



TOP Server DNP 3.0 Suite

Background & Best Practices

Software Toolbox
International Corporate
Headquarters, USA

148A East Charles Street
Matthews, NC 28105 USA
www.softwaretoolbox.com

TOLL FREE: 888-665-3678
GLOBAL: 704-849-2773
FAX: 704-849-6388



Table of Contents

OVERVIEW	4
BACKGROUND	5
TECHNICAL DNP PROTOCOL INFORMATION	6
Master and Outstation Databases	6
Layering	7
Device Addressing	7
CRC Checks	8
Link Layer Confirmation	8
Transport Layer	8
Application Layer Fragments	8
Static and Event Data	9
Variations	10
Groups	11
Objects	12
Reading Data	12
Other Functions	13
Unsolicited Responses	13
Implementation Levels	13





TOP SERVER DNP APPLICATION	15
Typical DNP Polling	15
TOP Server DNP Configuration	16
Creating a Channel	16
Creating a Device	26
Device Addressing	36
CONCLUSION	38
Contact Us	38



The DNP driver, due in large part to the protocol, is a different creature than most other TOP Server drivers. DNP is a synchronous protocol in nature, and it uses both unsolicited messaging and report by exception communications to increase efficiency in bandwidth utilization. The device must be configured to send messages back to the master based on a change or at some frequency to take advantage of this ability. Device configuration and firmware are key aspects in successful communications.

When used most effectively, the DNP decouples the scan rates between the client/server and the master/slave, meaning no relationship exists between your configured client items and the items scanned by the driver. This minimizes network traffic by only including data that has changed (in a similar way to an OPC DA subscription.) Occasionally, this is not the best method to use. In these cases, the protocol supports “on-demand” polling. However, by using on-demand polling, you forfeit the advantages of exception reporting.

DNP3 was designed to optimize the transmission of data acquisition information and control commands from one computer to another. It is intended for SCADA (Supervisory Control and Data Acquisition) applications.



The DNP3 protocol, first developed by GE Harris in 1993, was a comprehensive effort to achieve open standards-based interoperability between master stations and substation computers, RTUs, and IEDs (Intelligent Electronic Devices) for the electric utility industry. The protocol is based on the work of the IEC Tech committee 57, which also resulted in the IEC 60870-5 protocol. GE Harris later turned over ownership of the DNP protocol, along with the responsibility for further development, to the DNP3 users group.

DNP is a publicly open protocol that has also become widely utilized in adjacent industries such as water/waste water, transportation, and oil and gas. These industries often have geographically dispersed field RTUs over paid transmission media such as cellular radio. The need to minimize cost on these networks made DNP an attractive option for these businesses due to its high efficiency in bandwidth usage.

A typical electric company may have a common operations center that monitors the equipment at each of its substations. In the operations center, a powerful computer stores and displays all of the incoming data. Substations have many devices that need monitoring, including circuit breakers, current sensors, voltage transducers, etc. The operations personnel often need to switch sections of the power grid in or out of service. Computers are placed in substations to collect the data for transmission to the master station in the operations center. The substation computers are also called upon to energize or de-energize the breakers and voltage regulators.



DNP3 uses the term outstation to denote remote computers that are found in the field. These may also be referred to as slaves. The term master is used for the computers in the control centers.

Outstation computers gather data for transmission to the master, such as binary input data that is used to monitor two-state devices. For example, when a circuit breaker is closed or tripped, a pipeline pressure alarm shows normal or excessive. Other data can include analog input data that conveys voltages, currents, power, reservoir water levels and temperatures, count input data that reports energy in kilowatt hours or fluid volume, or files that contain configuration data.

The master station issues control commands (for example, to close or trip a circuit breaker, start or stop a motor, or open or close a valve) or analog output Values (for example, to set a regulated pressure or a desired voltage level.)

The master and slave also communicate with each other to synchronize the time and date, send historical or logged data, send waveform data, etc.

