Atop Technologies, Inc.

Protocol Gateway
DNP3.0 Client/Server

Protocol and
eNode Designer configuration

eNode Configuration Manual

V1.3

September 29th, 2017

This PDF Document contains internal hyperlinks for ease of navigation.
For example, click on any item listed in the Table of Contents to go to that page.

Published by:

Atop Technologies, Inc.

2F, No. 146, Sec. 1, Tung-Hsing Rd,
30261 Chupei City, Hsinchu County
Taiwan, R.O.C.

Tel: +886-3-550-8137
Fax: +886-3-550-8131
www.atoponline.com
www.atop.com.tw

Important Announcement

Disclaimer

We have checked the contents of this manual for agreement with the hardware and the software described. Since deviations cannot be precluded entirely, we cannot guarantee full agreement. However, the data in this manual is reviewed regularly and any necessary corrections will be included in subsequent editions.
Suggestions for improvement are welcome.
All other product's names referenced herein are registered trademarks of their respective companies.

Documentation Control

Author:	Matteo Tabarelli (Atop)
Revision:	1.3 Enhancement
Revision History:	eNode upgrade
Creation Date:	12 April 2017
Last Revision Date:	29 September 2017
Product Reference:	PG59XX Protocol Gateways
Document Status:	Released

Table of Contents

Table of Figures

1 Introduction

Thank you for Buying Atop's Protocol Gateway.

The product is bundled with the following three user manuals:

1) Hardware specific installation user manual, **not covered in this document.** It covers Atop's hardware installation procedure, wiring, power connection etc.
2) Getting started with Atop's Protocol Gateway user manual – configuration tool introduction, web configuration, software architecture introduction– **not covered in this document**. This manual covers the introduction, installation, network set-up maintenance and using of the configuration tool software, including the procedure to be followed for uploading new configurations to Atop's device.
3) Protocol specific user manual **(This Manual). One protocol-specific manual will be provided for each protocol installed on the device.** This manual covers:
    a. Basic device network configuration
    b. Step-by-step protocol set-up for in eNode designer
    c. Description of the protocol-specific software features, the device profile and the implementation table of supported functionalities.

This manual is for **DNP3.0 Client/Server** and describes how to use the **DNP3 eNode Designer Module** to configure Atop's *DNP3 ADH Application* within the eNode Designer configuration tool.

1.1 *Scope*

This document is divided into 3 major sections:

- **General Description**;
- **Configuration Guide**; and
- **DNP3.0 Client Device Profile/ Implementation table** and
- **DNP3.0 Server Device Profile/ Implementation table**

1.2 *Document Reference*

[1] Document Title: Getting started User Manual: 197-0100
Revision: Version 1.00 or higher

[2] Document Title: DNP3 Specification
Revision: 2.05, 24th June 2009

1.3 *List of Abbreviations*

ADH = Application Data Hub
DNP3 = Distributed Network Protocol 3
IED = Intelligent Electronic Device

IP = Internet Protocol
TCP = Transmission Control Protocol

2 General Description

The DNP3 eNode Module can be used to configure the DNP3 ADH Application as a master or slave. For naming consistency across eNode Designer, the master is called a client, and the slave is called a server.

The DNP3 client can communicate with many DNP3 server IEDs, all of whose data point details can be configured using this module.

2.1 *Configuration Theory*

Most configuration properties describe a *server*. When configuring the ADH application server, you are configuring the properties of the server itself. Atop protocol gateway supports one server application per protocol per port. When configuring the ADH Application client, you are describing the properties of all the remote servers with whom the client is communicating.

Configuring the protocol-specific information, such as object addresses, is handled in the module. This is explained in this document.

Communication port properties (such as Baud Rate) are configured on the communication port itself. The Device module handles the communication port properties, so heavy details are outside the scope of this document.
Port configuration instructions are provide in the eNode Designer general user manual.
Screenshots of the typical configuration method are anyway shown in section 4. The relevant properties of the communication ports automatically apply to the application. For example, in a DNP3 server application, the IP Address the application binds to is taken from the parent Ethernet port.

2.2 *General Screen Description*

A small configuration example is shown below to better help describe the layout of the screen.

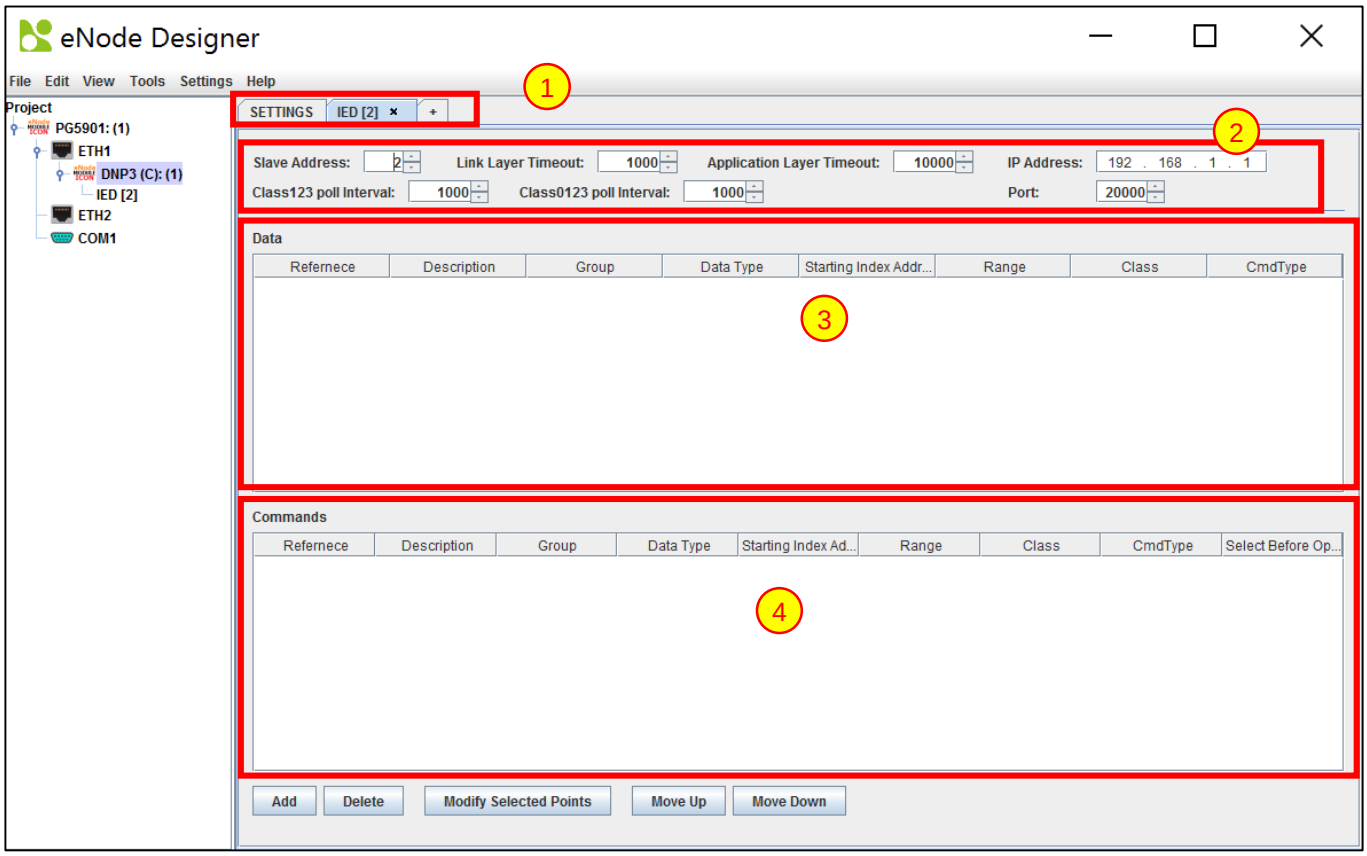


Figure 2-1 - Example Screen

1. **Tabs** –
2. **Server IED Properties** – Describes the protocol-specific properties of the server IED.
3. **Data Table and buttons** – Shows all (information) data associated with the IED, and shows the buttons to be used to modify them.
4. **Commands Table and buttons** – Shows all commands associated with the IED, and shows the buttons to be used to modify them.

The user is able edit contents of the data and commands tables freely.

3 DNP3 Configuration Guide

3.1 *Adding the Module in eNode Designer*

The DNP3 module can be added to both *Ethernet* and *Serial* ports.

The application can be set up as a Client or a Server. The choice will be presented when adding it to the project.

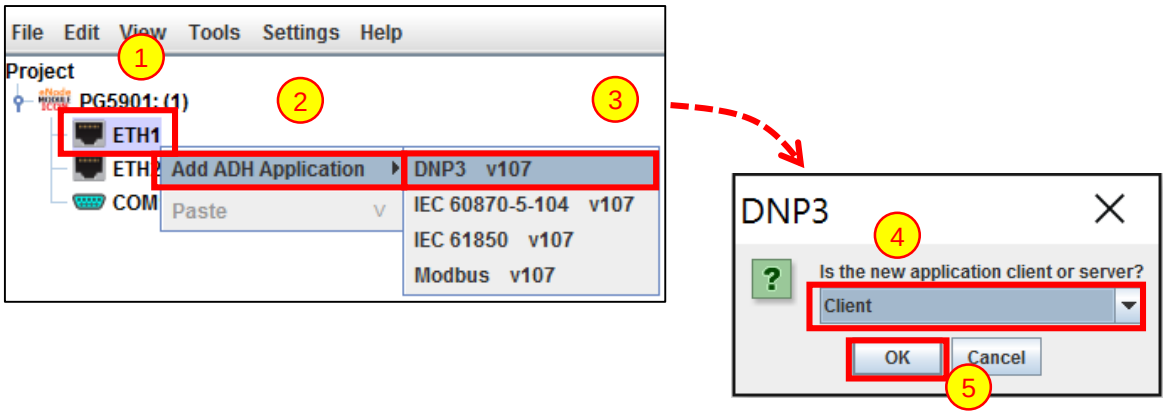


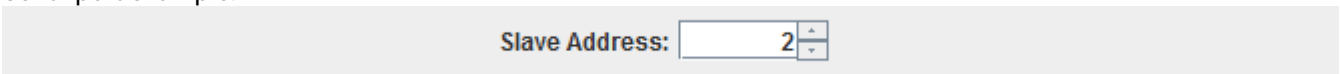
Figure 3-1 - Adding the module in eNode Designer.

1. Right click the desired **communication port**.
2. Open the **Add ADH Application** menu.
3. Select **DNP3**.
4. Select **Client** or **Server** from the drop-down menu.
5. Click **OK**.

3.2 *Server IED Properties*

The server IED properties are at the top of the module screen. The options available are limited to what is relevant for the communication port.

Serial port example:



Ethernet port example:

Each property is described in detail below.

3.2.1.1 Slave Address

Description	The slave address of the server IED. For servers it describes its own slave address. For clients, it describes the slave address of the remote server.
Data Entry	Integer
Range	*0 to 65519*
Input Option	Mandatory

3.2.1.2 IP Address

Ethernet and Client only

Description	The IP Address of the remote server IED. This option is only available in clients, since in servers, the IP Address is taken from the Ethernet port.
Data Entry	IP Address String
Range	*Valid IPv4 Addresses (0.0.0.0 to 255.255.255.255)*
Input Option	Mandatory

3.2.1.3 IP Port

Ethernet only

Description	The IP Port used by the server IED.
Data Entry	Integer
Range	1 *to 65535. Default: 20000*
Input Option	Mandatory

3.3 *Client Configuration*

Adding a DNP3 client application will immediately show the following figure. The first tab shows the settings that apply to the whole client application. Each tab after this shows a single DNP3 server with which the client is communicating. Each tab is named "IED [{X}: {Y}]" where {X} is the master address and {Y} is the slave address.

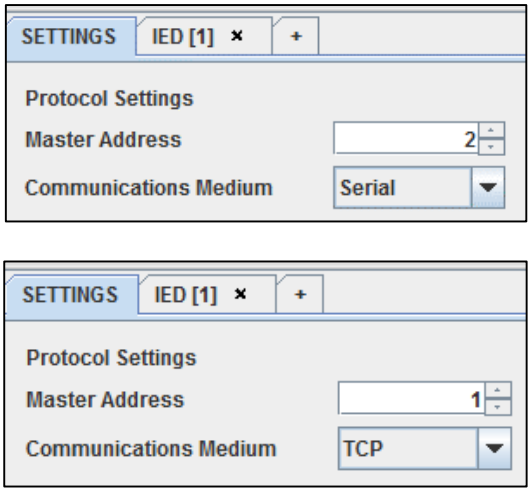


Figure 3-2 - Client settings panel (serial above; Ethernet below).

Selecting the IED tab will show the following view.

