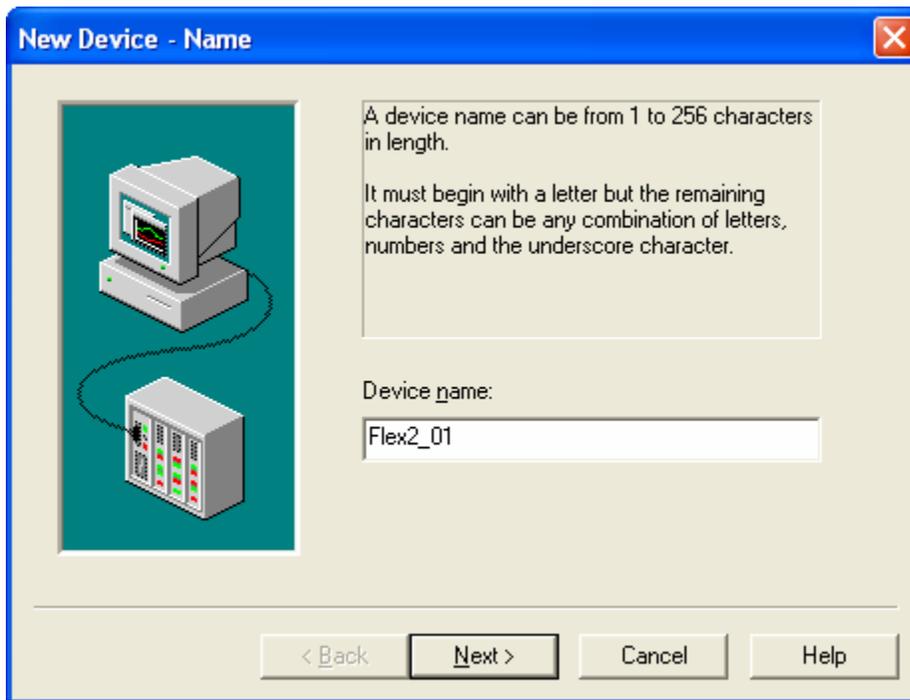
The TopServer left pane displays the channel that was just created. The red cross next to the channel indicates that no device has been created yet for this channel.



3) Create a Device

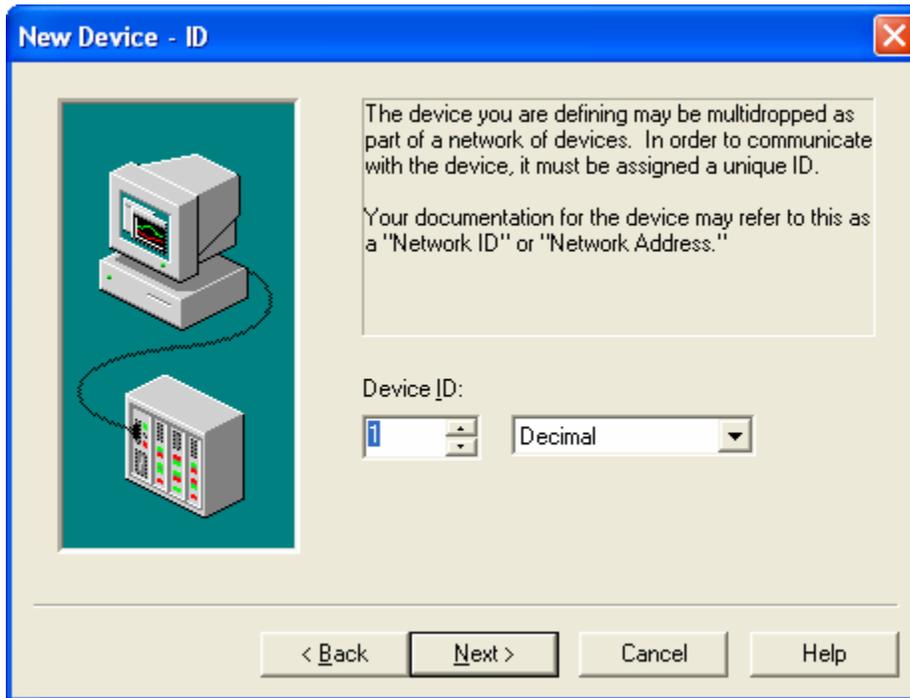
In this example, a device server is connected to the local channel (LAN). Two HygroFlex transmitters (Devices) with network address 01 and 02 are connected to the device server. Address 01 is connected directly by RS232 to the device server and address 02 is connected to address 01 by RS485.

Left click with the mouse to create the first device. To facilitate identification of the device, we used the name Flex2_01 for the transmitter with address 01.



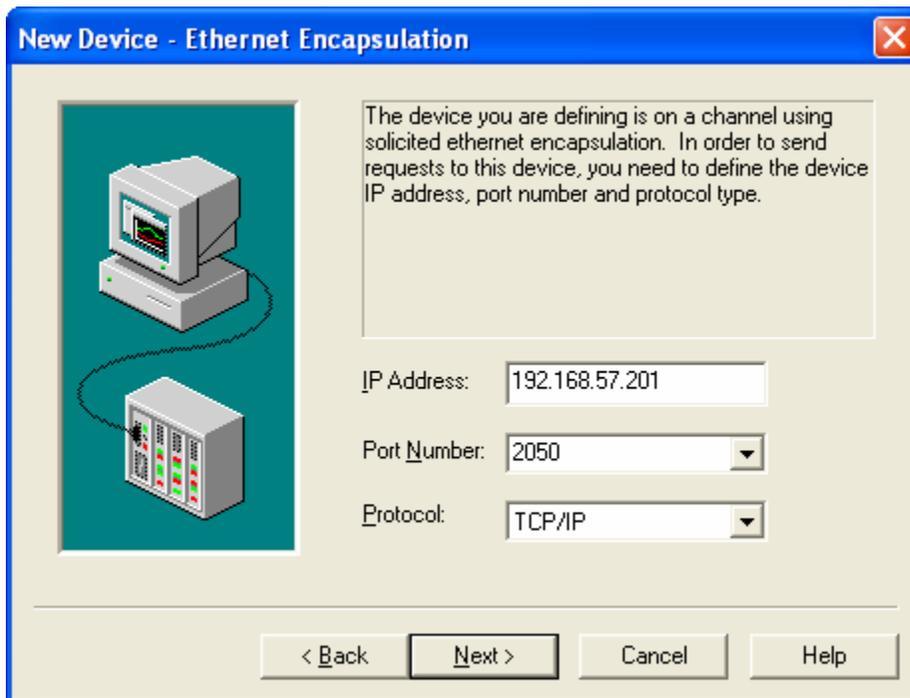
Click on Next.

Assign a unique ID to the transmitter (multi-dropped). In this example, we elected to use the address as the Device ID.



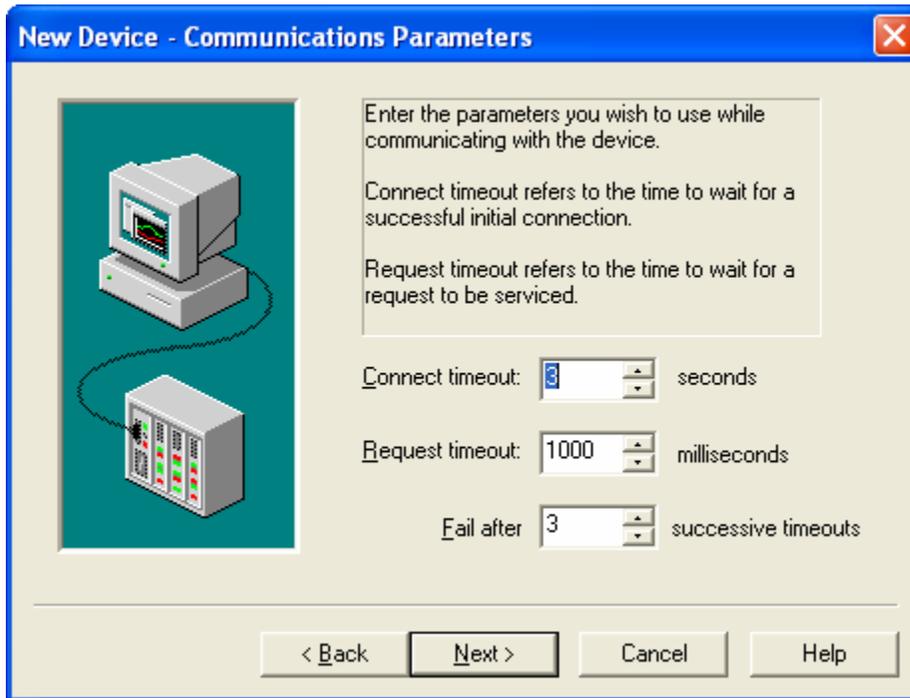
Click on Next.