Master and Outstation Databases

The conceptual design was to have one master computer gather information from the outstations, and then file the data from each outstation in its local database. The master uses values in its database for displaying system states, closed-loop controls, alarm notifications, billing, and more. The master keeps its database updated by polling, or sending requests to the outstation for it to return the values in the outstation database. The outstation responds to the master request by transmitting the contents of its database

The various data types in the databases are conceptually organized as arrays. An array of binary input values represents states of physical or logical Boolean devices. Values in the analog input array represent input quantities the outstation measured or computed. An array of counters represents count values (such as kilowatt hours) that are ever increasing until they reach a maximum. After reaching a maximum, the count values roll over to zero and start counting again. Control outputs are organized into an array representing physical or logical on-off, raise-lower and trip-close points. Lastly, the array of analog outputs represents physical or logical analog quantities such as those used for set points.



The elements of the arrays are labeled 0 through $N - 1$, where N is the number of blocks shown for the respective data type. In DNP3 terminology, the element numbers are called point indexes. Indexes are zero-based in DNP3, meaning the lowest element is always identified as zero.

Layering

Communication circuits between the devices are often susceptible to noise and signal distortion. DNP3 software is layered to provide reliable data transmission and establish an organized approach to the transmission of data and commands. The master and the outstation each have two software layers. The top layer is the DNP3 user layer. In the master, the DNP3 user layer interacts with the database and initiates the requests for the outstation data. In the outstation, this layer fetches the requested data from the outstation database.

The second layer is the link layer. The link layer is responsible for making the physical link reliable by providing error detection and duplicate frame detection. The link layer sends and receives packets, which are called frames in DNP3. The transmission of more than one frame is sometimes necessary to transport all of the information from one device to another. This is referred to as Application Layer Fragmenting.

Device Addressing

The destination address specifies which DNP3 device should process the data, and the source address identifies which DNP3 device sent the message. This explains the need for a DNP Master Node Address configured in the TOP Server Channel. Specifying both destination and source addresses satisfies a requirement for peer-to-peer communications, because it allows the receiver to know where to direct its responses. 65520 individual addresses are available. Every DNP3 device must have a unique address within the collection of devices sending and receiving messages to and from each other. When all the DNP Slaves are all trying to talk to the same Master Node, all of the devices need to be placed under one channel.

Three destination addresses are reserved by DNP3 to denote all-call messages, where the frame should be processed by all receiving DNP3 devices. One address is a universal address, and twelve addresses are reserved for special needs in the future.



The data payload in the link frame contains a pair of CRC octets for every 16 data octets. This provides a high degree of assurance that communication errors can be detected. The maximum number of octets in the data payload is 250, not including CRC octets. (The maximum length link layer frame is 292 octets if all the CRC and header octets are counted.)

Link Layer Confirmation

One often hears the term “link layer confirmation” when DNP3 is discussed. A feature of DNP3's link layer is the ability for the transmitter of the frame to request confirmation from the receiver that the frame arrived. Using this feature is optional, and it is often not employed because there are other methods for confirming receipt of data. This feature essentially provides an extra degree of assurance for reliable communications. If a confirmation is not received, the link layer may retry the transmission. Using link layer confirmation can be disadvantageous due to the extra time required for confirmation messages and multiple timeouts when retries are transmitted.

Transport Layer

The transport layer is responsible for breaking long application layer messages into smaller packets sized for the link layer to transmit. On the receiving end, this layer reassembles frames into longer application layer messages. In DNP3, the transport layer is incorporated into the application layer. The transport layer requires only a single octet overhead to do its job. Since the link layer can handle only 250 data octets, and one of those is used for the transport function, each link layer frame can hold up to 249 application layer octets.

Application Layer Fragments

Application layer messages are broken into fragments. The maximum fragment size is determined by the size of the receiving device's buffer. The normal range is 2048 to 4096 bytes. A message that is larger than one fragment requires multiple fragments. The application layer is responsible for fragmenting messages. It is important to note that an application layer fragment of size 2048 must be broken into 9 frames by the transport layer, and a fragment size of 4096 needs 17 frames. It has been shown that communication is sometimes more successful for systems operating in high noise environments if the fragment size is significantly reduced. This is a slave configuration setting that cannot be controlled in the TOP Server.



The TOP Server DNP driver will often queue multiple commands within a typical DNP timeout period. The DNP stack must dispose of these commands in the order they are received. Outstanding commands for still-responsive slave devices can be blocked until the command queue empties. This means that requesting too much data at too fast of a rate can be painful. However, due to the report by exception and unsolicited messaging features, it is very efficient in bandwidth usage.