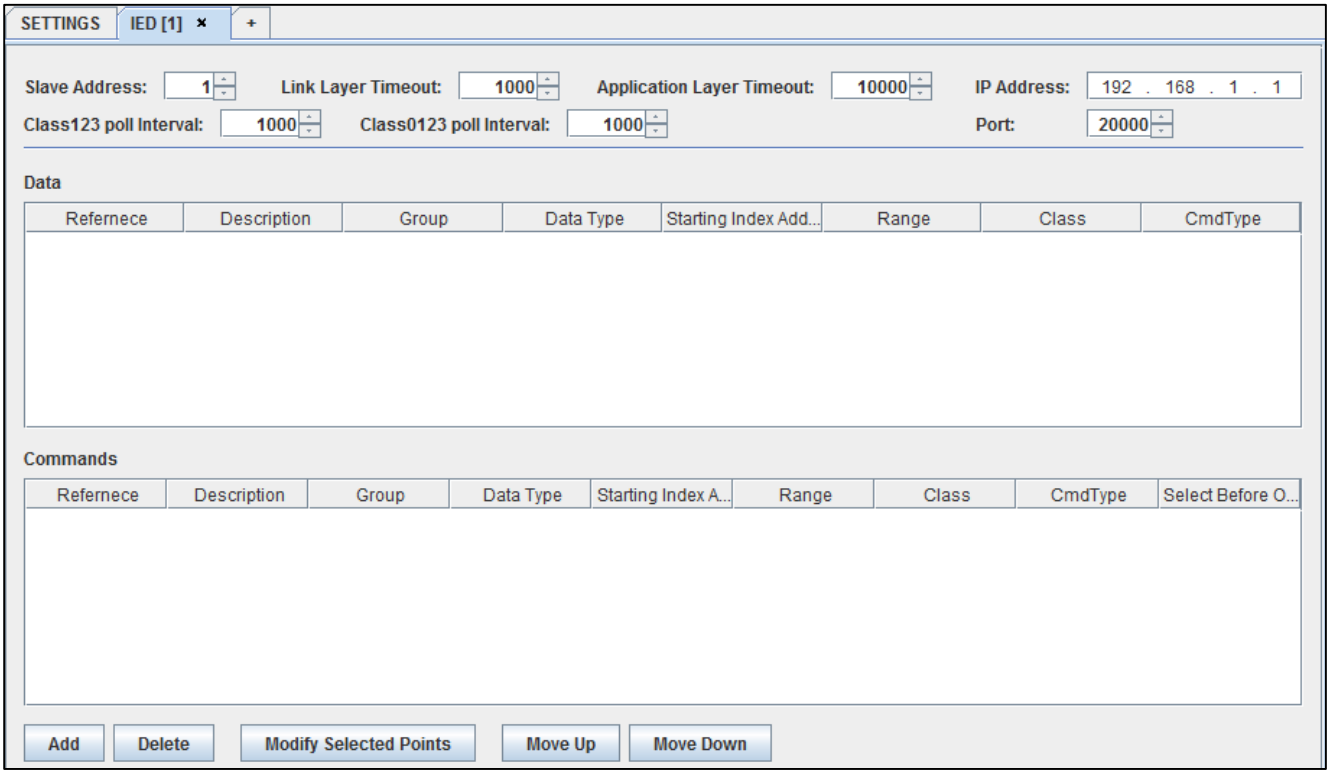


Figure 3-3 - Client IED panel (Ethernet).

Here the "Add" and "+1" button can be used to add data points. Adding data points is explained in the "Add data points" section, and the other buttons are described in section 6: Reference Guide.

3.3.1 Client Settings

3.3.1.1 Master Address

Description	The DNP3 master address to use.
Data Entry	Integer
Range	*0 to 65519*
Input Option	Mandatory

3.3.1.2 Communication Medium

Description	The communication method to use. The list will automatically be restricted based on the parent's port type.
Data Entry	Drop down menu
Options	*Serial, TCP, UDP*
Input Option	Mandatory

3.3.1.3 Link Layer Timeout (ms)

Description	The timeout for a data link layer confirmation in milliseconds.
Data Entry	Integer
Range	*100 to 65535*
Input Option	Mandatory

3.3.1.4 Application Layer Timeout (ms)

Description	The timeout for an application layer confirmation in milliseconds.
Data Entry	Integer
Range	*2000 to 65535*
Input Option	Mandatory

3.3.1.5 Class 0, 1, 2, 3 Poll Interval (ms)

Description	The poll interval of the event classes: class 1, 2 and 3. Every poll interval all event classes are polled. Measured in milliseconds.
Data Entry	Integer
Range	*1000 to 65535*
Input Option	Mandatory

3.3.1.6 Class 0 Poll Interval (ms)

Description	The poll interval of static data. Every poll interval all data is polled for its present value: class 0, 1, 2 and 3. Measured in milliseconds.
Data Entry	Integer
Range	*1000 to 65535*
Input Option	Mandatory

3.3.2 Adding Data Points

To add data points, left click the "**Add**" button beneath the tables in the main view. Doing so will show the following window. The window is used to add many data points at once with the specified values. For details on the meaning of each column, see section 6.2.

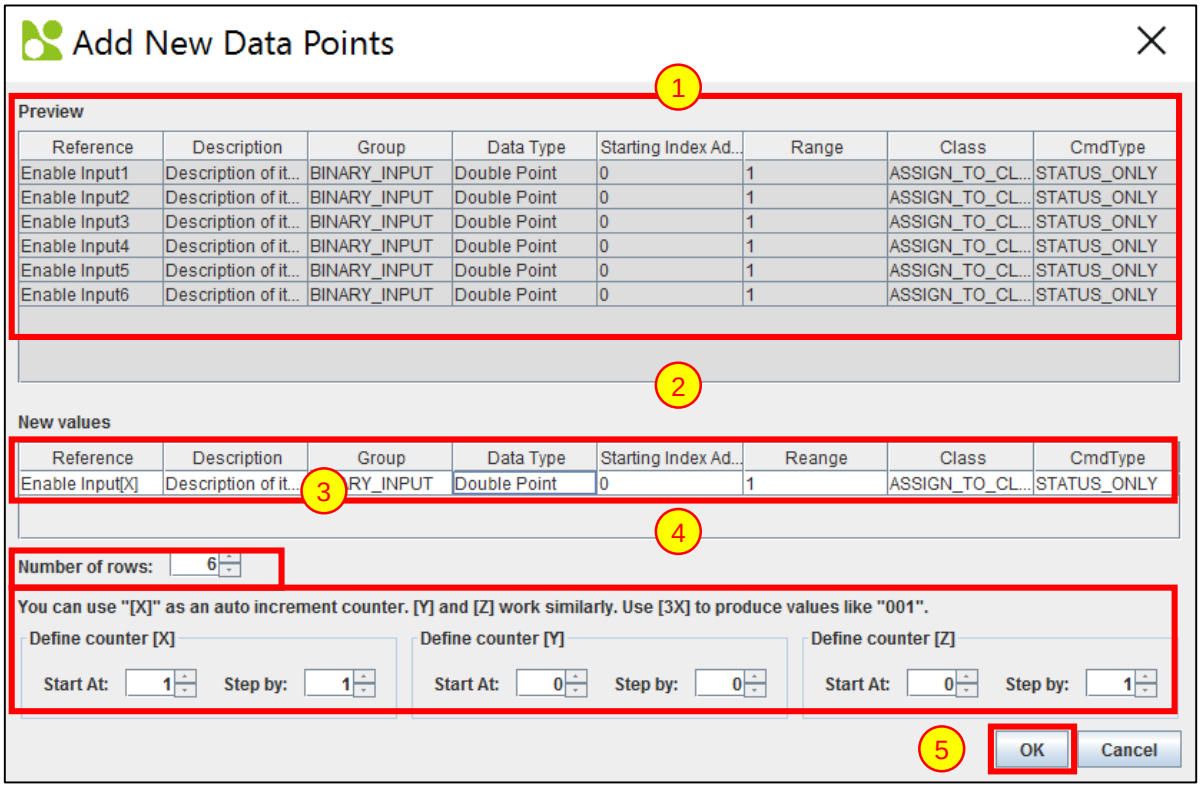


Figure 3-4 - Add data points window.

1 **Preview Area** – Shows the preview of the data points that will be added.

2 **New values** – This area is used to enter values. Tag, description, Starting Index Address and Range use manual data entry (click the box and type new values). Data type, CmdType and class use drop-down menus. Entering an integer into the address column will start at that number and automatically increment in each successive point.

3 **Number of rows –** This counter can be used to add many data points at once.

4 **Automatic Counters** – These counters can be used to add many data points at once. The starting values and step values can be changed in this area. See also 5 Using Auto-increment Counters.

5 **OK button** – to accept the new data points.

3.3.3 Servers (Remote IEDs)

Each slave IED is represented by a single tab and a tree node in the eNode Designer project tree.

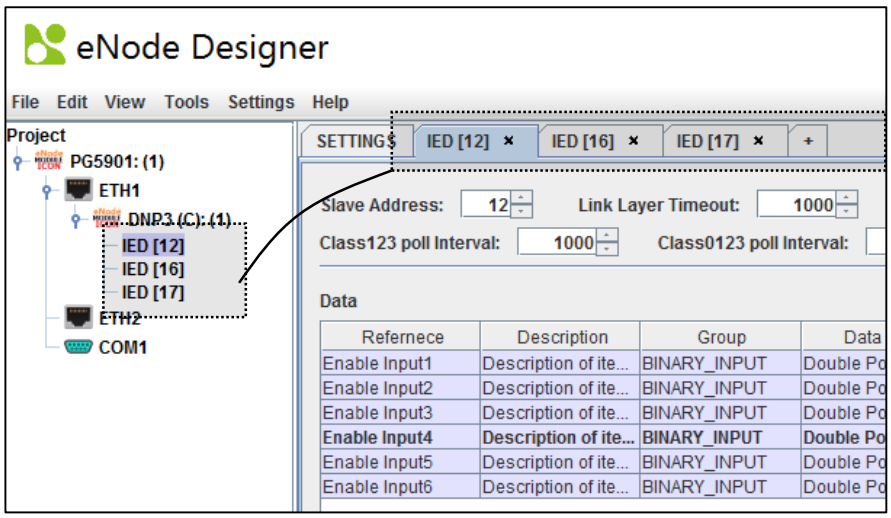


Figure 3-5 - Multiple connected servers example.

To modify the connected IEDs list follow the instructions below:

1. To **add** a new remote IED, click the **"+"** tab at the end of the existing remote server(s) list.

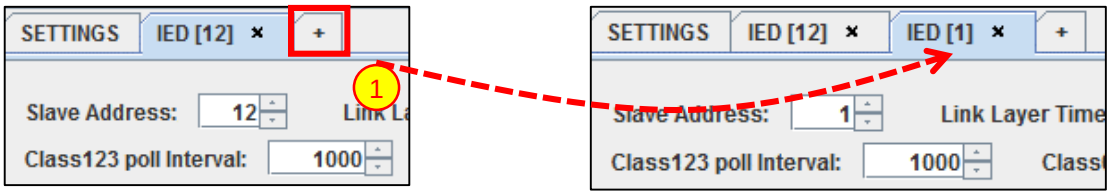


Figure 3-6 - Add a connected server.

To remove a remote IED, it must have no data points specified. If there are data points in the table and you still wish to remove the IED, you will have to remove such data points first.

1. To **remove** a remote IED, click the cross on the right side of the tab of the IED you wish to remove.

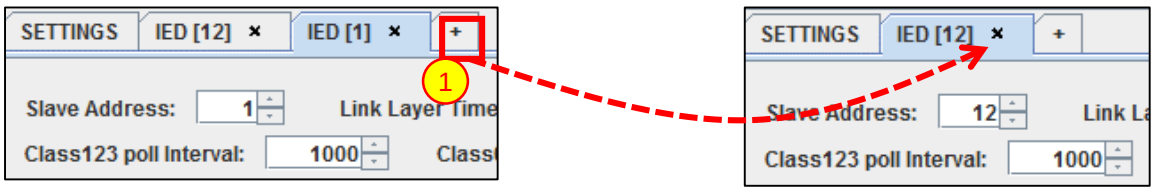


Figure 3-7 – Remove a connected server.

3.4 *Server Configuration*

A DNP3 server application outputs data from the ADH database, receives commands and passes them into the ADH system to command another application to perform the operation. Therefore, all server operations use data point *references* to already existing data points that have been created by other application clients or client-servers. Atop protocol gateway supports one server application per protocol per device.
The options describe the local server itself. The settings tab can be used to set the local settings, while the IED tab is similar to the client-tab. Example figures of both are shown below.

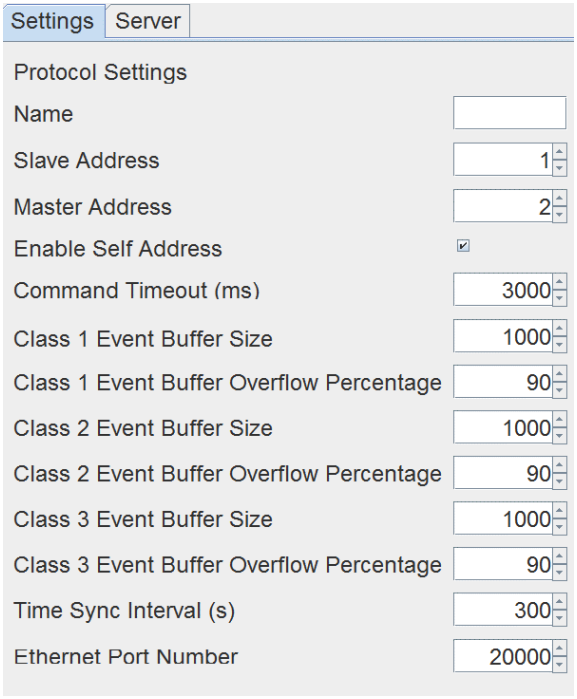


Figure 3-8 - Server settings panel extract.