Assign an IP address and a port number to the Device. In this example, this is the IP address and port number of the Ethernet server.

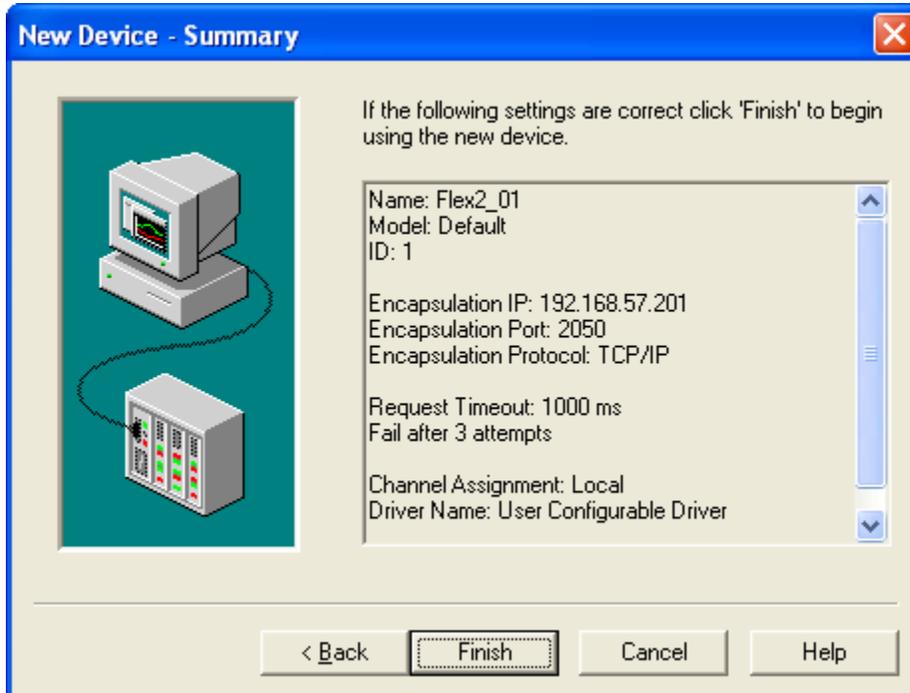


Click on Next.

In the following box, leave the default settings as they are.

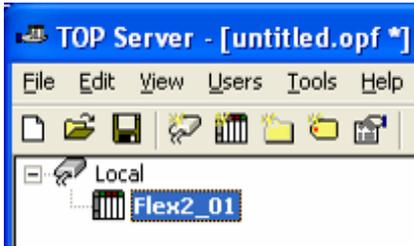


The following box provides a summary of the device settings.



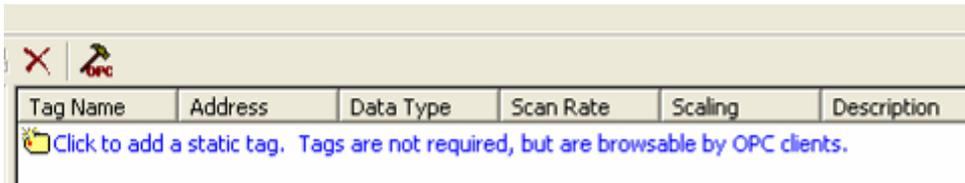
Click on Finish.

The TopServer left pane displays the Device that was just created.



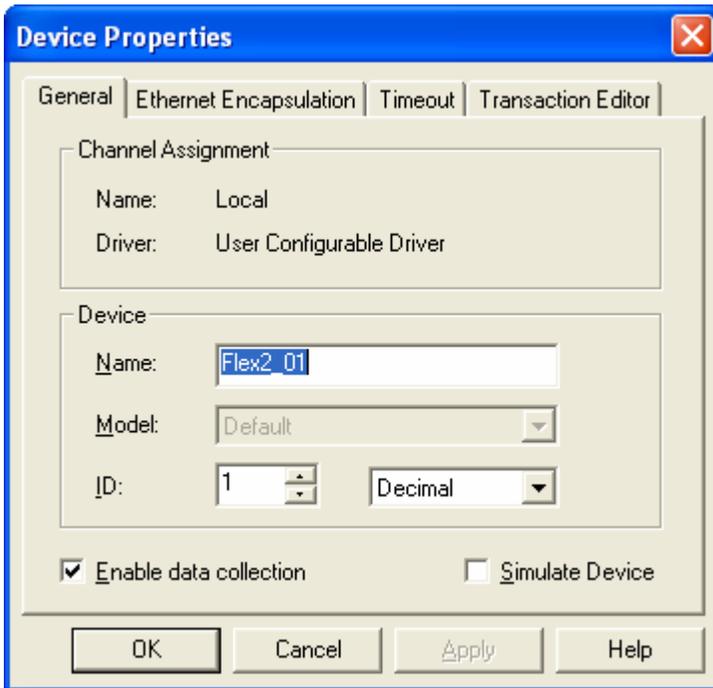
At this time, the device has no Tags attached to it.

Do not use the TopServer right pane to add a static tag. A tag will be created for the Device using a different process.

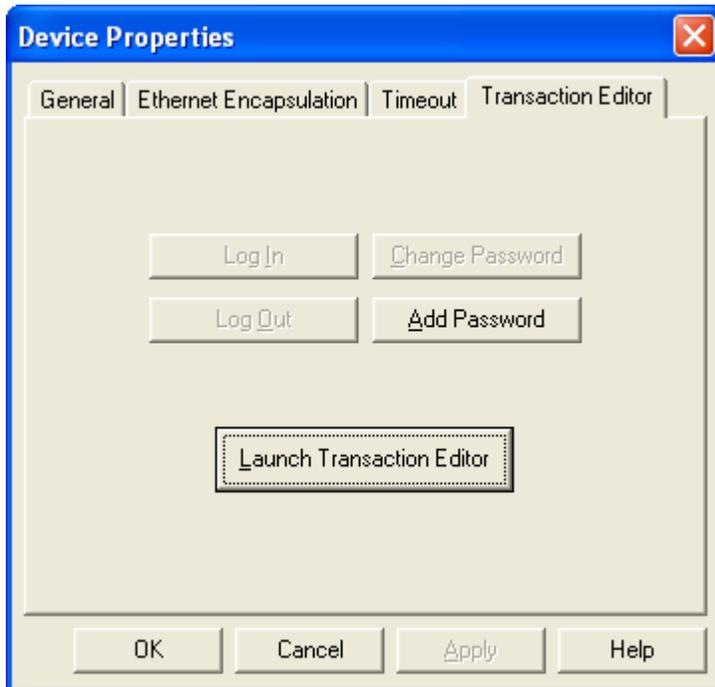


4) Add a Tag to the Device

Right click on the Device and select Properties to open the Device Properties box.

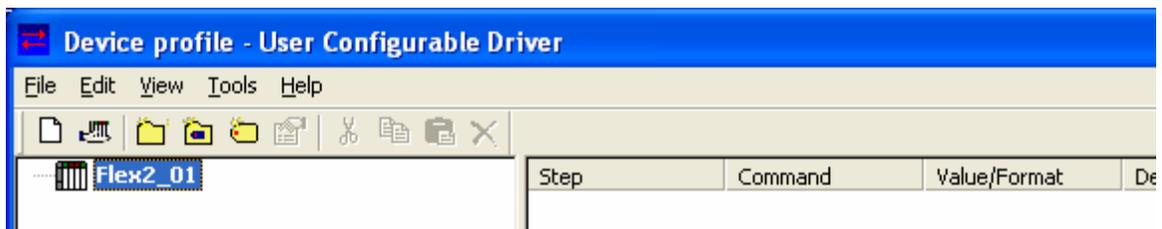


Select the Transaction Editor Tab.



Click on Launch Editor (this will work only is TopServer is currently running on your PC as a Service)

Transaction Editor opens the User Configurable Driver window. The left pane shows only devices (not Channels). Transactions will appear in the right pane as they are being created.