To reap the benefits of the DNP protocol, avoid demand polling and use reasonable integrity and event poll intervals given your transmission method and amount of data needed.

Static and Event Data

The application layer works together with the transport and link layers to enable reliable communications. It uses standard functions and data formatting to interact with the user layer.

In DNP3, the term static refers to the present value of data. Therefore, static binary input data refers to the present on/off state of a bi-state device. Static analog input data contains the value of an analog at the instant it is transmitted. DNP3 allows requests to obtain some or all of the static data in an outstation device.

The .Value at the end of a tag in TOP Server represents this static or present value information. The .Explicit at the end of a tag in TOP Server informs the TOP Server to ask for the static or present value whenever requested by the Client. This behavior is similar to an OPC DA synchronous device read request. These Explicit polls are completely separate from the Integrity and Event polls. If you need to read 10 tags from 10 devices every minute, the .Explicit feature consumes less network communications than an Integrity poll. Unlike event polls, Explicit polls provide fresh timestamps for all values.

DNP3 events are associated with significant data such as state changes, values exceeding a specific threshold, snapshots of varying data, transient data, and new information. An event occurs when a binary input changes from on to off, or when an analog value changes by more than its configured deadband limit. DNP3 provides the ability to report events with or without time stamps. Therefore, the master can generate a time sequence report if desired. The TOP Server supports these timestamps. The TOP Server Event Poll refers to a poll for any of these events from the device.

The master user layer can direct DNP3 to request events. Usually, a master is updated more quickly if it utilizes event polling most frequently from the outstation, and only occasionally performs an integrity



poll. The reason for this increase in speed is because the number of events generated between outstation interrogations is small and, therefore, less data must be returned to the master.

DNP3 goes a step further by classifying events into three classes. When DNP3 was conceived, class 1 events were considered a higher priority than class 2 events, and class 2 events were considered higher than class 3 events. While that scheme can be still be configured, some DNP3 users have developed other strategies more favorable to their operation for assigning events into classes. The user layer can request the application layer to poll for class 1, 2 or 3 events, or any combination of the three. The TOP Server supports the ability to select the class of events to poll in the Device DNP Slave Configuration setting.

There are references in DNP documentation that refer to an “Event Class 0”. This event class is synonymous to an Integrity poll. Other master solutions exist that specify Event Class Poll intervals for Classes 0, 1, 2 and 3. In these cases, you will need to specify the desired Event Class 0 poll interval in the Integrity Poll interval field in TOP Server.

Variations

DNP3 has the ability to represent data in different formats. An examination of analog data formats is helpful in understanding the flexibility of DNP3. Static, present Value, analog data can be represented by the following variation numbers:

1.	32-bit integer value with flag
2.	16-bit integer value with flag
3.	32-bit integer value
4.	16-bit integer value
5.	32-bit floating point value with flag
6.	64-bit floating point value with flag

The flag is a single octet containing bit fields that indicate if the source is online, the data source restarted, communications were lost with a downstream source, or the data is forced and the Value is over range.



Not all DNP3 devices can transmit or interpret all six variations, but they must be able to transmit the simplest variations so that any receiver can interpret the contents.

Event analog data can be represented by the following variations:

1.	32-bit integer value with flag
2.	16-bit integer value with flag
3.	32-bit integer value with flag and event time
4.	16-bit integer value with flag and event time
5.	32-bit floating point value with flag
6.	64-bit floating point value with flag
7.	32-bit floating point value with flag and event time
8.	64-bit floating point value with flag and event time

The flag has the same bit fields as for the static variations.

Groups

As you can see by looking at the above variations, analog events 1 and 2 cannot be differentiated from variation 1 and 2 static analog values. DNP3 solves this predicament by assigning group numbers. Static analog values are assigned as group 30, and event analog values are assigned as group 32. Static analog values (group 30) can be formatted in one of 6 variations, and event analog values (group 32) can be formatted in one of 8 variations.