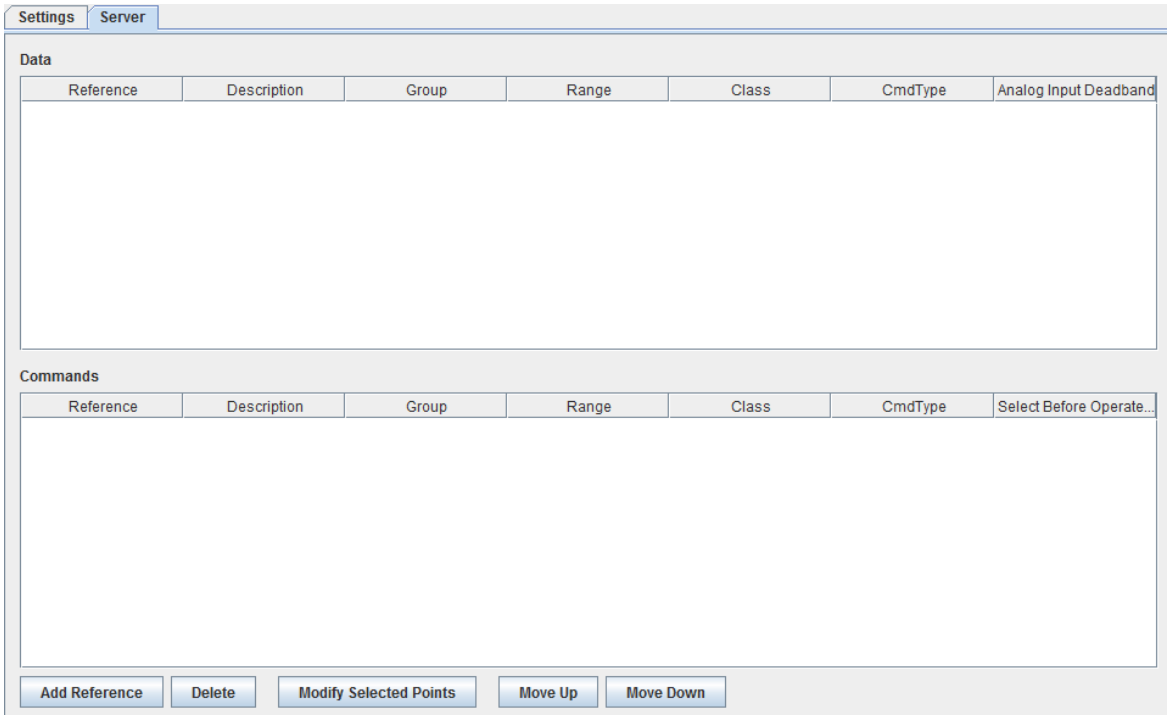


Figure 3-9 – Server IED panel.

Here the "Add Reference" button can be used to add data point references. The procedure is explained fully in the next section. The other buttons are described in section 6: Reference Guide.

3.4.1 Server Settings

All server settings are explained in the headings below.

3.4.1.1 Master Address

Description	The DNP3 master address to be used.
Data Entry	Integer
Range	*0 to 65519*
Input Option	Mandatory

3.4.1.2 Communication Medium

Description	The communication method to be used. The list will automatically be restricted based on the parent's port type.
Data Entry	Drop down menu
Options	*Serial, TCP, UDP*
Input Option	Mandatory

3.4.1.3 Enable Self Address

Description	The value of the "Enable self address" flag when the application is started. When unsolicited responses are enabled, unsolicited messages may be sent.

Data Entry	Check box
Range	☐ disabled, ☑ enabled
Input Option	Mandatory

3.4.1.4 Class {X} Event Buffer Size

Description	The buffer size for class {X}: the maximum number of events to store.
Data Entry	Integer
Range	*10 - 65535*
Input Option	Mandatory

3.4.1.5 Class {X} Event Buffer Overflow Percentage

Description	If the buffer for class {X} fills to this percentage, a buffer overflow event is sent to the master station.
Data Entry	Integer
Range	*25-100. Recommended 50-95. Default: 90*
Input Option	Mandatory

3.4.1.6 Default Static Variations

Description	The default static variations.
Data Entry	Drop down menus
Options	Binary Input: *Packed, With Flags* Double Input: *Packed, With Flags* Counter Input: *32bit with flags, 16 bit with flags, 32 bit (no flags), 16 bit (no flags)* Counter Input Frozen: *32bit with flags, 16 bit with flags, 32bit with flags and time, 16bit with flags and time, 32 bit (no flags), 16 bit (no flags)* Analog Input: *32bit with flags, 16 bit with flags, 32 bit (no flags), 16 bit (no flags), Single precision with flags and time* Analog Input Deadband: *16bit, 32bit, Single precision float* Binary Output: *Packed, With Flags* Analog Output: *32bit with flags, 16bit with flags, Single precision with flags*
Input Option	Mandatory

3.4.1.7 Default Event Variations

Description	The default event variations.
Data Entry	Drop down menus

Options	

Binary Input	*No Time, Absolute Time, Relative Time*
Double Input	*No Time, Absolute Time, Relative Time*
Counter Input	*32bit with flags, 16bit with flags, 32bit with flags and time, 16bit with flags and time*
Counter Input Frozen	*32bit with flags, 16bit with flags, 32bit with flags and time, 16bit with flags and time*
Analog Input	*32bit (no time), 16bit (no time), 32bit with time, 16bit with time, Single precision (no time), Single precision with time*

Input Option	Mandatory

3.4.2 Adding Data Point References

To add new data point references, left click the "**Add Reference**" button underneath the tables in the main view. This will bring up the Add References window defined by the eNode Designer main application. It should appear similar to the following figure. Here we are adding references to data points created by an IEC 60870-5-104 client.

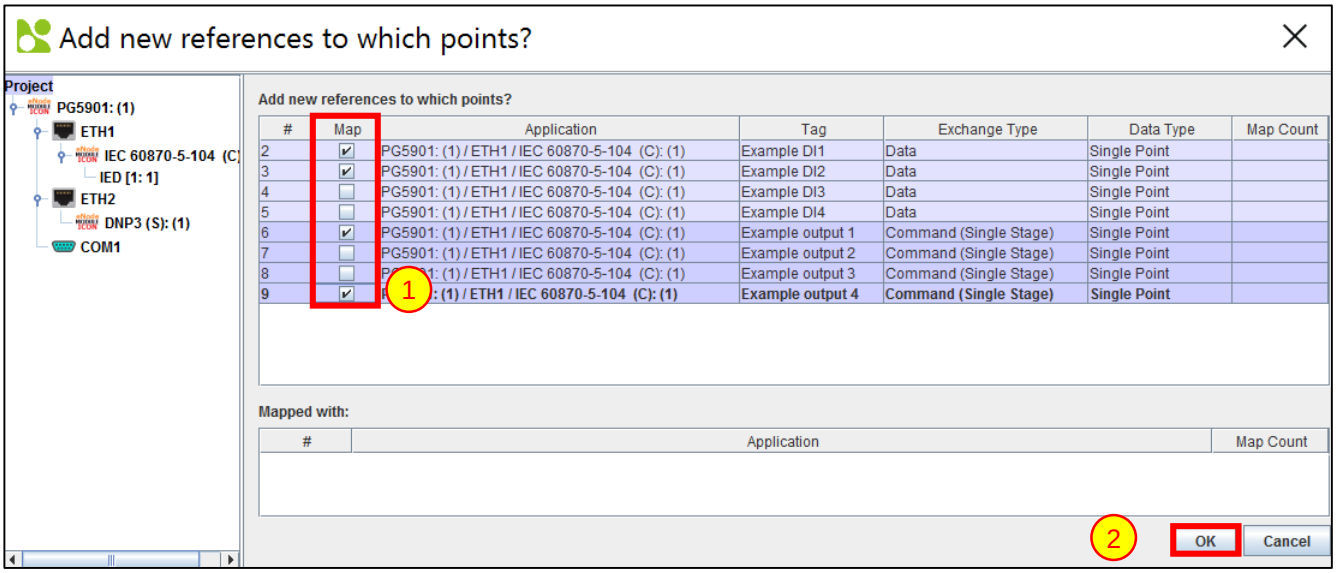


Figure 3-10 - Add new references window.

1 **Select Data Points** – Adding a reference to a point creates a "mapping" to that point. Select which data points the server application is interested in using.

2 Left Click **OK** when done to accept the new references.

The data points that will appear in the list and that will be available for mapping are those whose data point type is compatible with the DNP3 application. For the table matching DNP3 data types to ADH types, see section 0.

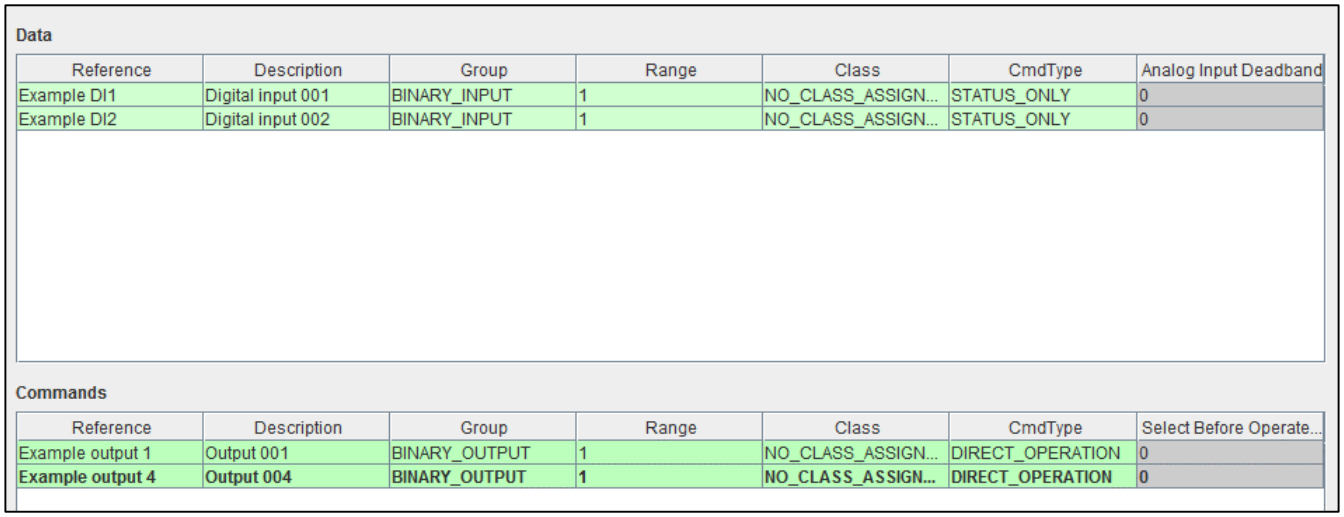


Figure 3-11 - Data point references added.

All properties will be automatically defined based on the data point's values.

3.4.3 Contiguous Server Object Addresses

The DNP3 protocol specifies that all addresses (indices) must be contiguous for each point type. For example if there are 10 binary inputs, they must have addresses 0 through 9. The data points in server's table are always automatically sorted so that each point type is listed consecutively, with incremental addresses.

For this reason, in the server, the **address values cannot be changed directly** by the user. In order to change the address of a data point, use the "Move Up" and "Move Down" buttons within the point group (Binary Input / …).

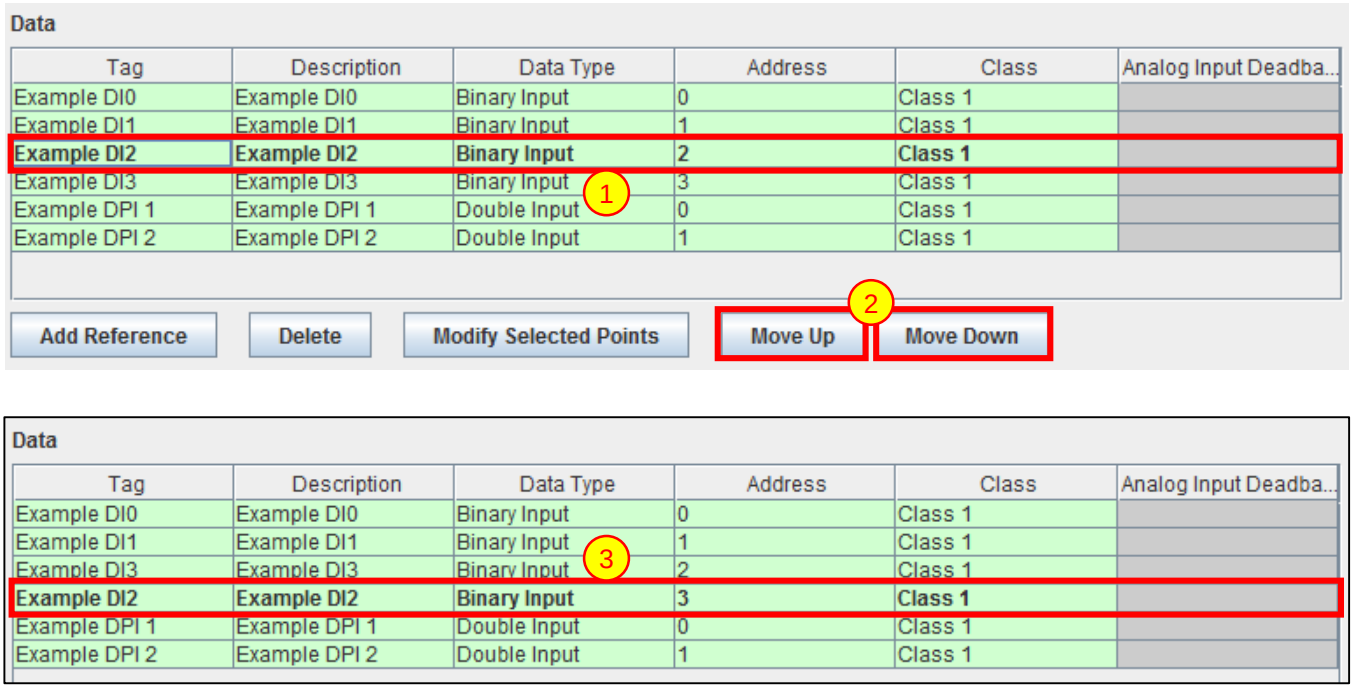


Figure 3-12 - Changing the address of a server point.