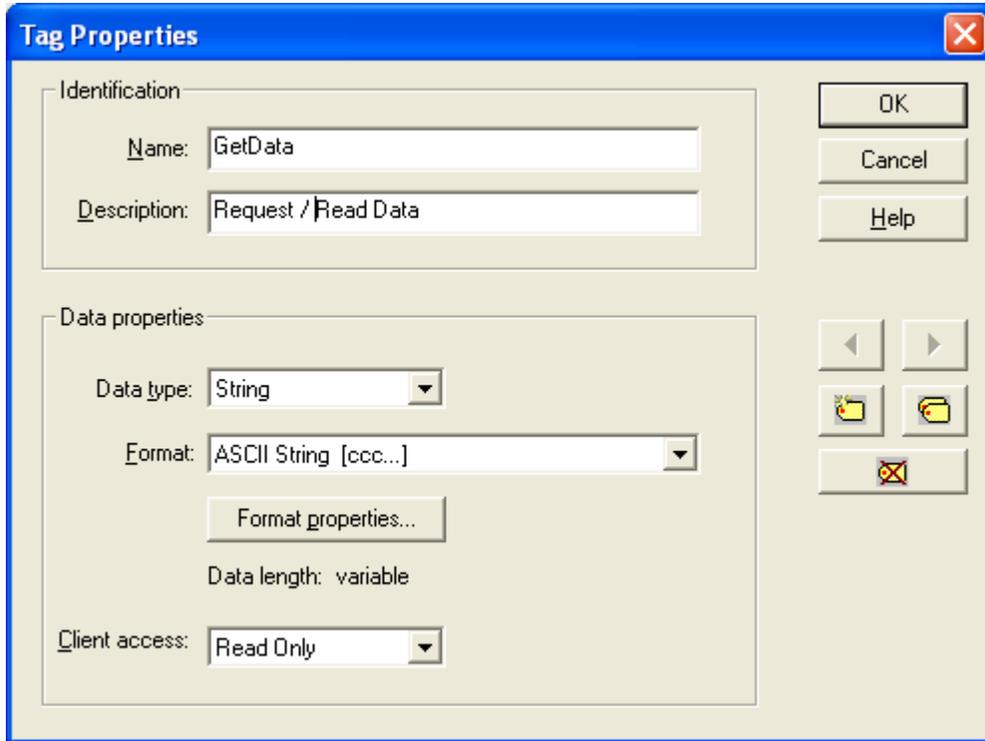


Right click on the Device and select New Tag.

This opens the Tag Properties box.

The names of the Tags attached to a device should be unique. Different devices may share the same Tag names. In this example, we use a Tag name that is easy to remember.

In the Tag properties box, enter the following text and make the following selections.



A transaction consists primarily of a data request and a response.

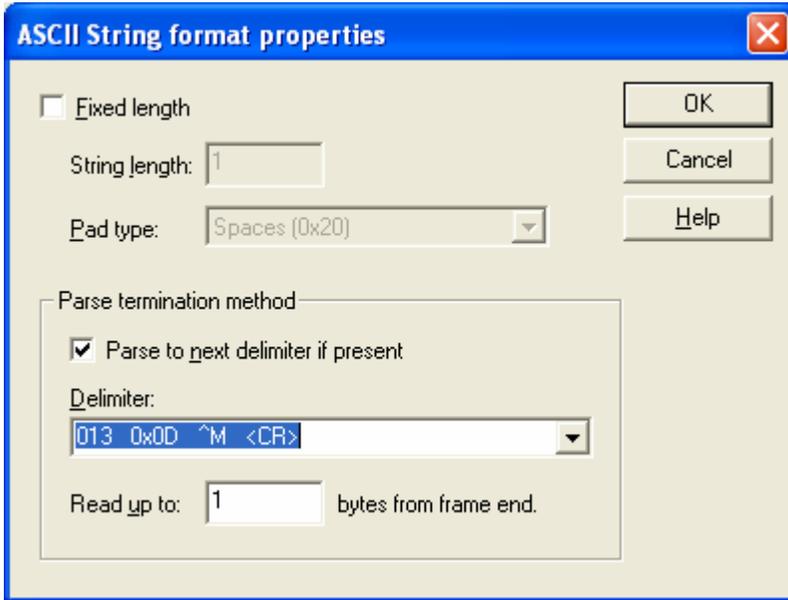
A data request to the HygroFlex 2 is a string immediately followed by a carriage return. The HygroFlex sends data back in the form of another string also immediately followed by a carriage return.

At the end of each transaction, the GetData Tag will hold the data string sent by the HygroFlex. Rather than defining the length of this string, we will make use of the fact that the string is immediately followed by a carriage return.

To do this, click on Format properties. This opens the ASCII string format properties box.

Uncheck the box “Fixed length” and check the box “Parse to next delimiter if present”.

For the delimiter, select the carriage return (013)

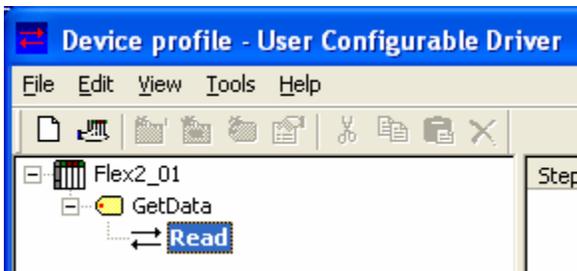


Click on OK to return to the Tag Properties dialog box. In the case of case of the HygroFlex, set the value of Client access can to Read Only. Click on OK.

The left pane of the User Configurable driver window displays the Tag that was just created.

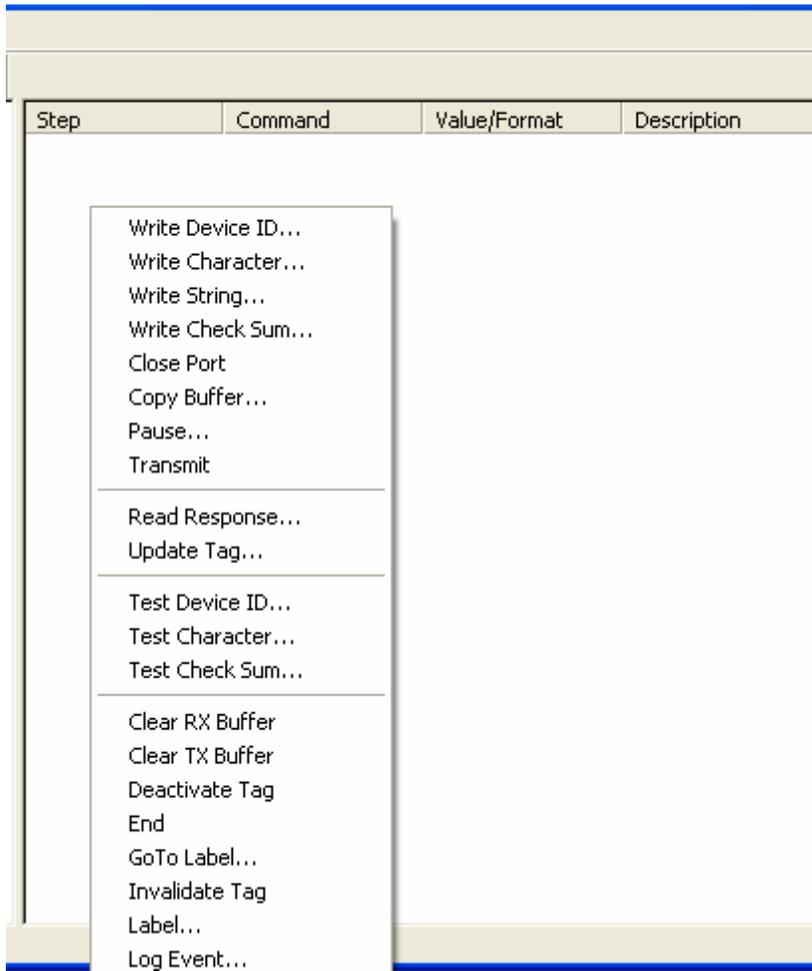


Expand the Tag by clicking on the + sign located to left of the tag.



Left click on Read.

Now, right click anywhere in the empty right pane of the User Configurable driver window. This causes the transaction menu to be displayed.