When a DNP3 outstation transmits a message containing response data, the message identifies the group number and variation of every value within the message. Group and variation numbers are also assigned for counters, binary inputs, controls, and analog outputs. In fact, all valid data types and formats in DNP3 are identified by group and variation numbers. Defining the allowable groups and variations helps DNP3



ensure interoperability between devices. DNP3 basic documentation contains a library of valid groups and their variations.

Objects

When data from an index is transmitted across the wire, the sender must properly encode the information to enable a receiving device to parse and interpret this data. The bits and bytes for each index in the message are called an object. This means that objects in the message are the encoded representation of the data from a point, or other structure, and the object format depends upon which group and variation number are chosen.

An Object is the fully encoded message based on the combination of the group number, variation requested, index starting point in the array, and .sub attribute. Object has become synonymous with Group in some DNP device and software documentation, and the TOP Server uses Object number for Group number.

DNP3 also uses a construct known as Request Qualifier Codes for certain Objects. There is a list of Request Qualifier Codes per Object type in the driver help file. Specifically, we know DNP Object 41 supports two different qualifiers codes—Code 17 or Code 28. Code 17 is used by TOP Server for requesting indexes 0-255, and Code 28 is used by TOP Server for all indexes larger than 255, as per the DNP3 Protocol Specification. Code 28 can technically be used for indexes smaller than 255, but this is more inefficient because the packet size for such a request would be larger than if Code 17 were used.

Reading Data

The master user layer formulates its request for data from the outstation by telling the application layer what function to perform and by specifying the data types it wants from the outstation. The request can specify how many objects it wants, or it can specify a range of objects. The application layer then passes the request down through the transport layer to the link layer. In turn, the link layer sends the message to the outstation. The outstation link layer checks the frames for errors and passes them to the transport layer where the complete message is assembled in the outstation application layer. The application layer then tells its user layer which groups and variations were requested.

Responses work in a similar fashion. The outstation user layer fetches the desired data and presents it to the application layer. In turn, the application layer uses the group and variation numbers to format user



layer data into objects. Data is then passed across the communication channel to the master application layer, and the data objects are presented to the master user layer.

When not using the .Explicit sub attribute in the TOP Server, the scan rate between the client/server and DNP master/slave are completely independent. The client sends a data update request at one interval which causes the client to poll the driver data buffer at this rate. Meanwhile, the driver is performing integrity and event polls at the rate specified in the driver, and updating the buffer at this rate.

Other Functions

DNP3 software is designed to handle other functions besides reading data. The master can set the time in the outstation, transmit freeze accumulator requests, and transmit requests for control operations and setting of analog output values using select-before-operate or direct-operate sequences.

Unsolicited Responses

Unsolicited messaging is a mode of communication in which the outstation spontaneously transmits a response without having received a specific request for the data. Not all outstations have this capability. This mode is useful when the system has many outstations and the master requires notification as soon as possible after a change occurs. Rather than waiting for a master station to poll, the outstation simply transmits the change. It is possible to turn all Integrity and Event polling off in the TOP Server and only receive unsolicited messages. This is often implemented when using .Explicit sub attributes.

Before configuring a system for unsolicited messaging, a few things need to be considered. First, spontaneous transmissions should generally occur infrequently. Otherwise, excess contention can occur, and it would be better to control media access via master station polling. The second issue is that the outstation needs to know whether it can transmit data without stepping on another outstation message. DNP3 leaves specification of algorithms to the system implementer.

Implementation Levels

The DNP3 organization recognizes that supporting every feature of DNP3 is not necessary for every device. Some devices are limited in memory and speed and do not need specific features, while other devices must have more advanced features to accomplish their task. DNP3 organizes complexity into three levels. At the lowest level, level 1, only very basic functions must be provided and all others are



optional. Level 2 handles more functions, groups, and variations, and level 3 is even more sophisticated. Within each level, only certain combinations of request formats and response formats are required. This was implemented to limit software code in masters and outstations while ensuring interoperability.



There are four types of communications—three polling types and unsolicited messaging.

An integrity poll requests all data from the slave regardless of data changes and what the client is asking for. This is usually done at start up and then is performed very infrequently after startup.

Event polls request all data changes since the last event or integrity poll, regardless of what the client is polling. Make sure that the slave is configured to send timestamps so that you have the correct event time stamp in your application.