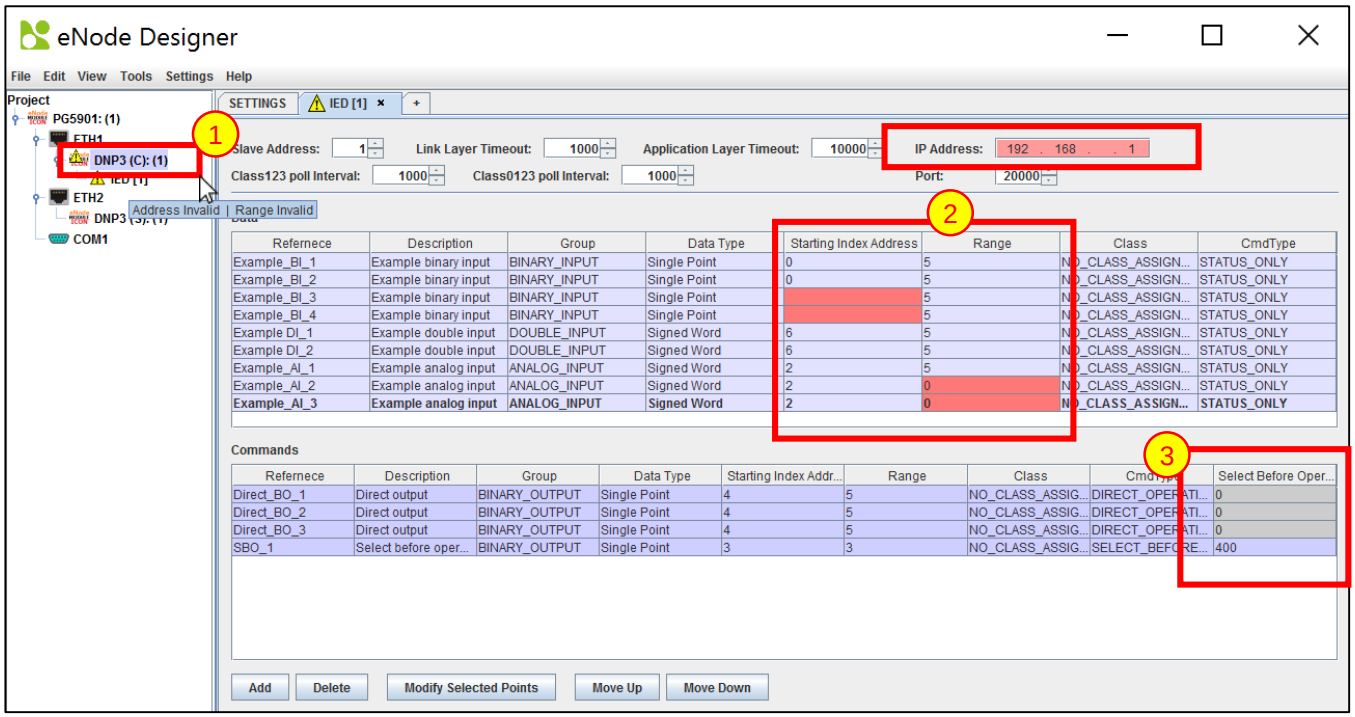
1. **Select the row(s)** you want to move (to change the address).
2. Use the **Move Up** or **Move Down** button. In this example, *Move Down* was used.
3. The row(s) have been moved and the addresses have been updated accordingly. Note that this is the same data point as selected in (1).

Move down will by nature increase addresses by one (per click) and move up will reduce addresses by one (per click).

3.5 *Miscellaneous Common*

3.5.1 Incomplete, Conflicting and not needed Information

Incomplete or conflicting information is shown in red, and will cause warning symbols on the tab and in the project tree. Hovering over the warning icons will show further details about the cause of the warning. This allows the user to quickly fix invalid information.



1. **Mouse-over a warning** to show a tooltip explaining the warning.

2. **Invalid Data** shows in red. The darker red means the data is invalid, and the lighter red means there is an address conflict.

3. **Unneeded Data** is hidden and not editable. For example, the SBO timeout is not required for direct operates. These table cells have a grey background, no contents and cannot be edited.

3.5.2 Modify Selected Points Window

The “Modify Selected Points” window is used to change many row properties in one single step.

Select the data points you want to change, and then press the “Modify Selected Points” button beneath the tables. It will generate the following window.

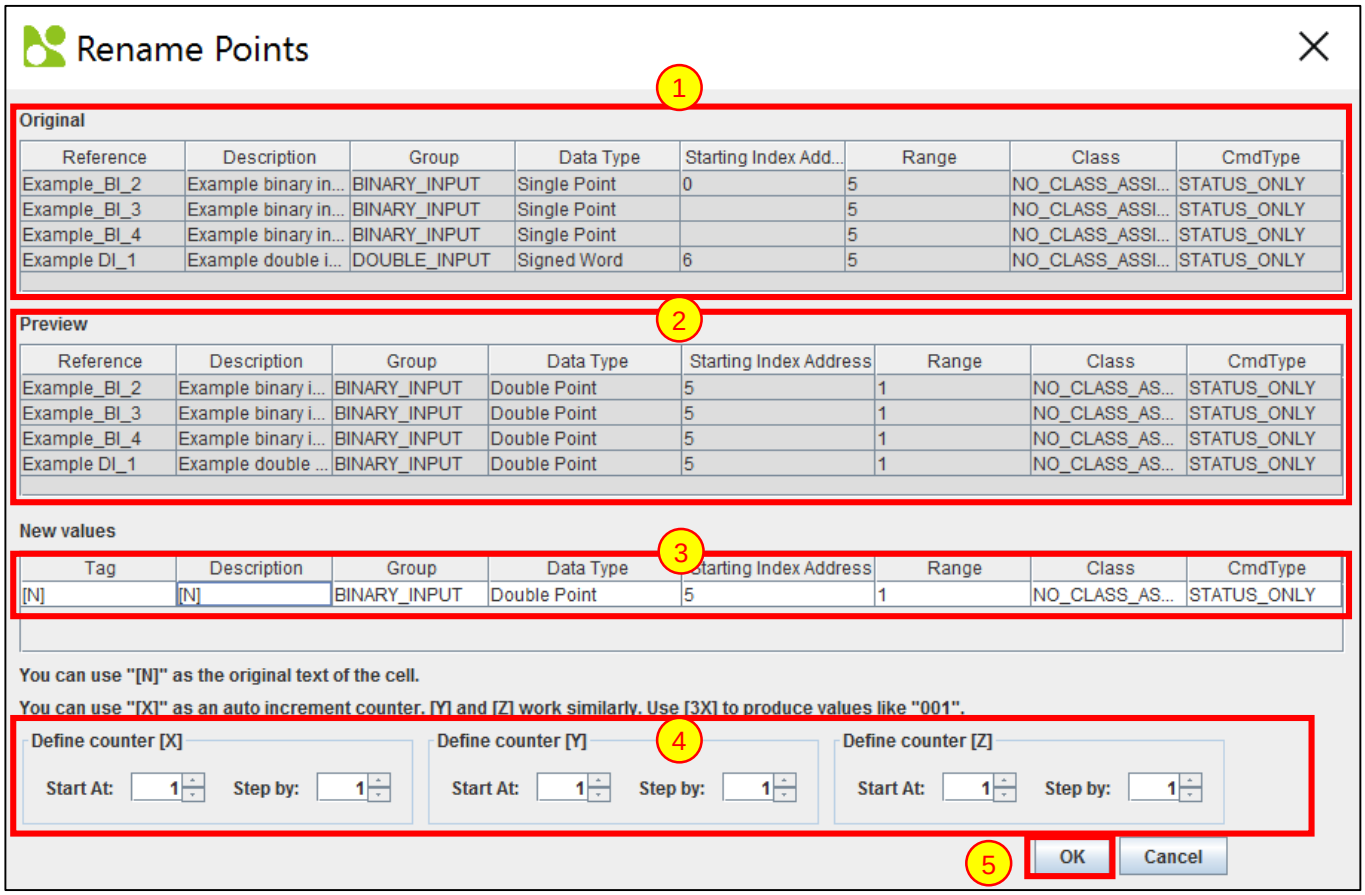


Figure 3-13 - Modify data points window example.

1. **Original data table** – Shows the original data table.

2. **Preview** – Shows the new data table that will be used if the modifications are accepted. These fields update automatically according to the contents of (3).

3. **New values –** Contain the new values for the table cells. “[N]” can be used to maintain the original value of the cell, and the auto-incrementing counters [X], [Y] and [Z] can be used to add numbers. For details, see Using Auto-increment.

4. **Counter properties** – Sets the initial values and step amounts of the counters [X], [Y] and [Z].

5. **OK button** – to accept the modifications.

Data point references always use the *tag* and *description* of the “real” point. Consequently, these values will not be changed by a server application. The object addresses in the server are required to be contiguous, so the addresses cannot be changed using this window either. See server configuration for details.

4 Communication Port Properties

The device module handles the communication port properties. The typical method is described briefly below.

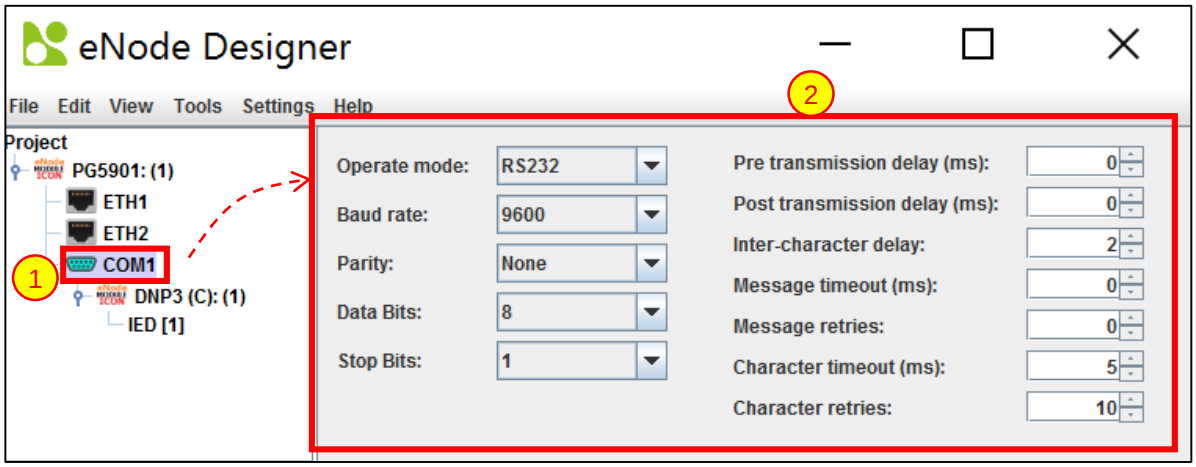


Figure 4-1 - Serial port properties.

1 **Select the communication port in the project tree** – This will typically cause the central panel to show the port's properties.

2 **Properties** – The communication port's properties can be set.

Similarly, the Ethernet properties are shown below.

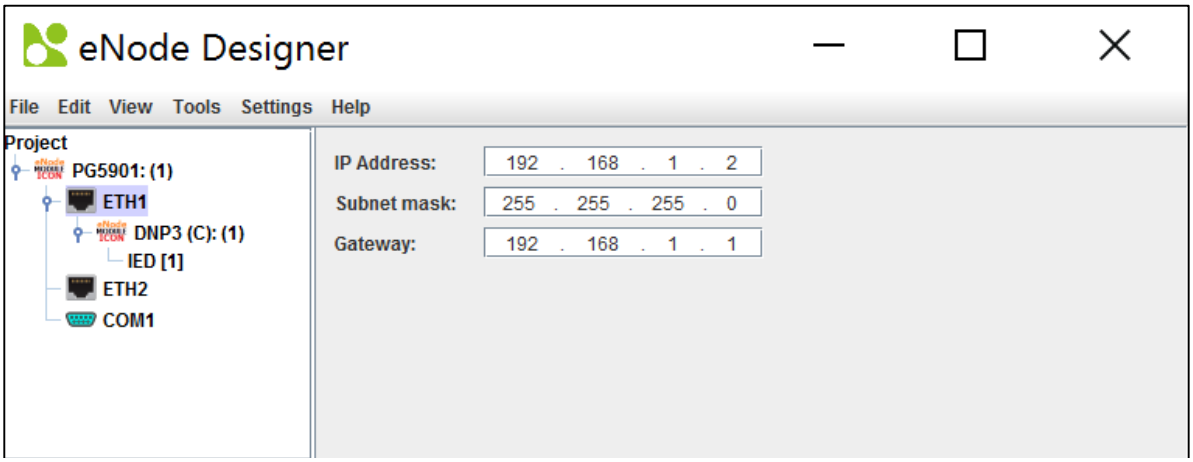


Figure 4-2 – Ethernet port properties.