The transaction menu is used to enter the different steps that define a data request / data read transaction.

In the case of the HygroFlex 2 with network address 01, we will use the following data request::

{m01RDD0;} + CR

m : defines the transmitter as a HygroFlex 2
01 : network address 01 (use two digits)
RDD0; : requires humidity, temperature and calculated parameter from all probes

If the HygroFlex transmitter uses RS-232 to connect to the Device Server (iServer), you should add the symbol | (decimal code 124) as the first character of the data request just before the left curly bracket: |{m01RDD0;}

If the connection to the Ethernet server uses RS-485, do not use the | symbol.

Based on the above data request command, the response string from the HygroFlex 2 with network address 01 is structured as follows:

```
{m01RDD 0025.90;0015.82;-003.69;----.--;----.--;----.--;S
```

{m01RDD : repeats the data request command + blank space

0025.90 : %RH – probe 1
; : separator
0015.82 : °C – probe 1
; : separator
-003.69 : dew point °C – probe 1
; : separator
----.-- : %RH – probe 2 (optional)
; : separator
----.-- : °C – probe 2 (optional)
; : separator
----.-- : dew point °C – probe 2
; : separator
S : check sum (usually 2 characters)

Note: both the temperature unit and calculated parameter depend on the instrument configuration.

Several HygroFlex transmitters can be multi-dropped to form a network of transmitters sharing a single Ethernet server. Only the first transmitter is connected to the Ethernet server. Use either RS-232 or RS-485 for this connection. The second transmitter is connected to the first, etc. Use RS-485 for these connections. Each transmitter in the multi-drop should be assigned a unique network address.

We now create the transaction step by step as follows:

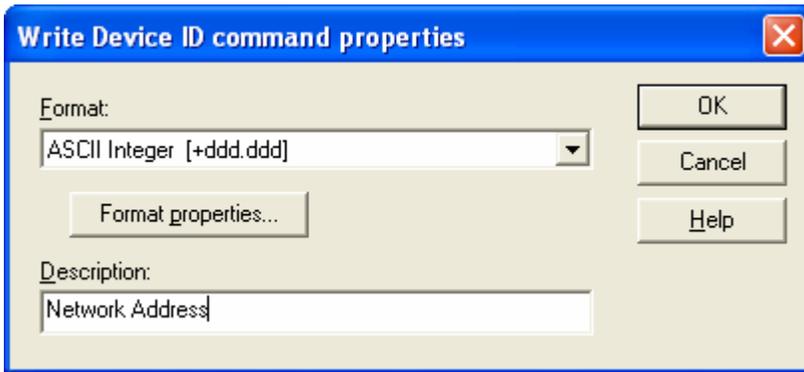
Step1: select “write string” from the transaction menu. Type in the beginning of the data request string and an appropriate description.



Click on OK. Step 1 of the transaction appears in the right pane.

Step	Command	Value/Format	Description
1	Write String	{m	1st part of data request

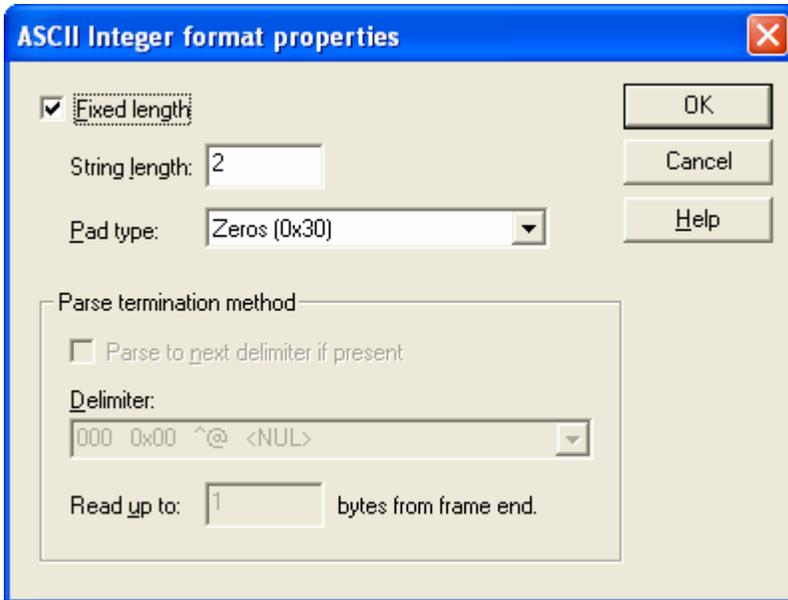
Step 2: In the right pane, right click below the line corresponding to step 1 of the transaction. This opens the transaction menu. Select "Write Device ID" and make the selections and entries shown below.



In this example the device ID is the same as the network address of the HygroFlex. Within a request command, the network address should be sent to the HygroFlex as a 2-character string (01 for address 1). In other words, a single digit address should be automatically padded with a zero.

To do this, click on Format properties.

In the ASCII Integer format properties box, make the following selections:



Click on OK to return to the Write Device ID form. Click on OK to close the form.

Steps 1 and 2 of the transaction now appear in the right pane.

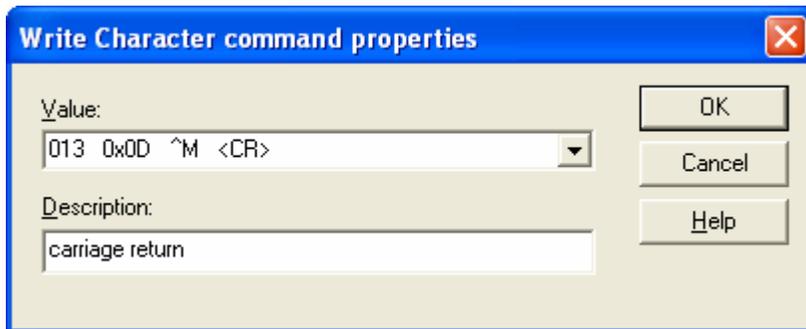
Step	Command	Value/Format	Description
1	Write String	{m	1st part of data request
2	Write Device ID	ASCII Integer ...	Network Address

Step 3: In the right pane, right click below the line corresponding to step 2 of the transaction. This opens the transaction menu. Select "Write String" and type in the remainder of the data request string.



Click on OK.

Step 4: In the right pane, right click below the line corresponding to step 3 of the transaction. This opens the transaction menu. Select "Write Character" and select the carriage return (013).

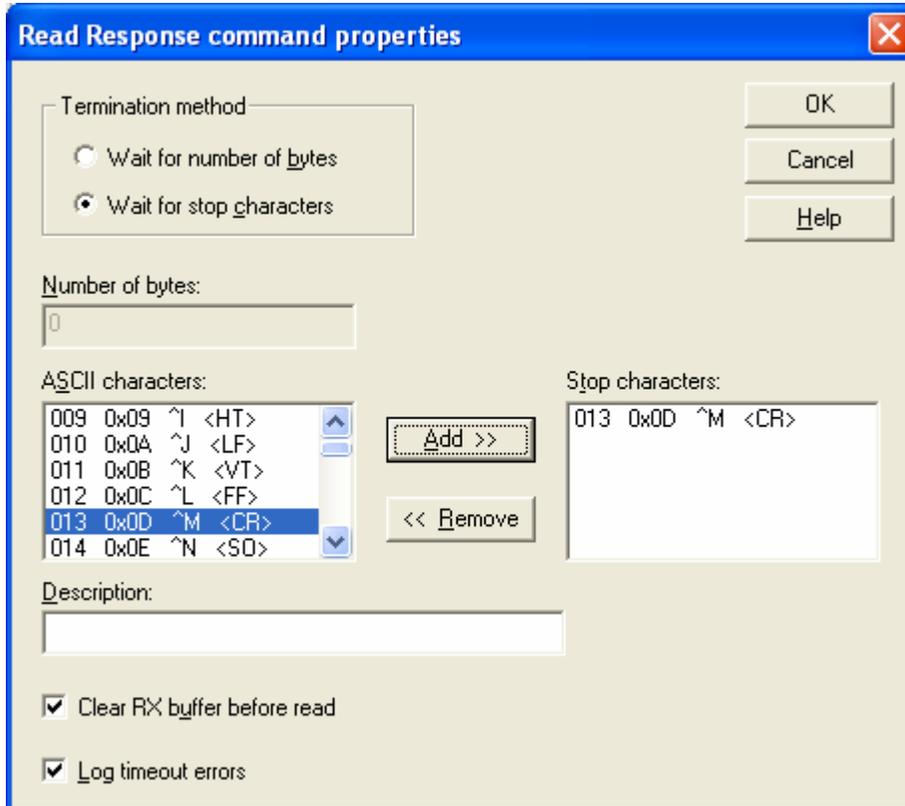


Click on OK.