Unsolicited messages are sent by the slave on an event or data change, if it is configured to do so, without a poll request from the master. How these are handled in regards to timing, deadband, etc. are handled by the slave. Please note the importance of firewall settings. If you use unsolicited messages and your firewall does not have the proper exceptions, your unsolicited packets will be rejected.

Demand polling allows the driver to work in a traditional master/slave polling method. This will retrieve data for points defined for demand polling whether the data has changed or not, which removes the efficiencies of the DNP protocol. This is not recommended unless you absolutely must use this.

Typical DNP Polling

One common example of DNP polling is when you connect to an HMI or SCADA, causing the DNP stack to initialize the session. During initialization, an integrity poll is performed. After the integrity poll, the server only listens for unsolicited messages, never performing an event poll.

More commonly we see a mix of event polling and unsolicited messaging. After the initial integrity poll, event polls are performed at regular intervals, and integrity polls are performed at infrequent intervals. Meanwhile, the server is listening for unsolicited messages (if the slave is configured for such.)

Sometimes devices have data points that can only be read explicitly from the device. This requires demand polling. Demand polling may also be used for regulatory reporting requirements that require a point(s) to be reported at specific intervals, regardless of value change.

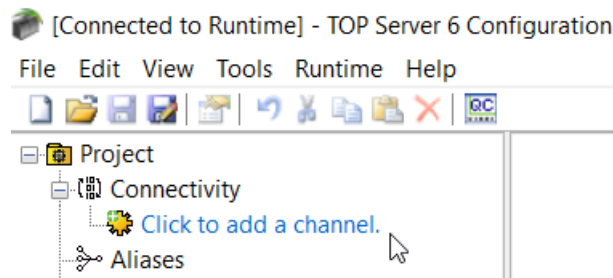


Please see the section below for step by step instructions on the configuration of a DNP channel and device in the TOP Server.

Creating a Channel

The first step in connecting to your device is creating a channel in the TOP Server. Ethernet settings or Ethernet encapsulation are done at the channel level (instead of the device level as with most drivers.)

When you open a blank configuration, you will see in the left-hand pane of the interface “Click to add a channel”. This launches the channel configuration wizard.



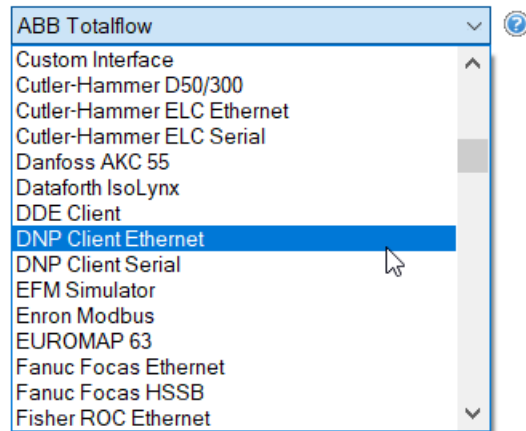
The first thing we must do is select the driver. There are two Drivers, DNP Master Serial and DNP Master Ethernet. We will start with an Ethernet setup. Click next to give the channel a name.





← Add Channel Wizard

Select the type of channel to be created:

A list box showing various channel types. The list includes: ABB Totalflow, Custom Interface, Cutler-Hammer D50/300, Cutler-Hammer ELC Ethernet, Cutler-Hammer ELC Serial, Danfoss AKC 55, Dataforth IsoLynx, DDE Client, DNP Client Ethernet (highlighted), DNP Client Serial, EFM Simulator, Enron Modbus, EUROMAP 63, Fanuc Focas Ethernet, Fanuc Focas HSSB, and Fisher ROC Ethernet. A mouse cursor is pointing at "DNP Client Ethernet".

Next

Cancel

This can be any name that is useful or meaningful to the project. In this case, we will use "DNP_Ethernet."







Add Channel Wizard

Specify the identity of this object.

Name:

 A text input field containing the text "DNP_Ethernet". To the right of the input field is a small circular icon with a question mark, typically used for help.




 Add Channel Wizard

Limit data transmissions to one channel at a time by assigning this channel to a virtual network.