5 Using Auto-increment Counters

The following is a full example showing how auto-increment works. The example given shows the IEC 60870-5-104 window, however the DNP3 auto-increment works in the same way.

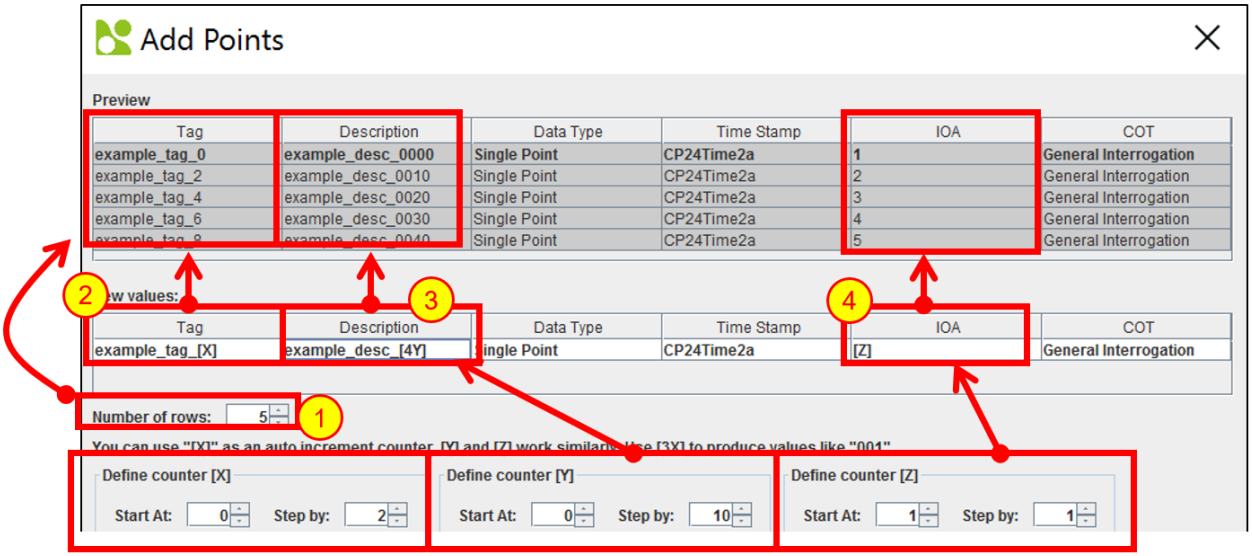


Figure 5-1 – Using Auto Increment when adding Data Points or Commands.

1 The *Number of Rows* can be modified to set the number of data points or commands created from the *New values* section. As shown in the example above, five data points/commands are created and shown in the preview section as the *Number of Rows* is set to 5.

When using the auto increment counters by default, they will start at one and increment by one. Anyway auto-increment value has its own section for configuration. Adjusting *Start At* will change the value that the first data point/command receives. Adjusting *Step By* will change the value that the second and subsequent values will be incremented by.

2 In this example, the *[X]* counter is used. The *Start At* value has been set to 0 and the *Step By* value has been set to 2. This results in the values seen in the preview section.

It is also possible to include a number within the square brackets and before the X, Y or Z while using auto increment. This will produce values that contain the entered number of digits. Any digit that is not taken up by the value determined by the *Start At* and *Step By* values will be shown as zeros.

3 In this example, the *[Y]* counter has been used with the integer 4 to indicate the number structure. This results in the values shown in the preview section.

4 In this example, the *[Z]* counter has been used. The *Start At* and *Step By* values have been left at default, this results in the values shown.

If no auto increment value is entered in any field, each data point/command field value will be created the same with the exception of *Tag* and *IOA*. The first new data point/command's *Tag* value will represent what was entered in the *New value* section. However, the subsequent data points/commands will contain the initial *Tag* value followed by an underscore and a number incrementing by one from 1 onwards. (Example: tag, tag_1, tag_2 etc.). This is an artefact of eNode Designer ensuring all data point tag names are unique.

5.1 *Automatic Increments in Constant Values*

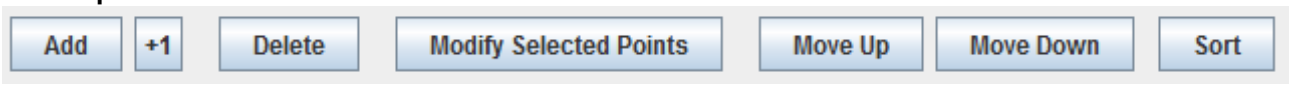
In the DNP3 eNode module, the following fields will be automatically increased by one for each row, even if a constant value is entered in the “New value” field.

- Address

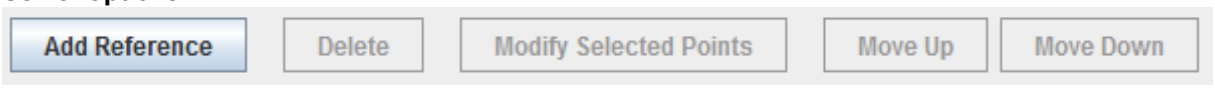
6 Reference Guide

6.1 *Table Buttons*

Client Options:



Server Options:



Add	Adds new data points in the client. See section 3.3.2.
+1	Adds a single new data point in the client. See section 3.3.2.
Add Reference	Adds a new data point reference in the server. See section 3.4.2.
Delete	Deletes the selected data points
Modify Selected Points	Modify the properties of the selected data points. See section 3.5.2.
Move Up	Moves the selected data points up one row in the table
Move Down	Moves the selected data points down one row in the table
Sort	Sorts the table: groups by data type, then by address order.

6.2 *Table Columns*

6.2.1.1 Tag

Description	A unique Tag name for each data point.
Data Entry	String
Min Length	1
Max Length	N/A
Input Option	Mandatory

6.2.1.2 Description

Description	User defined description for each data point.
Data Entry	String
Min Length	1
Max Length	N/A
Input Option	Mandatory

6.2.1.3 Groups

Description	The DNP3 point type.
Data Entry	Drop Down Menu
Groups	*Binary Input, Double Input, Counter Input, Analog Input, Binary Output, Analog Output*
Input Option	Mandatory

6.2.1.4 Start Address

Description	The DNP3 starting *index* in the DNP3 point type.
Data Entry	Integer
Start Address	*n/a*
Count	*0 to 65535 (the max Address starting from "start addr"+"count" can't exceed 65535)*
Input Option	Mandatory

6.2.1.5 Class

Server Only

Description	The DNP3 class. Using *No class* means that no events are raised for the data point.
Data Entry	Drop down menu
Options	*No class, Class 1, Class 2, Class 3*
Input Option	Mandatory

6.2.1.6 Analog Input Deadband

Server Data Only

Description	The dead-band of the Analog input. An event is triggered when the Analog input changes an amount greater than the deadband value.
Data Entry	Floating point number
Range	*0.0 or greater*
Input Option	Mandatory for Analog inputs

6.2.1.7 Command Type

Description	The command security level: direct or SBO.
Data Entry	Drop down menu
Range	*Direct Operate, Select Before Operate*
Input Option	Mandatory for commands

6.2.1.8 SBO Timeout (ms)

Server commands only

Description	The timeout to wait for a SBO (Select before operate) to complete in milliseconds.
Data Entry	Integer
Range	*0 to 65000*
Input Option	Mandatory if command type is *Select Before Operate*

6.3 *DNP3's Related ADH Types*

The DNP3 data types correspond to the ADH types given in the table below.

DNP3 Point Type	ADH Data Type	ADH Exchange Type
Binary Input	Single Point	Data
Double Input	Double Point	Data
Counter Input	Unsigned 32	Data
Analog Input	Float 32	Data
Binary Output	Single Point	Command
Analog Output	Float 32	Command

Table 6-1 – DNP3 data types relation to ADH data point types.

7 DNP3 Client Properties

7.1 *Device Profile*

This document defines the options of the DNP3 protocol used by Atop DNP3 devices and must be accompanied by an implementation table.

Vendor Name: Atop Technologies, Inc.

Device Name: PG59XX Series DNP3 Server over Ethernet or Serial

Device Function:
- ■ *Master*
- □ *Outstation*

DNP Levels Supported for:
Request and Response
- □ None
- ■ Level 1
- ■ Level 2
- □ Level 3
- □ Level 4

Supported Function Blocks:
- ■ Self-Address Support
- □ Data Sets
- □ File Transfer
- □ Virtual Terminals
- ■ Mapping to IEC61850 Object Models defined in a DNP3 XML file
- □ Function code 31, activate configuration
- □ Authentication

Connections Supported:
- ■ Serial
- ■ IP Networking

Serial Connections:

Serial Connection Parameters:
- ■ Asynchronous – 8 Data Bits, 1 Start Bit, 1 Stop Bit, No Parity

Baud Rate:
- □ fixed
- ■ Configurable – 110 to 115200

Flow Control:
- ■ None
- □ Hardware flow control
- □ Software flow control

Interval to Request Link Status:
- ■ Not supported
- □ Fixed at seconds
- □ Configurable – 0 to 2147483647

Supports DNP3 Collision Avoidance:
- ■ No
- □ Yes, using back-off time = (Min + Random) method

Receiver Inter-character Timeout:
- ■ Not checked
- □ No gap permitted
- □ Fixed a bit times
- □ Fixed a ms
- □ Configurable

Inter-character Gap in Transmission:
- ■ None
- □ Maximum bit times
- □ Maximum ms

IP Networking:

Type of End Point:
- ■ TCP Initiating
- ■ TCP Listening
- ■ UDP datagram

TCP Listen Port Number:
- □ Fixed at 20000
- ■ Configurable, range 1 to 65535 (default 20000)

TCP Keep-alive timer:
- ■ Fixed at 19000 ms
- □ Configurable, range _____ to _____ ms

Local UDP Port:
- □ Fixed at 20000
- ■ Configurable, range 1 to 65535 (default 20000)

Multiple Outstation Connections:
- □ Not supported
- ■ Supports multiple outstations (maximum is 64)

Time Synchronization Support:
- □ Not supported
- □ DNP3 LAN Procedure
- ■ DNP3 Write Time

Data Link Address:
- □ Fixed at 292
- ■ Configurable, range 0 to 65519 (default 2)

Self-Address Support using address 0xFFFC:
- ■ Yes
- □ No

Sends Confirmed User Data Frames:
- ■ Never
- □ Sometimes, explain
- □ Always

Data Link Layer Confirmation Timeout:
- ■ None
- □ Fixed at 2000 ms
- □ Configurable, range _____ to _____ ms

Maximum Data Link Retries:
- ■ Never Retries
- □ Fixed at 3
- □ Configurable, range _____ to _____

Maximum number of octets Transmitted in a Data Link Frame: ■ Fixed at 292 □ Configurable, range _____ to _____	*Maximum number of octets that can be Received in a Data Link Frame:* ■ Fixed at 292 □ Configurable, range _____ to _____
Maximum number of octets Transmitted in an Application Layer Fragment: ■ Fixed at 2048 □ Configurable, range _____ to _____	*Maximum number of octets that can be Received in an Application Layer Fragment:* ■ Fixed at 249 □ Configurable, range _____ to _____
Timeout waiting for Complete Application Layer Fragment: □ None □ Fixed at 6000 ms ■ Configurable, range _1000_ to _65535_ ms	
Control Status Codes Supported: □ 1 – TIMEOUT ■ 2 – NO_SELECT ■ 3 – FORMAT_ERROR ■ 4 – NOT_SUPPORTED ■ 5 – ALREADY_ACTIVE ■ 6 – HARDWARE_ERROR ■ 7 – LOCAL □ 8 – TOO_MANY_OBJS ■ 9 – NOT_AUTHORIZED ■ 10 – AUTOMATION_INHIBIT	 ■ 11 – PROCESSING_LIMITED ■ 12 – OUT_OF_RANGE □ 13 – DOWNSTREAM_LOCAL □ 14 – ALREADY_COMPLETE □ 15 – BLOCKED □ 16 – CANCELLED □ 17 – BLOCKED_OTHER_MASTER □ 18 – DOWNSTREAM_FAIL ■ 126 – RESERVED ■ 127 – UNDEFINED