Step 5: In the right pane, right click below the line corresponding to step 4 of the transaction. This opens the transaction menu. Select "Transmit" (send the string to the HygroFlex).

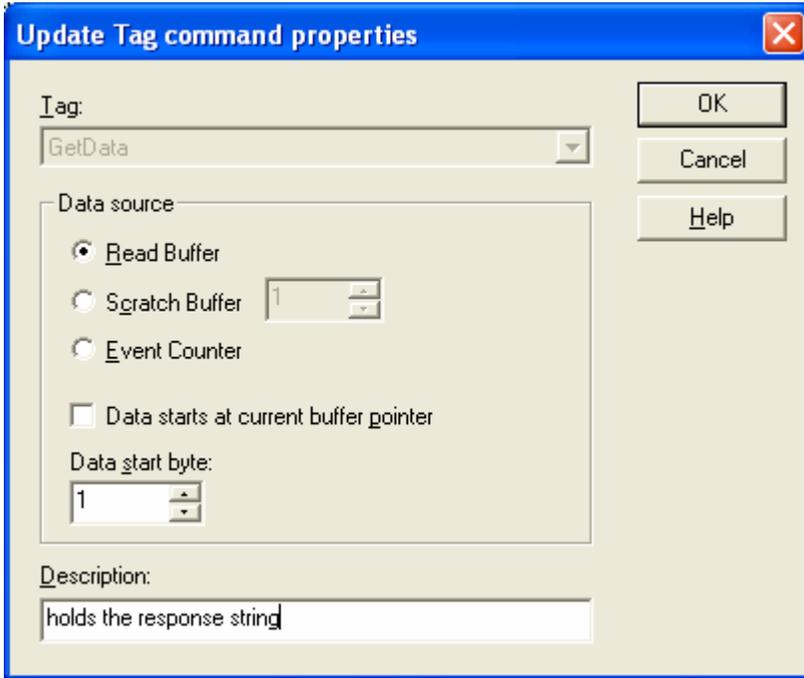
Step 6: In the right pane, right click below the line corresponding to step 5 of the transaction. This opens the transaction menu. Select "Read Response". This step causes the response of the HygroFlex to be written to the RX buffer (after clearing the buffer).

In the Read Response command properties, check "wait for stop characters", select carriage return as the stop character, click on Add. Leave the other settings as they are.



Click on OK

Step 7: In the right pane, right click below the line corresponding to step 6 of the transaction. This opens the transaction menu. Select "Update Tag". This step causes the contents of the RX buffer to be written to the GetData Tag.



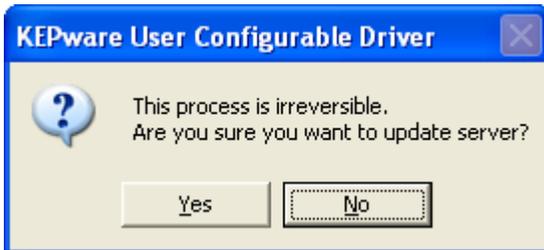
Click on OK.

All of the steps required by the transaction have been created:

Step	Command	Value/Format	Description
1	Write String	{m	1st part of data request
2	Write Device ID	ASCII Integer ...	Network Address
3	Write String	RDD0;}	2nd part of data request
4	Write Character	013 0x0D ^...	Carriage Return
5	Transmit	N/A	
6	Read Response	N/A	Response
7	Update Tag	GetData	Response String

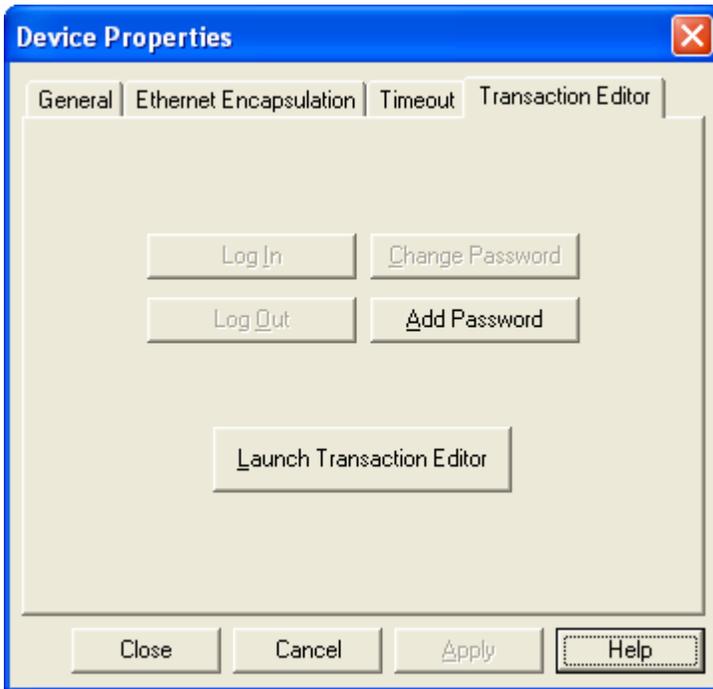
In the menu bar, click on File and on Update Server.

Click on YES in the following message box.

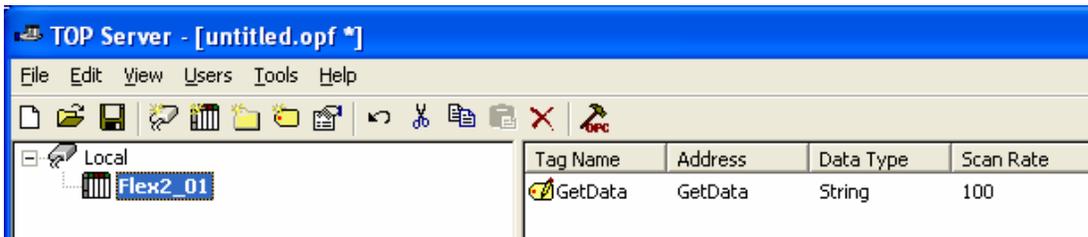


This returns TopServer to its main window.

Click on "close" in the Device Properties box.



The right pane of the TopServer main window shows the Tag that was just created.



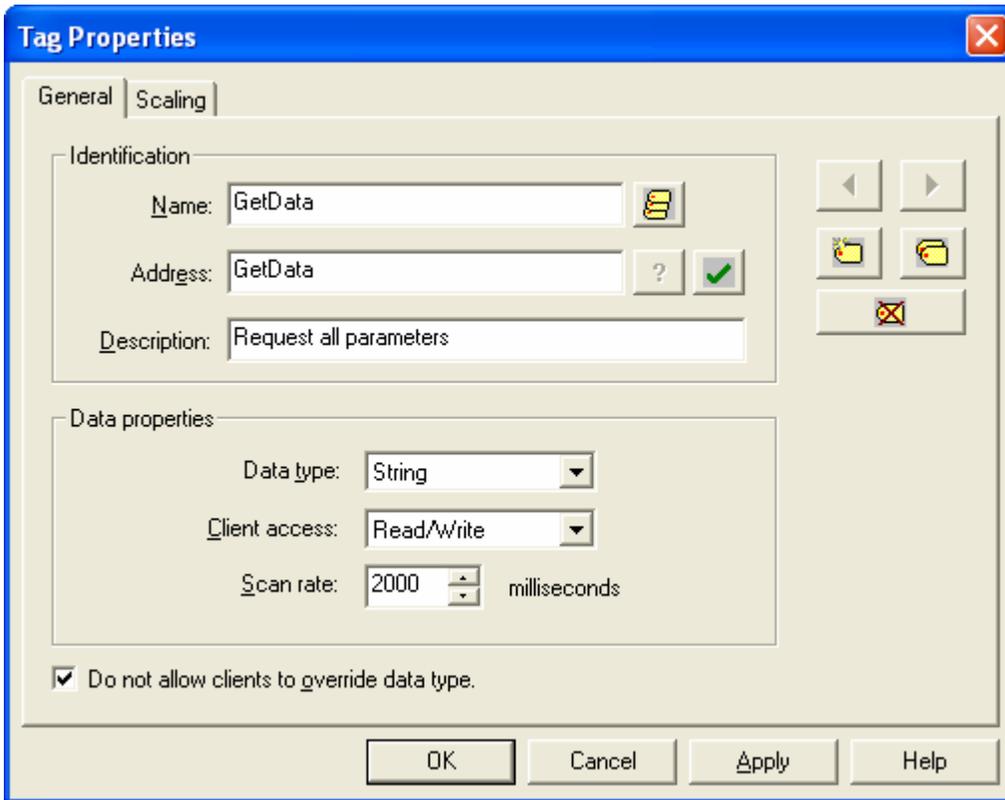
As this time, you should adjust the Scan Rate of the Tag to a value that is appropriate for both the Device and the Network.