Virtual Network:


None



Specify the number of transactions to perform when a channel is given permission to communicate.

Transactions per Cycle:

1





Next

Cancel

The next channel parameters are for network interface and write optimizations. In systems with more than one network card or IP address, you can pick which card or IP address to bind to using the dropdown. If you have a single NIC or IP address, default is acceptable. Write optimization settings are covered in the help file, but it is usually best to leave these at the default.







 Add Channel Wizard

Specify the name of a network adapter to bind or allow the OS to select the default.

Network Adapter:



Next

Cancel



✕


←

Add Channel Wizard

Choose how to send invalid floating-point numbers to the client.

Floating-Point Values:

Unmodified ▼



Next

Cancel



X

← Add Channel Wizard

Choose how write data is passed to the underlying communications driver when more than one write exists in the write queue.

Optimization Method:

Write Only Latest Value for All Tags ?

Specify the ratio of write operations to read operations, based on one read per configurable number of writes.

Duty Cycle:

10 ?

Next Cancel

Pictured below are the Ethernet settings on the channel level. The master node must also be configured at the slave and must match the setting here (this is the Node used to represent the TOP Server Channel). The IP Address, slave port and connection type are also configurable at the slave and must match the master settings. This is due to the point-to-point nature of the protocol. If using UDP then you must also configure the UDP Listener port to match at both the master and slave.






Add Channel Wizard

Specify the protocol that should be used for communicating with the DNP outstation.

Protocol:



Specify the port used to receive UDP traffic. Configuring this will bind the DNP client to the specified port.

Source Port:



Specify the destination IP address or hostname.

Destination Host:



Specify the destination port.

Destination Port:



Next

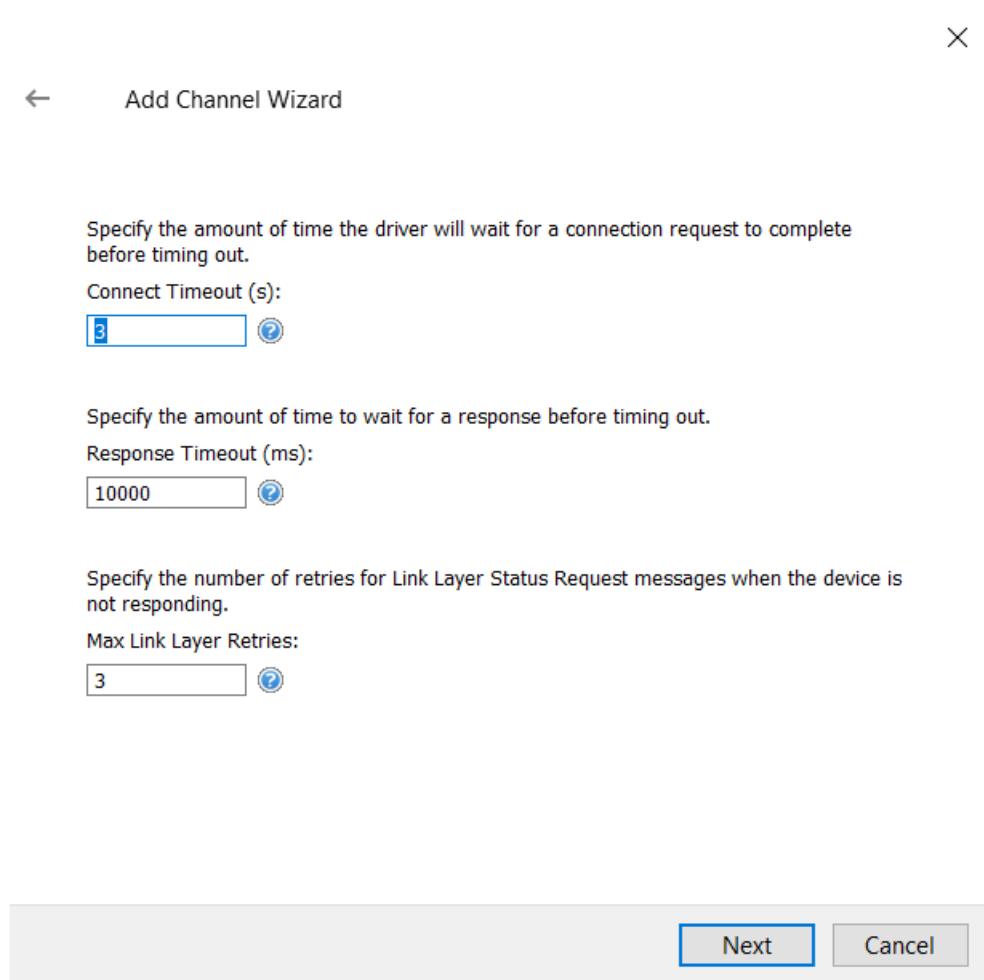
Cancel

The Channel defines your computer as a DNP Master device in a DNP Master/Slave conversation. Many DNP slaves require that you configure in the slave the address of the Master device. You may not be used to doing this in typical PLC communications, but it is very important for DNP. If your DNP Master ID set at the channel level does not match what is configured in your slave device, your unsolicited messages from the slave will not arrive, and you may experience no successful communication at all.

Also, with the DNP drivers, each device must go under a unique channel. (This is already a recommended performance optimization to prevent an offline device from slowing down other device communication.) It is a requirement to use one device per channel for Ethernet and Ethernet Encapsulated DNP devices. Pure serial DNP connections do not have this restriction, but this can be a problem when all the DNP Slaves are configured for the same Master Node number, because then each Channel must have a unique Node number.



There are two timeout settings at the channel level, Connect Timeout and Response Timeout. These are common to both the serial and Ethernet drivers. In DNP, there are several timeout settings that we do not see in any other driver.

A screenshot of the "Add Channel Wizard" dialog box. It has a title bar with a close button (X) and a back arrow. The main area contains three sections: 1. "Specify the amount of time the driver will wait for a connection request to complete before timing out." with a label "Connect Timeout (s):" and a text box containing "3". 2. "Specify the amount of time to wait for a response before timing out." with a label "Response Timeout (ms):" and a text box containing "10000". 3. "Specify the number of retries for Link Layer Status Request messages when the device is not responding." with a label "Max Link Layer Retries:" and a text box containing "3". Each text box has a help icon (question mark in a circle) to its right. At the bottom right, there are "Next" and "Cancel" buttons.

The Response Timeout is simply the time we wait for a response once the message has been sent.

The Connect Timeout is used with TCP only, and is irrelevant with UDP. This is the time we wait to establish the TCP Connection. The Request Timeout, which is set at the device level, is the timeout period that includes both time on the wire and time in the queue. This should be greater than the Response Timeout.



The general rule is that the Request Timeout should be greater than or equal to the number of devices in the channel plus 1, multiplied by the Response Timeout.

$$\text{Request Timeout} \geq (\text{Number of devices in channel} + 1) * (\text{Response Timeout})$$

For example, if we take 1 channel with a single device and use the default Response Timeout of 10 seconds, then $(1 \text{ device} + 1) * 10$ makes a Request Timeout of at least 20 seconds.

The Channel Wizard will conclude with a summary of the settings.



Add Channel Wizard

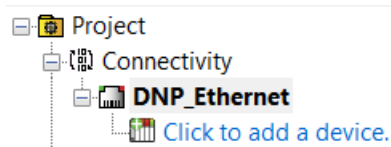
Identification	
Name	DNP_Ethernet
Description	
Driver	DNP Client Ethernet
Diagnostics	
Diagnostics Capture	Disable
Tag Counts	
Static Tags	0
Ethernet Settings	
Network Adapter	Default
Write Optimizations	
Optimization Method	Write Only Latest Value for All Tags
Duty Cycle	10
Non-Normalized Float Handling	
Floating-Point Values	Unmodified
Channel-Level Settings	
Virtual Network	None