7.2 Implementation Table

OBJECT			REQUEST (Library will parse)		RESPONSE (Library will respond with)	
Object	Variation	Description	Application Layer Function Codes (Decimal)	Qualifier Codes (hex)	Application Layer Function Codes (Decimal)	Qualifier Codes (hex)
1	0	Binary Input - All Variations (Variation 0 is used to request default variation)	1(Read), 22(Assign Class)	00, 01 (start-stop) 06 (no range, or all) 07, 08 ,(limited qty) 17, 28 (index)	129(Response)	00, 01 17, 28
1	1	Binary Input - Packed Format	1	00, 01 (start-stop) 06 (no range, or all) 07, 08 ,(limited qty) 17, 28 (index)	129,	00, 01 17, 28
1	2	Binary Input with Status	1	00, 01 (start-stop) 06 (no range, or all) 07, 08 ,(limited qty) 17, 28 (index)	129	00, 01 17, 28
2	0	Binary Input Change - All Variations (Default variation)	1	06,07,08	129	17, 28
2	1	Binary Input Change without Time	1	06,07,08	129, 130 (Unsolicited	17, 28

OBJECT			REQUEST (Library will parse)		RESPONSE (Library will respond with)	
Object	Variation	Description	Application Layer Function Codes (Decimal)	Qualifier Codes (hex)	Application Layer Function Codes (Decimal)	Qualifier Codes (hex)
					Response)	
2	2	Binary Input Change with Time	1	06,07,08	129, 130	17, 28
2	3	Binary Input Change with Relative Time	1	06,07,08	129, 130	17, 28
3	0	Double-bit Binary Input - All Variations (Variation 0 is used to request default variation)	1(Read), 22(Assign Class)	00, 01 (start-stop) 06 (no range, or all) 07, 08 ,(limited qty) 17, 28 (index)	129(Response)	00, 01 17, 28
3	1	Double-bit Binary Input – Packed Format	1	00, 01 (start-stop) 06 (no range, or all) 07, 08 ,(limited qty) 17, 28 (index)	129,	00, 01 17, 28
3	2	Double-bit Binary Input	1	00, 01 (start-stop) 06 (no range, or all) 07, 08 ,(limited qty) 17, 28 (index)	129	00, 01 17, 28
4	0	Double-bit Binary Input Change - All Variations (Default variation)	1	06,07,08	129	17, 28

OBJECT			REQUEST (Library will parse)		RESPONSE (Library will respond with)	
Object	Variation	Description	Application Layer Function Codes (Decimal)	Qualifier Codes (hex)	Application Layer Function Codes (Decimal)	Qualifier Codes (hex)
4	1	Double-bit Binary Input Change without Time	1	06,07,08	129, 130 (Unsolicited Response)	17, 28
4	2	Double-bit Binary Input Change with Time	1	06,07,08	129, 130	17, 28
4	3	Double-bit Binary Input Change with Relative Time	1	06,07,08	129, 130	17, 28
10	0	Binary Output - All Variations	1(Read), 22(Assign Class)	00, 01 (start-stop) 06 (no range, or all) 07, 08 ,(limited qty) 17, 28 (index)	129	00, 01 17, 28
10	1	Binary Output	1	00, 01 (start-stop) 06 (no range, or all) 07, 08 ,(limited qty) 17, 28 (index)	129,	00, 01 17, 28
10	2	Binary Output Status	1	00, 01 (start-stop) 06 (no range, or all) 07, 08 ,(limited qty) 17, 28 (index)	129,	00, 01 17, 28

OBJECT			REQUEST (Library will parse)		RESPONSE (Library will respond with)	
Object	Variation	Description	Application Layer Function Codes (Decimal)	Qualifier Codes (hex)	Application Layer Function Codes (Decimal)	Qualifier Codes (hex)
12	0	Control Block - All Variations				
12	1	Control Relay Output Block	3(Select), 4(Operate), 5 (Direct Operate), 6 (Direct Operate NR)	17, 28	129	echo of request
20	0	Binary Counter - All Variations	1(Read), 22(Assign Class) 7(Immediate Freeze), 8 (Immediate Freeze - No Response), 9 (Freeze and Clear), 10 (Freeze and Clear – No Response)	00, 01 (start-stop) 06 (no range, or all) 07, 08 ,(limited qty) 17, 28 (index)	129	00, 01 17, 28
20	1	32-Bit Binary Counter	1,	00, 01 (start-stop) 06 (no range, or all) 07, 08 ,(limited qty) 17, 28 (index)	129,	00, 01 17, 28
20	2	16-Bit Binary Counter	1,	00, 01 (start-stop) 06 (no range, or all) 07, 08 ,(limited qty) 17, 28 (index)	129,	00, 01 17, 28

OBJECT			REQUEST (Library will parse)		RESPONSE (Library will respond with)	
Object	Variation	Description	Application Layer Function Codes (Decimal)	Qualifier Codes (hex)	Application Layer Function Codes (Decimal)	Qualifier Codes (hex)
20	5	32-Bit Binary Counter without Flag	1	00, 01 (start-stop) 06 (no range, or all) 07, 08 ,(limited qty) 17, 28 (index)	129,	00, 01 17, 28
20	6	16-Bit Binary Counter without Flag	1	00, 01 (start-stop) 06 (no range, or all) 07, 08 ,(limited qty) 17, 28 (index)	129,	00, 01 17, 28
21	0	Frozen Counters - All Variations	1(Read), 22(Assign Class)	00, 01 (start-stop) 06 (no range, or all) 07, 08 ,(limited qty) 17, 28 (index)	129(Response)	00, 01 17, 28
21	1	32-Bit Frozen Counter	1	00, 01 (start-stop) 06 (no range, or all) 07, 08 ,(limited qty)	129,	00, 01 17, 28

OBJECT			REQUEST (Library will parse)		RESPONSE (Library will respond with)	
Object	Variation	Description	Application Layer Function Codes (Decimal)	Qualifier Codes (hex)	Application Layer Function Codes (Decimal)	Qualifier Codes (hex)
				17, 28 (index)		
21	2	16-Bit Frozen Counter	1	00, 01 (start-stop) 06 (no range, or all) 07, 08 ,(limited qty) 17, 28 (index)	129,	00, 01 17, 28
21	5	32-Bit Frozen Counter with Time of Freeze	1	00, 01 (start-stop) 06 (no range, or all) 07, 08 ,(limited qty) 17, 28 (index)	129,	00, 01 17, 28
21	6	16-Bit Frozen Counter with Time of Freeze	1	00, 01 (start-stop) 06 (no range, or all) 07, 08 ,(limited qty) 17, 28 (index)	129,	00, 01 17, 28
21	9	32-Bit Frozen Counter without Flag	1	00, 01 (start-stop) 06 (no range, or all)	129,	00, 01 17, 28

OBJECT			REQUEST (Library will parse)		RESPONSE (Library will respond with)	
Object	Variation	Description	Application Layer Function Codes (Decimal)	Qualifier Codes (hex)	Application Layer Function Codes (Decimal)	Qualifier Codes (hex)
				07, 08 ,(limited qty) 17, 28 (index)		
21	10	16-Bit Frozen Counter without Flag	1	00, 01 (start-stop) 06 (no range, or all) 07, 08 ,(limited qty) 17, 28 (index)	129,	00, 01 17, 28
22	0	Counter Change Event - All Variations	1	06,07,08	129	17, 28
22	1	32-Bit Counter Change Event without Time	1	06,07,08	129, 130	17, 28
22	2	16-Bit Counter Change Event without Time	1	06,07,08	129, 130	17, 28
22	5	32-Bit Counter Change Event with Time	1	06,07,08	129, 130	17, 28
22	6	16-Bit Counter Change Event with Time	1	06,07,08	129, 130	17, 28
23	0	Frozen Counter Events - All Variations	1	06,07,08	129	17, 28
23	1	32-Bit Frozen Counter Event without Time	1	06,07,08	129, 130	17, 28

OBJECT			REQUEST (Library will parse)		RESPONSE (Library will respond with)	
Object	Variation	Description	Application Layer Function Codes (Decimal)	Qualifier Codes (hex)	Application Layer Function Codes (Decimal)	Qualifier Codes (hex)
23	2	16-Bit Frozen Counter Event without Time	1	06,07,08	129, 130	17, 28
23	5	32-Bit Frozen Counter Event with Time	1	06,07,08	129, 130	17, 28
23	6	16-Bit Frozen Counter Event with Time	1	06,07,08	129, 130	17, 28
30	0	Analog Input - All Variations	1(Read), 22(Assign Class) 7(Immediate Freeze), 8 (Immediate Freeze - No Response), 9 (Freeze and Clear), 10 (Freeze and Clear – No Response)	00, 01 (start-stop) 06 (no range, or all) 07, 08 ,(limited qty) 17, 28 (index)	129(Response)	00, 01 17, 28
30	1	32-Bit Analog Input	1	00, 01 (start-stop) 06 (no range, or all) 07, 08 ,(limited qty) 17, 28 (index)	129,	00, 01 17, 28
30	2	16-Bit Analog Input	1	00, 01 (start-stop) 06 (no range, or all)	129,	00, 01 17, 28

OBJECT			REQUEST (Library will parse)		RESPONSE (Library will respond with)	
Object	Variation	Description	Application Layer Function Codes (Decimal)	Qualifier Codes (hex)	Application Layer Function Codes (Decimal)	Qualifier Codes (hex)
				07, 08 ,(limited qty) 17, 28 (index)		
30	3	32-Bit Analog Input without flag	1	00, 01 (start-stop) 06 (no range, or all) 07, 08 ,(limited qty) 17, 28 (index)	129,	00, 01 17, 28
30	4	16-Bit Analog Input without flag	1	00, 01 (start-stop) 06 (no range, or all) 07, 08 ,(limited qty) 17, 28 (index)	129,	00, 01, 17, 28
30	5	Single-precision float –point with flag	1	00, 01 (start-stop) 06 (no range, or all) 07, 08 ,(limited qty) 17, 28 (index)	129,	00, 01, 17, 28
31	0	Frozen Analog Input - All Variations	1(Read), 22(Assign Class)	00, 01 (start-stop)	129(Response)	00, 01,

OBJECT			REQUEST (Library will parse)		RESPONSE (Library will respond with)	
Object	Variation	Description	Application Layer Function Codes (Decimal)	Qualifier Codes (hex)	Application Layer Function Codes (Decimal)	Qualifier Codes (hex)
				06 (no range, or all) 07, 08 ,(limited qty) 17, 28 (index)		17, 28
31	1	32-Bit Frozen Analog Input	1	00, 01 (start-stop) 06 (no range, or all) 07, 08 ,(limited qty) 17, 28 (index)	129,	00, 01, 17, 28
31	2	16-Bit Frozen Analog Input	1	00, 01 (start-stop) 06 (no range, or all) 07, 08 ,(limited qty) 17, 28 (index)	129,	00, 01, 17, 28
31	3	32-Bit Frozen Analog Input with Time of Freeze	1	00, 01 (start-stop) 06 (no range, or all) 07, 08 ,(limited qty) 17, 28 (index)	129,	00, 01, 17, 28
31	4	16-Bit Frozen Analog Input with Time of	1		129,	

OBJECT			REQUEST (Library will parse)		RESPONSE (Library will respond with)	
Object	Variation	Description	Application Layer Function Codes (Decimal)	Qualifier Codes (hex)	Application Layer Function Codes (Decimal)	Qualifier Codes (hex)
		Freeze		00, 01 (start-stop) 06 (no range, or all) 07, 08 ,(limited qty) 17, 28 (index)		00, 01, 17, 28
31	5	32-Bit Frozen Analog Input without Flag	1	00, 01 (start-stop) 06 (no range, or all) 07, 08 ,(limited qty) 17, 28 (index)	129,	00, 01, 17, 28
31	6	16-Bit Frozen Analog Input without Flag	1	00, 01 (start-stop) 06 (no range, or all) 07, 08 ,(limited qty) 17, 28 (index)	129,	00, 01, 17, 28
31	7	Single-precision float –point with flag	1	00, 01 (start-stop) 06 (no range, or all) 07, 08 ,(limited qty) 17, 28 (index)	129,	00, 01, 17, 28

OBJECT			REQUEST (Library will parse)		RESPONSE (Library will respond with)	
Object	Variation	Description	Application Layer Function Codes (Decimal)	Qualifier Codes (hex)	Application Layer Function Codes (Decimal)	Qualifier Codes (hex)
32	0	Analog Change Event - All Variations	1	06,07,08	129	17, 28
32	1	32-Bit Analog Change Event without Time	1	06,07,08	129, 130	17, 28
32	2	16-Bit Analog Change Event without Time	1	06,07,08	129, 130	17, 28
32	3	32-Bit Analog Change Event with Time	1	06,07,08	129, 130	17, 28
32	4	16-Bit Analog Change Event with Time	1	06,07,08	129, 130	17, 28
32	5	Single-precision float –point Analog Change Event with out Time	1	06,07,08	129, 130	17, 28
32	7	Single-precision float –point Analog Change Event with Time	1	06,07,08	129, 130	17, 28
33	0	Frozen Analog Event - All Variations	1	06,07,08	129	17, 28
33	1	32-Bit Frozen Analog Event without Time	1	06,07,08	129, 130	17, 28