In general, a data request and response transaction can be completed by the HygroFlex 2 in about 250 to 300 ms. Data from the HygroClip probe used by the HygroFlex is updated about every 700 ms. When two probes are connected to the HygroFlex, the data of the response string is fully updated every 1,400 ms.

The scan rate that is appropriate for the network depends on the traffic present on the network (LAN / Internet). In this example, we select a 2000 ms scan rate for the GetData Tag.

To change the Scan Rate, right click on the Tag and select Properties.

In the Tag Properties box, change the Scan Rate to 2000 ms.

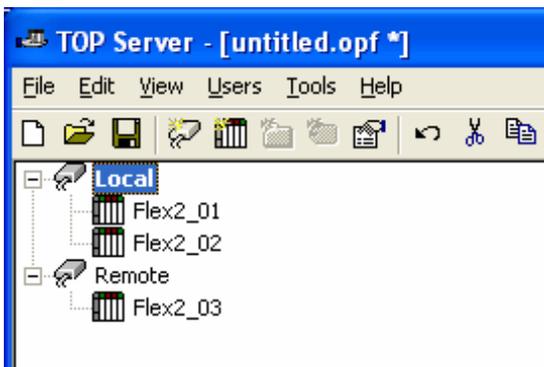


Click on OK.

The second HygroFlex (address 02) can be created in the server, under Local channel, in the same manner as we created the HygroFlex with address 01. The GetData Tag should be configured to reflect the correct network address of the transmitter.

The third HygroFlex is located on a remote network. In principle, it should be possible to add this transmitter to the same channel (Local) as used by the other two transmitters. The Device IP address should be a registered public IP address (not the IP address of the remote Ethernet server) and the port should be the port number that was set on the remote Ethernet server. The Tag Scan Rate should be compatible with the Internet.

In this example, we have elected to create a second channel with the name Remote. The properties of this channel are the exact same as the properties of the Local Channel. The IP address of Device Flex_03 (network address 03) is the public IP address of the remote router. The router is configured to forward requests received on port 2060 to the private IP address 192.168.57.200. The final result is as follows.



Using Microsoft Excel as an OPC client

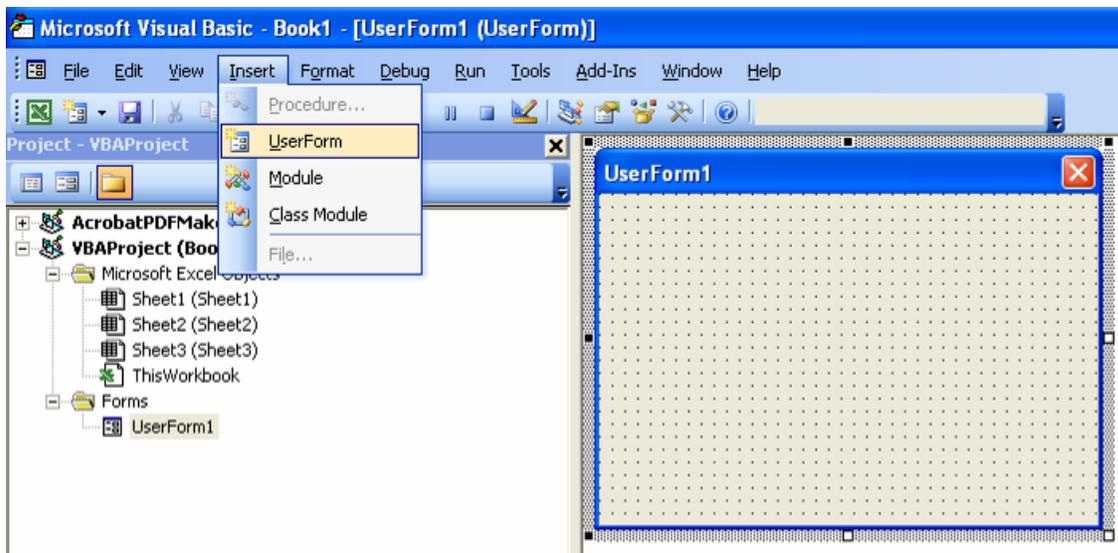
Microsoft Excel can be enabled to be an OPC client by embedding OPC Data ActiveX control in Excel. The OPC Data ActiveX control is designed to automatically open TOPServer.

Embedding OPC Data ActiveX Control in Excel / VBA

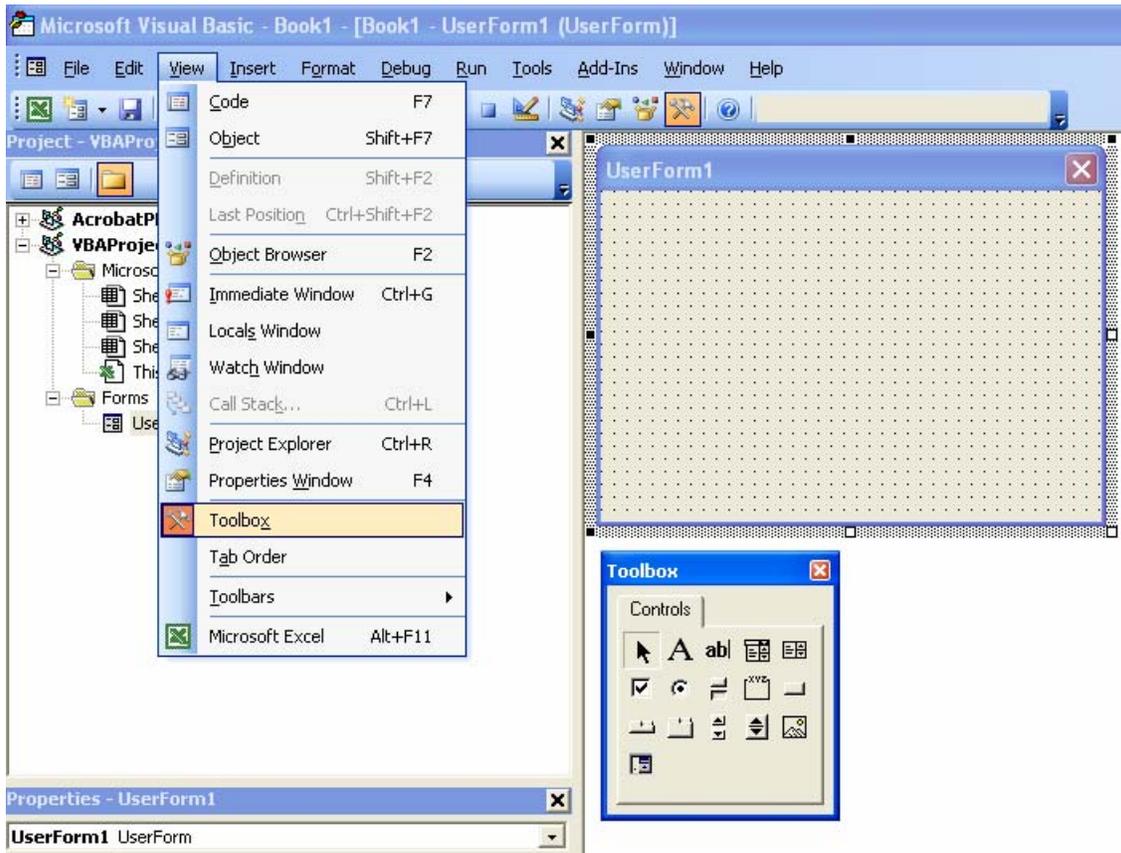
OPC Data ActiveX Control is a product from Software Toolbox, Inc. A fully functional 10-day free demonstration version can be downloaded using following link: [OPC Data ActiveX control](#)

With the OPC Data ActiveX control installed on your PC, open a new workbook in Excel and open Visual Basic (VBA).

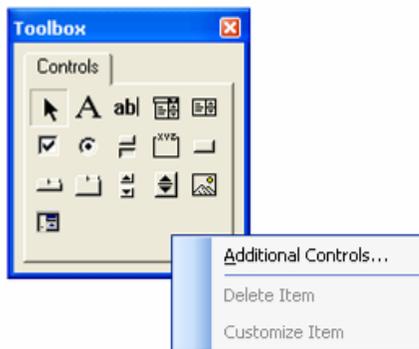
In the main menu bar of Visual Basic click on Insert and select and select User Form. This creates a blank user form.



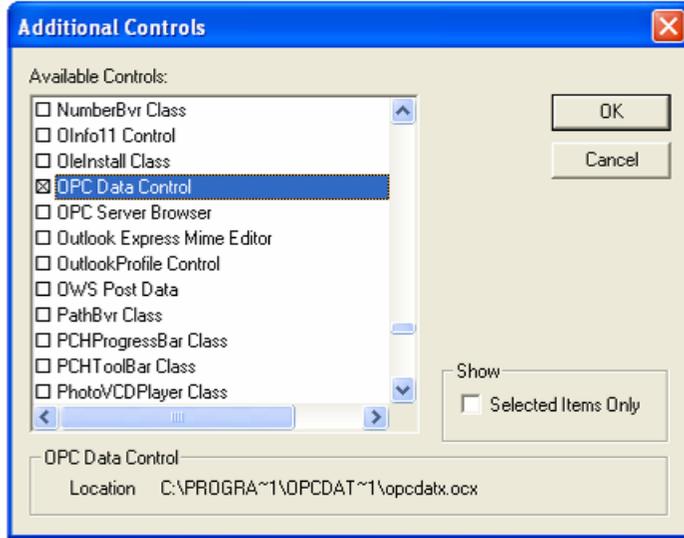
In the main menu bar of Visual Basic click on View and select Toolbox. This displays the Toolbox that is used to add controls to a form such as a text box, a label, a command button, etc.



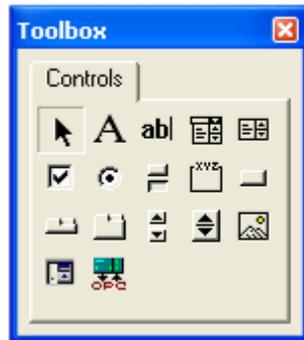
Right click on the Toolbox: and click on Additional Controls:



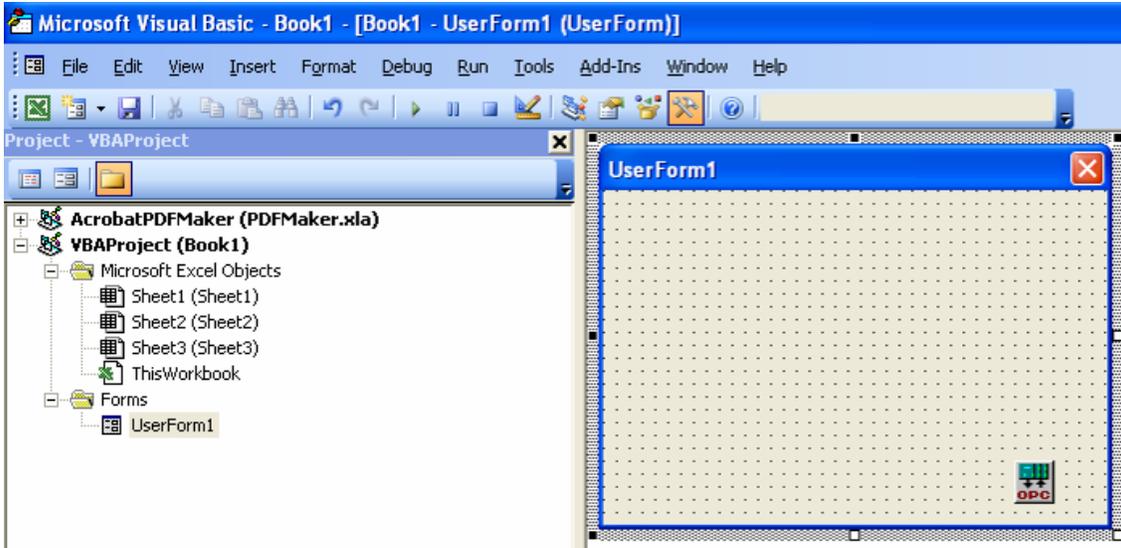
Go down the list of Controls and click on the line that says OPC Data Control. Click on OK.



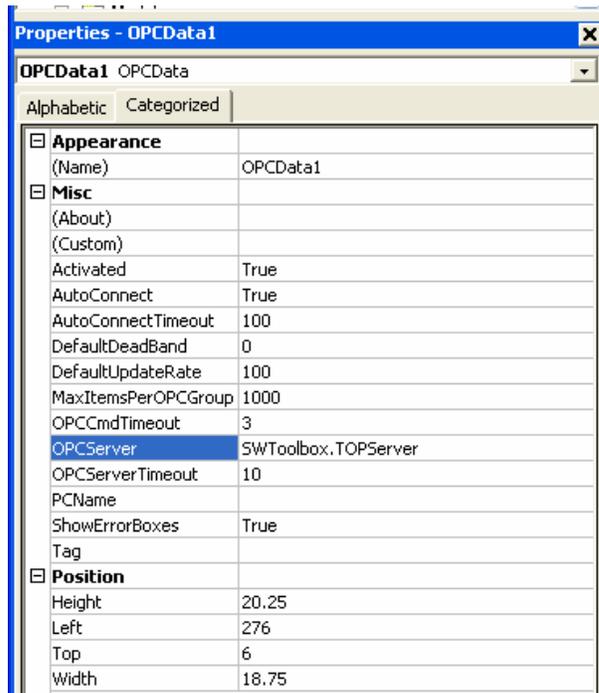
The OPC Data Control appears on the Toolbox:



In order to allow the form and its underlying code to communicate with the OPC server, the OPC Data Control should be added to the form just like any other control (text box, etc.).



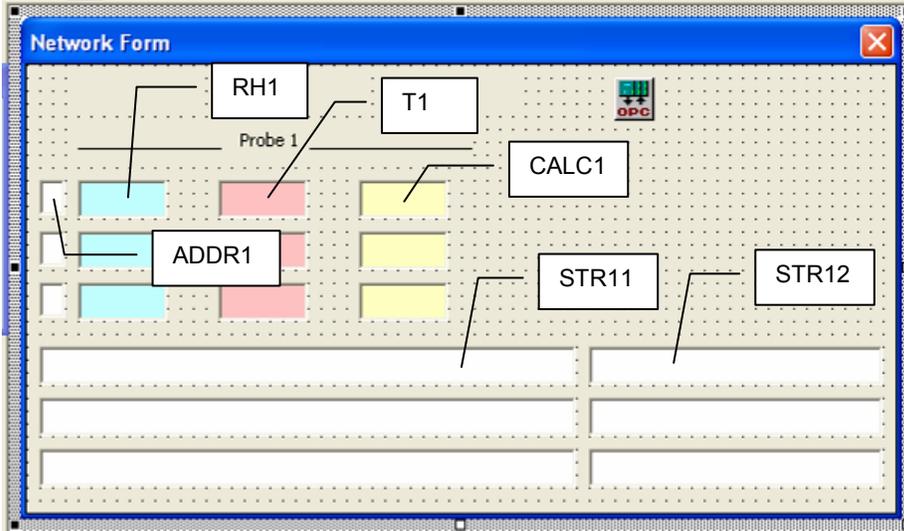
Right click on the OPC Data Control in the form. Visual Basic displays the properties of this item:



Under OPC Server, enter:

SWToolbox.TOPServer

As mentioned earlier, this technical paper assumes that the reader is already reasonably familiar with either Visual Basic or VBA. Therefore, this paper does not include instructions on how to create the following example of a form:



In the above form, text boxes have been created to display data from 3 HygroFlex 2 transmitters. For example, the following text boxes are used for the first HygroFlex 2:

- ADDR1 : network address (white background)
- RH1 : relative humidity, probe 1 (blue background)
- T1 : temperature, probe 1 (pink background)
- CALC1 : calculated parameter, probe 1 (yellow background)

Additional text boxes STR11 and STR 12 (white background) are used to display the name of the OPC tag and the value of the tag (in this case, the response string of the HygroFlex 2).