OBJECT			REQUEST (Library will parse)		RESPONSE (Library will respond with)	
Object	Variation	Description	Application Layer Function Codes (Decimal)	Qualifier Codes (hex)	Application Layer Function Codes (Decimal)	Qualifier Codes (hex)
33	2	16-Bit Frozen Analog Event without Time	1	06,07,08	129, 130	17, 28
33	3	32-Bit Frozen Analog Event with Time	1	06,07,08	129, 130	17, 28
33	4	16-Bit Frozen Analog Event with Time	1	06,07,08	129, 130	17, 28
33	5	Single-precision float –point Frozen Analog Change Event with out Time	1	06,07,08	129, 130	17, 28
33	7	Single-precision float –point Frozen Analog Change Event with Time	1	06,07,08	129, 130	17, 28
40	0	Analog Output Status - All Variations	1(Read), 22(Assign Class)	00, 01 (start-stop) 06 (no range, or all) 07, 08 ,(limited qty) 17, 28 (index)	129(Response)	00, 01, 17, 28
40	1	32-Bit Analog Output Status	1	00, 01 (start-stop) 06 (no range, or all) 07, 08 ,(limited qty)	129,	00, 01, 17, 28

OBJECT			REQUEST (Library will parse)		RESPONSE (Library will respond with)	
Object	**Variation**	**Description**	**Application Layer Function Codes (Decimal)**	**Qualifier Codes (hex)**	**Application Layer Function Codes (Decimal)**	**Qualifier Codes (hex)**
				17, 28 (index)		
40	2	16-Bit Analog Output Status	1	00, 01 (start-stop) 06 (no range, or all) 07, 08 ,(limited qty) 17, 28 (index)	129,	00, 01, 17, 28
40	3	Single-precision float –point Analog Output Status	1	00, 01 (start-stop) 06 (no range, or all) 07, 08 ,(limited qty) 17, 28 (index)	129,	00, 01, 17, 28
41	1	32-Bit Analog Output Block	3, 4, 5, 6	17, 28	129	echo of request
41	2	16-Bit Analog Output Block	3, 4, 5, 6	17, 28	129	echo of request
41	3	Analog Output – Single-precision float – point	3, 4, 5, 6	17, 28	129	echo of request
50	1	Time and Date	2(Write)	07 (Quantity = 1)	129	07 (quantity = 1)

OBJECT			REQUEST (Library will parse)		RESPONSE (Library will respond with)	
Object	Variation	Description	Application Layer Function Codes (Decimal)	Qualifier Codes (hex)	Application Layer Function Codes (Decimal)	Qualifier Codes (hex)
60	1	Class 0 Data	1	06	129	
60	2	Class 1 Data	1	06,07,08	129	
			20 (Enable Unsolicited), 21,(Disable Unsolicited)	06		
60	3	Class 2 Data	1	06,07,08	129	
			20 (Enable Unsolicited), 21,(Disable Unsolicited)	06		
60	4	Class 3 Data	1	06,07,08	129	
			20 (Enable Unsolicited), 21,(Disable Unsolicited)	06		
80	1	Internal Indications		00 index=7		
			2(write)			

8 DNP3 Server Properties

8.1 *Device Profile*

DNP3 Device Profile

This document defines the options of the DNP3 protocol used by Atop DNP3 devices and is accompanied by an implementation table.

Vendor Name:
Atop Technologies, Inc.

Device Name:
PG59XX Series DNP3 Server over Ethernet or Serial

Device Function:
□ Master
■ Outstation

DNP Levels Supported for:	*Supported Function Blocks:*
Request and Response □ None ■ Level 1 ■ Level 2 □ Level 3 □ Level 4	■ Self-Address Support □ Data Sets □ File Transfer □ Virtual Terminals □ Mapping to IEC61850 Object Models defined in a DNP3 XML file □ Function code 31, activate configuration □ Authentication

Connections Supported:
■ Serial
■ IP Networking

Serial Connections:
Serial Connection Parameters:
■ Asynchronous – 8 Data Bits, 1 Start Bit, 1 Stop Bit, No Parity

Baud Rate:
□ fixed
■ Configurable – 110 to 115200

Flow Control:
■ None
□ Hardware flow control
□ Software flow control

Interval to Request Link Status:
■ Not supported
□ Fixed at seconds
□ Configurable – 0 to 2147483647

Supports DNP3 Collision Avoidance:
■ No

□ Yes, using back-off time = (Min + Random) method

Receiver Inter-character Timeout:
- □ Not checked
- □ No gap permitted
- □ Fixed a bit times
- □ Fixed a ms
- ■ Configurable– 0 to 60000

Inter-character Gap in Transmission:
- ■ None
- □ Maximum bit times
- □ Maximum ms

IP Networking:

Type of End Point:
- ■ TCP Listening
- ■ UDP datagram

TCP Listen Port Number:
- □ Fixed at 20000
- ■ Configurable, range 1 to 65535 (default 20000)

TCP Keep-alive timer:
- ■ Fixed at 19000 ms
- □ Configurable, range _____ to _____ ms

Local UDP Port:
- □ Fixed at 20000
- ■ Configurable, range 1 to 65535 (default 20000)

Multiple Master Connections:
- □ Not supported
- ■ Supports multiple masters (maximum is 5)

Time Synchronization Support:
- □ Not supported
- □ DNP3 LAN Procedure
- ■ DNP3 Write Time

Data Link Address: □ Fixed at ■ Configurable, range 0 to 65519 (default 1)	Self-Address Support using address 0xFFFC: ■ Yes □ No
Sends Confirmed User Data Frames: ■ Never □ Sometimes, explain □ Always	Data Link Layer Confirmation Timeout: ■ None □ Fixed at 2000 ms □ Configurable, range _____ to _____ ms

Maximum Data Link Retries:
- ■ Never Retries
- □ Fixed at 3
- □ Configurable, range _____ to _____

Maximum number of octets Transmitted in a Data Link Frame: ■ Fixed at 292 □ Configurable, range _____ to _____	Maximum number of octets that can be Received in a Data Link Frame: ■ Fixed at 292 □ Configurable, range _____ to _____

Maximum number of octets Transmitted in an Application Layer Fragment: ■ Fixed at 2048 □ Configurable, range _____ to _____	Maximum number of octets that can be Received in an Application Layer Fragment: ■ Fixed at 249 □ Configurable, range _____ to _____
Timeout waiting for Complete Application Layer Fragment: □ None ■ Fixed at 6000 ms □ Configurable, range _____ to _____ ms	
Timeout waiting for Application Confirm of solicited response message: ■ None □ Fixed at 6000 ms □ Configurable, range 0 to 2147483647 ms (default 10000)	
Requests Application Confirmation for event response and non-final fragments: ■ Yes □ No □ Configurable	
Sends Multi-Fragment Responses: ■ *Yes* □ *No*	Last Fragment Confirmation: □ *Always* ■ *Sometimes, Only when it contains events* □ *Never*
Maximum number of objects allowed in a single control request for CROB (group 12): ■ *Fixed at 16* □ *Configurable, range _____ to _____*	
Maximum number of objects allowed in a single control request for Analog Outputs (group 41): ■ *Fixed at 16* □ *Configurable, range _____ to _____*	
Control Status Codes Supported: □ 1 – TIMEOUT ■ 2 – NO_SELECT ■ 3 – FORMAT_ERROR ■ 4 – NOT_SUPPORTED ■ 5 – ALREADY_ACTIVE ■ 6 – HARDWARE_ERROR ■ 7 – LOCAL □ 8 – TOO_MANY_OBJS ■ 9 – NOT_AUTHORIZED ■ 10 – AUTOMATION_INHIBIT	 ■ 11 – PROCESSING_LIMITED ■ 12 – OUT_OF_RANGE □ 13 – DOWNSTREAM_LOCAL □ 14 – ALREADY_COMPLETE □ 15 – BLOCKED □ 16 – CANCELLED □ 17 – BLOCKED_OTHER_MASTER □ 18 – DOWNSTREAM_FAIL ■ 126 – RESERVED ■ 127 – UNDEFINED
Supports Unsolicited Reporting: ■ *Not Supported* □ *Configurable, selectable from On and Off*	
Unsolicited Response Confirmation Timeout: □ Fixed at _____ ms □ Configurable, range 0 to 4294967295 ms	Number of Unsolicited Retries: □ Fixed at 5 □ Configurable, range _____ to _____
Event Buffer Overflow Behavior: ■ Discard the oldest event □ Discard the newest event	

□ Other, explain _____
Event Buffer Organization: □ *Per Object* ■ *Per Class* *Class 1:* □ *Fixed at* _____ ■ *Configurable, range 50 to 65535* *Class 2:* □ *Fixed at* _____ ■ *Configurable, range 50 to 65535* *Class 3:* □ *Fixed at* _____ ■ *Configurable, range 50 to 65535* □ *Single Buffer* □ *Fixed at* □ *Configurable, range* _____ *to* _____
Outstation Unsolicited Response Trigger Conditions: (Number of events) class 1: ■ Not used to trigger Unsolicited Response □ Fixed at □ Configurable class 2: ■ Not used to trigger Unsolicited Response □ Fixed at □ Configurable class 3: ■ Not used to trigger Unsolicited Response □ Fixed at □ Configurable

8.2 Implementation Table

OBJECT			REQUEST (Library will parse)		RESPONSE (Library will respond with)	
Object	Variation	Description	Application Layer Function Codes (Decimal)	Qualifier Codes (hex)	Application Layer Function Codes (Decimal)	Qualifier Codes (hex)
1	0	Binary Input - All Variations (Variation 0 is used to request default variation)	1(Read), 22(Assign Class)	00, 01 (start-stop) 06 (no range, or all) 07, 08 ,(limited qty) 17, 28 (index)	129(Response)	00, 01 17, 28
1	1	Binary Input - Packed Format	1	00, 01 (start-stop) 06 (no range, or all) 07, 08 ,(limited qty) 17, 28 (index)	129,	00, 01 17, 28
1	2	Binary Input with Status	1	00, 01 (start-stop) 06 (no range, or all) 07, 08 ,(limited qty) 17, 28 (index)	129	00, 01 17, 28
2	0	Binary Input Change - All Variations (Default variation)	1	06,07,08	129	17, 2

OBJECT			REQUEST (Library will parse)		RESPONSE (Library will respond with)	
Object	Variation	Description	Application Layer Function Codes (Decimal)	Qualifier Codes (hex)	Application Layer Function Codes (Decimal)	Qualifier Codes (hex)
2	1	Binary Input Change without Time	1	06,07,08	129, 130 (Unsolicited Response)	17, 28
2	2	Binary Input Change with Time	1	06,07,08	129, 130	17, 28
2	3	Binary Input Change with Relative Time	1	06,07,08	129, 130	17, 28
3	0	Double-bit Binary Input - All Variations (Variation 0 is used to request default variation)	1(Read), 22(Assign Class)	00, 01 (start-stop) 06 (no range, or all) 07, 08 ,(limited qty) 17, 28 (index)	129(Response)	00, 01 17, 28
3	1	Double-bit Binary Input – Packed Format	1	00, 01 (start-stop) 06 (no range, or all) 07, 08 ,(limited qty) 17, 28 (index)	129,	00, 01 17, 28
3	2	Double-bit Binary Input	1	00, 01 (start-stop) 06 (no range, or all) 07, 08 ,(limited qty) 17, 28 (index)	129	00, 01 17, 28
4	0	Double-bit Binary Input Change - All Variations (Default variation)	1	06,07,08	129	17, 28

OBJECT			REQUEST (Library will parse)		RESPONSE (Library will respond with)	
Object	**Variation**	**Description**	**Application Layer Function Codes (Decimal)**	**Qualifier Codes (hex)**	**Application Layer Function Codes (Decimal)**	**Qualifier Codes (hex)**
4	1	Double-bit Binary Input Change without Time	1	06,07,08	129, 130 (Unsolicited Response)	17, 28
4	2	Double-bit Binary Input Change with Time	1	06,07,08	129, 130	17, 28
4	3	Double-bit Binary Input Change with Relative Time	1	06,07,08	129, 130	17, 28
10	0	Binary Output - All Variations	1(Read), 22(Assign Class)	00, 01 (start-stop) 06 (no range, or all) 07, 08 ,(limited qty) 17, 28 (index)	129	00, 01 17, 28
10	1	Binary Output	1	00, 01 (start-stop) 06 (no range, or all) 07, 08 ,(limited qty) 17, 28 (index)	129,	00, 01 17, 28
10	2	Binary Output Status	1	00, 01 (start-stop) 06 (no range, or all) 07, 08 ,(limited qty) 17, 28 (index)	129,	00, 01 17, 28

OBJECT			REQUEST (Library will parse)		RESPONSE (Library will respond with)	
Object	**Variation**	**Description**	**Application Layer Function Codes (Decimal)**	**Qualifier Codes (hex)**	**Application Layer Function Codes (Decimal)**	**Qualifier Codes (hex)**
12	0	Control Block - All Variations				
12	1	Control Relay Output Block	3(Select), 4(Operate), 5 (Direct Operate), 6 (Direct Operate NR)	17, 28	129	echo of request
20	0	Binary Counter - All Variations	1(Read), 22(Assign Class) 7(Immediate Freeze), 8 (Immediate Freeze - No Response), 9 (Freeze and Clear), 10 (Freeze and Clear – No Response(	00, 01 (start-stop) 06 (no range, or all) 07, 08 ,(limited qty) 17, 28 (index)	129	00, 01 17, 28
20	1	32-Bit Binary Counter	1,	00, 01 (start-stop) 06 (no range, or all) 07, 08 ,(limited qty) 17, 28 (index)	129,	00, 01 17, 28
20	2	16-Bit Binary Counter	1,	00, 01 (start-stop) 06 (no range, or all) 07, 08 ,(limited qty) 17, 28 (index)	129,	00, 01 17, 28