In Visual Basic, each text box has properties such as Text or Value. The Text or the Value properties define the contents of a text box. Data can be displayed after connecting a text box property such as Value with an OPC tag using the code underlying the form.

Code specific to the OPC Data ActiveX Control

The ConnectObject method of the OPC Data ActiveX Control can be used to create a **Subscription Read**. In a Subscription Read the OPC client tells the OPC server the data item that it wants updates for and how often it wants the data to be checked. Whenever the OPC Server detects that the item value has changed, it automatically makes a "callback" to the OPC client and delivers the data.

Syntax

```
result = object.ConnectObject (ConnectedObject, ConnectionTable)
```

Part	Description
result	A long value that indicates whether an error has occurred. Result is equal to zero if no error occurs.
object	(Required). An object expression that evaluates to an object in the Applies To list.
ConnectedObject	(Required). An object expression that evaluates to the object, which is to be connected. Typically, this is the property of a control that will be changed by the OPC Data Control (example: Me.Text1.Value - the Value property of text box Text1 in the active form). If this parameter is set to "Nothing", the OPC Data Control will generate the ValueChanged event if the connected variable changes.

ConnectionTable (Optional). A connection table. If ConnectionTable is omitted, the OPC Data Control will read the property ConnectionTable of the ConnectedObject.

A Connection Table consists of the following 4 entries:

- name of the property the OPC Data Control will write to and read from
- name of the OPC server item (or tag)
- update rate (or scan rate) to be used by the OPC server to check for changes in the OPC item
- dead band

Simple Visual Basic Example:

The following subroutine connects the Text property of the Text Box Text1 with an OPC Item:

```
Private Sub Command1_Click( )  
  
    Dim result As Long  
    Dim ConnectTable(3) As String  
  
    ConnectTable(0) = "Text"      ' Name of the property (text box Text1)  
    ConnectTable(1) = "ITEM"     ' Name of the OPC item  
    ConnectTable(2) = "200"      ' Update rate (ms)  
    ConnectTable(3) = "0.0"     ' Dead band  
  
    result = OPCData1.ConnectObject(Text1, ConnectTable)  
  
End Sub
```

Example of form programming (subscription read)

The following example illustrates how the ValueChanged event of the OPC Data Control can be used within a subroutine to call another subroutine where the response of the 3 HygroFlex 2 (network address 1, 2 and 3) is processed and then displayed in a form (see form example above).

1) Subroutine UserForm_Activate

This subroutine uses a special name in VBA and is called automatically when the form is opened. The form itself can be opened for example by clicking on a command button placed in an Excel spreadsheet. The subroutine connects to the OPC server and retrieves the contents of the tags that hold the data. Any change in the value of a tag triggers an event. When there is an event, the value changed subroutine is called by the ActiveX control.

```
Private Sub UserForm_Activate()  
  
    Dim Result1, Result2, Result3 As Long  
  
    'Address 01  
  
    Dim EventTable1(3) As String  
    EventTable1(0) = "Value"  
    EventTable1(1) = "Local.Flex2_01.GetData"  
    EventTable1(2) = "5000"  
    EventTable1(3) = "0.0"  
  
    Result1 = OPCData1.ConnectObject(Nothing, EventTable1)
```

```
'Address 02

Dim EventTable2(3) As String
EventTable2(0) = "Value"
EventTable2(1) = "Local.Flex2_02.GetData"
EventTable2(2) = "5000"
EventTable2(3) = "0.0"

Result2 = OPCData1.ConnectObject(Nothing, EventTable2)

'Address 03

Dim EventTable3(3) As String
EventTable3(0) = "Value"
EventTable3(1) = "Remote.Flex2_03.GetData"
EventTable3(2) = "5000"
EventTable3(3) = "0.0"

Result3 = OPCData1.ConnectObject(Nothing, EventTable3)

If Result1 <> 0 Then
    MsgBox "Error " & Result1 & " binding to Local.Flex2_01.GetData"
End If

If Result2 <> 0 Then
    MsgBox "Error " & Result2 & " binding to Local.Flex2_02.GetData"
End If

If Result3 <> 0 Then
    MsgBox "Error " & Result3 & " binding to Remote.Flex2_03.GetData"
End If

End Sub
```

Note:

The above 3 Connection Tables refer to the tags that were created in TOPServer (the OPC server), one for each HygroFlex 2 transmitter. The OPC server (TOPServer) will check the contents of these tags every 5000 ms (this overrides the 2000 ms that we specified directly in TOP Server). In order to fire the ValueChanged event, "Nothing" is used as the first parameter of the ConnectObject method. When the ValueChanged event is fired, the subroutine OPCData1_ValueChanged is called automatically.

2) Subroutine OPCData1_ValueChanged

This subroutine can be created automatically by double clicking on the OPCData1Control added earlier to the form. When the ConnectObject method fires a ValueChanged event, it automatically calls this subroutine and passes the following variables, one tag at a time:

Property	: string with the name of the property
VarName	: string with the name of the connected OPC item (or tag)
Value	: variant with the new value of the OPC item (or tag)
Quality	: integer with the quality of the new value (a value of 192 indicates a good OPC connection)

In this example, the subroutine is as follows:

```
Private Sub OPCData1_ValueChanged(ByVal Property As String, ByVal VarName As String, ByVal Value As Variant, ByVal Quality As Integer)

    Dim ADDR As String
    Dim RH, T, CALC As Variant
```

```
ADDR = Mid(Value, 3, 2)
```

```
RH = Mid(Value1, 9, 7)
```

```
If RH <> "----.-" Then
```

```
    RH = Val(RH)
```

```
Else
```

```
    RH = "n/a"
```

```
End If
```

```
T = Mid(Value, 17, 7)
```

```
If T <> "----.-" Then
```

```
    T = Val(T)
```

```
Else
```

```
    T = "n/a"
```

```
End If
```

```
CALC = Mid(Value, 25, 7)
```

```
If CALC <> "----.-" Then
```

```
    CALC = Val(CALC)
```

```
Else
```

```
    CALC = "n/a"
```

```
End If
```

```
'Address 01 _____
```

```
If VarName = "Local.Flex2_01.GetData" Then
```

```
    Me.ADDR1.Text = ADDR
```

```
    Me.RH1.Value = Format(RH, "#,##0.00")
```

```
    Me.T1.Value = Format(T, "#,##0.00")
```

```
    Me.CALC1.Value = Format(CALC, "#,##0.00")
```

```
    Me.STR11.Text = Value
```

```
    Me.STR12.Text = VarName
```

```
End If
```

```
'Address 02 _____
```

```
If VarName = "Local.Flex2_02.GetData" Then
```

```
    Me.ADDR2.Text = ADDR
```

```
    Me.RH2.Value = Format(RH, "#,##0.00")
```

```
    Me.T2.Value = Format(T, "#,##0.00")
```

```
    Me.CALC2.Value = Format(CALC, "#,##0.00")
```

```
    Me.STR21.Text = Value
```

```
    Me.STR22.Text = VarName
```

```
End If
```

```
'Address 03 _____
```

```
If VarName = "Remote.Flex2_03.GetData" Then
```

```
    Me.ADDR3.Text = ADDR
```

```
    Me.RH3.Value = Format(RH, "#,##0.00")
```

```
    Me.T3.Value = Format(T, "#,##0.00")
```

```
    Me.CALC3.Value = Format(CALC, "#,##0.00")
```

```
    Me.STR31.Text = Value
```

```
    Me.STR32.Text = VarName
```

```
End If
```

End Sub

Notes:

- Whenever there is a change, data (the response string from a HygroFlex 2) is passed to the subroutine one tag (one HygroFlex 2) at a time.
- The response string from the HygroFlex 2 is broken down into its components such as address, relative humidity, etc.
- Based on the name of the OPC tag, the new data is written to the form in the appropriate text boxes.

For synchronized reads, use any of the following methods of the OPC Data ActiveX Control: ReadVariable or ReadMultiVariables