OBJECT			REQUEST (Library will parse)		RESPONSE (Library will respond with)	
Object	Variation	Description	Application Layer Function Codes (Decimal)	Qualifier Codes (hex)	Application Layer Function Codes (Decimal)	Qualifier Codes (hex)
20	5	32-Bit Binary Counter without Flag	1	00, 01 (start-stop) 06 (no range, or all) 07, 08 ,(limited qty) 17, 28 (index)	129,	00, 01 17, 28
20	6	16-Bit Binary Counter without Flag	1	00, 01 (start-stop) 06 (no range, or all) 07, 08 ,(limited qty) 17, 28 (index)	129,	00, 01 17, 28
21	0	Frozen Counters - All Variations	1(Read), 22(Assign Class)	00, 01 (start-stop) 06 (no range, or all) 07, 08 ,(limited qty) 17, 28 (index)	129(Response)	00, 01 17, 28
21	1	32-Bit Frozen Counter	1	00, 01 (start-stop) 06 (no range, or all) 07, 08 ,(limited qty) 17, 28 (index)	129,	00, 01 17, 28

OBJECT			REQUEST (Library will parse)		RESPONSE (Library will respond with)	
Object	**Variation**	**Description**	**Application Layer Function Codes (Decimal)**	**Qualifier Codes (hex)**	**Application Layer Function Codes (Decimal)**	**Qualifier Codes (hex)**
21	2	16-Bit Frozen Counter	1	00, 01 (start-stop) 06 (no range, or all) 07, 08 ,(limited qty) 17, 28 (index)	129,	00, 01 17, 28
21	5	32-Bit Frozen Counter with Time of Freeze	1	00, 01 (start-stop) 06 (no range, or all) 07, 08 ,(limited qty) 17, 28 (index)	129,	00, 01 17, 28
21	6	16-Bit Frozen Counter with Time of Freeze	1	00, 01 (start-stop) 06 (no range, or all) 07, 08 ,(limited qty) 17, 28 (index)	129,	00, 01 17, 28
21	9	32-Bit Frozen Counter without Flag	1	00, 01 (start-stop) 06 (no range, or all) 07, 08 ,(limited qty) 17, 28 (index)	129,	00, 01 17, 28

OBJECT			REQUEST (Library will parse)		RESPONSE (Library will respond with)	
Object	**Variation**	**Description**	**Application Layer Function Codes (Decimal)**	**Qualifier Codes (hex)**	**Application Layer Function Codes (Decimal)**	**Qualifier Codes (hex)**
21	10	16-Bit Frozen Counter without Flag	1	00, 01 (start-stop) 06 (no range, or all) 07, 08 ,(limited qty) 17, 28 (index)	129,	00, 01 17, 28
22	0	Counter Change Event - All Variations	1	06,07,08	129	17, 28
22	1	32-Bit Counter Change Event without Time	1	06,07,08	129, 130	17, 28
22	2	16-Bit Counter Change Event without Time	1	06,07,08	129, 130	17, 28
22	5	32-Bit Counter Change Event with Time	1	06,07,08	129, 130	17, 28
22	6	16-Bit Counter Change Event with Time	1	06,07,08	129, 130	17, 28
23	0	Frozen Counter Events - All Variations	1	06,07,08	129	17, 28
23	1	32-Bit Frozen Counter Event without Time	1	06,07,08	129, 130	17, 28
23	2	16-Bit Frozen Counter Event without Time	1	06,07,08	129, 130	17, 28
23	5	32-Bit Frozen Counter Event with Time	1	06,07,08	129, 130	17, 28

OBJECT			REQUEST (Library will parse)		RESPONSE (Library will respond with)	
Object	**Variation**	**Description**	**Application Layer Function Codes (Decimal)**	**Qualifier Codes (hex)**	**Application Layer Function Codes (Decimal)**	**Qualifier Codes (hex)**
23	6	16-Bit Frozen Counter Event with Time	1	06,07,08	129, 130	17, 28
30	0	Analog Input - All Variations	1(Read), 22(Assign Class) 7(Immediate Freeze), 8 (Immediate Freeze - No Response), 9 (Freeze and Clear), 10 (Freeze and Clear – No Response)	00, 01 (start-stop) 06 (no range, or all) 07, 08 ,(limited qty) 17, 28 (index)	129(Response)	00, 01 17, 28
30	1	32-Bit Analog Input	1	00, 01 (start-stop) 06 (no range, or all) 07, 08 ,(limited qty) 17, 28 (index)	129,	00, 01 17, 28
30	2	16-Bit Analog Input	1	00, 01 (start-stop) 06 (no range, or all) 07, 08 ,(limited qty) 17, 28 (index)	129,	00, 01 17, 28

OBJECT			REQUEST (Library will parse)		RESPONSE (Library will respond with)	
Object	**Variation**	**Description**	**Application Layer Function Codes (Decimal)**	**Qualifier Codes (hex)**	**Application Layer Function Codes (Decimal)**	**Qualifier Codes (hex)**
30	3	32-Bit Analog Input without flag	1	00, 01 (start-stop) 06 (no range, or all) 07, 08 ,(limited qty) 17, 28 (index)	129,	00, 01 17, 28
30	4	16-Bit Analog Input without flag	1	00, 01 (start-stop) 06 (no range, or all) 07, 08 ,(limited qty) 17, 28 (index)	129,	00, 01, 17, 28
30	5	Single-precision float –point with flag	1	00, 01 (start-stop) 06 (no range, or all) 07, 08 ,(limited qty) 17, 28 (index)	129,	00, 01, 17, 28
31	0	Frozen Analog Input - All Variations	1(Read), 22(Assign Class)	00, 01 (start-stop) 06 (no range, or all) 07, 08 ,(limited qty) 17, 28 (index)	129(Response)	00, 01, 17, 28

OBJECT			REQUEST (Library will parse)		RESPONSE (Library will respond with)	
Object	**Variation**	**Description**	**Application Layer Function Codes (Decimal)**	**Qualifier Codes (hex)**	**Application Layer Function Codes (Decimal)**	**Qualifier Codes (hex)**
31	1	32-Bit Frozen Analog Input	1	00, 01 (start-stop) 06 (no range, or all) 07, 08 ,(limited qty) 17, 28 (index)	129,	00, 01, 17, 28
31	2	16-Bit Frozen Analog Input	1	00, 01 (start-stop) 06 (no range, or all) 07, 08 ,(limited qty) 17, 28 (index)	129,	00, 01, 17, 28
31	3	32-Bit Frozen Analog Input with Time of Freeze	1	00, 01 (start-stop) 06 (no range, or all) 07, 08 ,(limited qty) 17, 28 (index)	129,	00, 01, 17, 28
31	4	16-Bit Frozen Analog Input with Time of Freeze	1	00, 01 (start-stop) 06 (no range, or all) 07, 08 ,(limited qty) 17, 28 (index)	129,	00, 01, 17, 28

OBJECT			REQUEST (Library will parse)		RESPONSE (Library will respond with)	
Object	**Variation**	**Description**	**Application Layer Function Codes (Decimal)**	**Qualifier Codes (hex)**	**Application Layer Function Codes (Decimal)**	**Qualifier Codes (hex)**
31	5	32-Bit Frozen Analog Input without Flag	1	00, 01 (start-stop) 06 (no range, or all) 07, 08 ,(limited qty) 17, 28 (index)	129,	00, 01, 17, 28
31	6	16-Bit Frozen Analog Input without Flag	1	00, 01 (start-stop) 06 (no range, or all) 07, 08 ,(limited qty) 17, 28 (index)	129,	00, 01, 17, 28
31	7	Single-precision float –point with flag	1	00, 01 (start-stop) 06 (no range, or all) 07, 08 ,(limited qty) 17, 28 (index)	129,	00, 01, 17, 28
32	0	Analog Change Event - All Variations	1	06,07,08	129	17, 28
32	1	32-Bit Analog Change Event without Time	1	06,07,08	129, 130	17, 28
32	2	16-Bit Analog Change Event without Time	1	06,07,08	129, 130	17, 28

OBJECT			REQUEST (Library will parse)		RESPONSE (Library will respond with)	
Object	**Variation**	**Description**	**Application Layer Function Codes (Decimal)**	**Qualifier Codes (hex)**	**Application Layer Function Codes (Decimal)**	**Qualifier Codes (hex)**
32	3	32-Bit Analog Change Event with Time	1	06,07,08	129, 130	17, 28
32	4	16-Bit Analog Change Event with Time	1	06,07,08	129, 130	17, 28
32	5	Single-precision float –point Analog Change Event without Time	1	06,07,08	129, 130	17, 28
32	7	Single-precision float –point Analog Change Event with Time	1	06,07,08	129, 130	17, 28
33	0	Frozen Analog Event - All Variations	1	06,07,08	129	17, 28
33	1	32-Bit Frozen Analog Event without Time	1	06,07,08	129, 130	17, 28
33	2	16-Bit Frozen Analog Event without Time	1	06,07,08	129, 130	17, 28
33	3	32-Bit Frozen Analog Event with Time	1	06,07,08	129, 130	17, 28
33	4	16-Bit Frozen Analog Event with Time	1	06,07,08	129, 130	17, 28
33	5	Single-precision float –point Frozen Analog Change Event without Time	1	06,07,08	129, 130	17, 28
33	7	Single-precision float –point Frozen Analog Change Event with Time	1	06,07,08	129, 130	17, 28

OBJECT			REQUEST (Library will parse)		RESPONSE (Library will respond with)	
Object	**Variation**	**Description**	**Application Layer Function Codes (Decimal)**	**Qualifier Codes (hex)**	**Application Layer Function Codes (Decimal)**	**Qualifier Codes (hex)**
40	0	Analog Output Status - All Variations	1(Read), 22(Assign Class)	00, 01 (start-stop) 06 (no range, or all) 07, 08 ,(limited qty) 17, 28 (index)	129(Response)	00, 01, 17, 28
40	1	32-Bit Analog Output Status	1	00, 01 (start-stop) 06 (no range, or all) 07, 08 ,(limited qty) 17, 28 (index)	129,	00, 01, 17, 28
40	2	16-Bit Analog Output Status	1	00, 01 (start-stop) 06 (no range, or all) 07, 08 ,(limited qty) 17, 28 (index)	129,	00, 01, 17, 28
40	3	Single-precision float –point Analog Output Status	1	00, 01 (start-stop) 06 (no range, or all) 07, 08 ,(limited qty) 17, 28 (index)	129,	00, 01, 17, 28

OBJECT			REQUEST (Library will parse)		RESPONSE (Library will respond with)	
Object	**Variation**	**Description**	**Application Layer Function Codes (Decimal)**	**Qualifier Codes (hex)**	**Application Layer Function Codes (Decimal)**	**Qualifier Codes (hex)**
41	1	32-Bit Analog Output Block	3, 4, 5, 6	17, 28	129	echo of request
41	2	16-Bit Analog Output Block	3, 4, 5, 6	17, 28	129	echo of request
41	3	Analog Output – Single-precision float – point	3, 4, 5, 6	17, 28	129	echo of request
50	1	Time and Date	2(Write)	07 (Quantity = 1)	129	07 (quantity = 1)
60	1	Class 0 Data	1	06	129	
60	2	Class 1 Data	1	06,07,08	129	
			20 (Enable Unsolicited), 21,(Disable Unsolicited)	06		
60	3	Class 2 Data	1	06,07,08	129	
			20 (Enable Unsolicited), 21,(Disable Unsolicited)	06		
60	4	Class 3 Data	1	06,07,08	129	
			20 (Enable Unsolicited), 21,(Disable Unsolicited)	06		
80	1	Internal Indications		00 index=7		

Atop Technologies, Inc.

www.atoponline.com
www.atop.com.tw

TAIWAN HEADQUARTER:

2F, No. 146, Sec. 1, Tung-Hsing Rd,
30261 Chupei City, Hsinchu County
Taiwan, R.O.C.
Tel: +886-3-550-8137
Fax: +886-3-550-8131

ATOP CHINA BRANCH:

3F, 75th, No. 1066 Building,
Qingzhou North Road,
Shanghai, China
Tel: +86-21-64956231

ATOP INDIA OFFICE:

Abhishek Srivastava
Head of India Sales
Atop Communication Solution(P) Ltd.
No. 22, Kensington Terrace,
Kensington Rd,
Bangalore, 560008, India
Tel: +91-80-4920-6363
E-mail: Abhishek.S@atop.in

ATOP INDONESIA BRANCH:

Jopson Li
Branch Director
Wisma Lampung Jl.
No. 40, Tomang Raya
Jakarta, Barat, 11430, Indonesia
Tel: +62-857-10595775
E-mail: jopsonli@atop.com.tw

ATOP EMEA OFFICE:

Bhaskar Kailas (BK)
Vice President (Business Development)
Atop Communication Solution(P) Ltd.
No. 22, Kensington Terrace,
Kensington Rd,
Bangalore, 560008, India
Tel: +91-988-0788-559
E-mail: Bhaskar.k@atop.in

ATOP AMERICAs OFFICE:

Venke Char
Sr. Vice President & Head of Business
11811 North Tatum Blvd, Suite 3031
Phoenix, AZ 85028,
United States
Tel: +1-602-953-7669
E-mail: venke@atop.in