Finish

Cancel

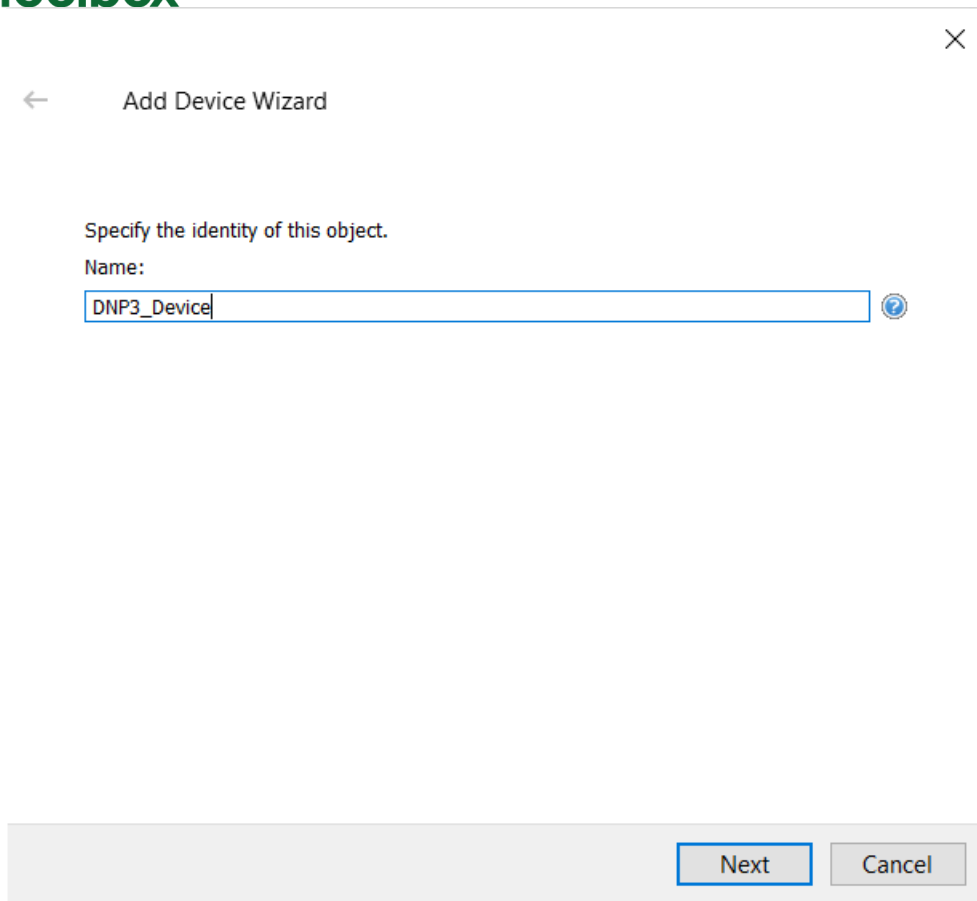
Creating a Device

Now you see your channel you have just configured, and underneath you see "Click to add a device." Click to start the device configuration wizard.



Just as with the channel configuration, the first thing you will do is provide a meaningful name to the device.



A screenshot of the "Add Device Wizard" dialog box. It has a title bar with a close button (X). Below the title bar is a back arrow and the text "Add Device Wizard". The main area contains the instruction "Specify the identity of this object." followed by a "Name:" label and a text input field containing "DNP3_Device". A help icon (question mark in a circle) is to the right of the input field. At the bottom right are "Next" and "Cancel" buttons.

← Add Device Wizard

Specify the identity of this object.

Name:

DNP3_Device

Next Cancel

You will soon reach a Communications Dialog as shown below.

The Communication Settings section is used to specify the DNP master and slave's 16 bit addresses, the Request Timeout we mentioned earlier, and the keep-alive interval.

The Max Timeouts specifies the maximum number of successive timeouts that can occur with the same request before the device is considered to be in error. A timeout occurs when the entire request and response do not complete within the Device Request Timeout, or when the request successfully transmits but the response is not received within the Channel Response Timeout.

Due to incremented sequence numbers, the regenerated request is not identical to the original request. Requests to and responses from other devices on the same channel may occur between retries. The valid range is 1 to 10 timeouts.



The Keep-Alive Interval specifies when to transmit a keep-alive status request to the slave. The valid range is 0 to 86400 seconds. The default setting is 0 seconds (which indicates that a keep-alive status request message will not be sent.)

Each DNP point has .timestamp subtype that stores the timestamp of an associated event. You can choose to display this in UTC notation or in your local time format.

In the DNP protocol, the slave may send a synchronization request to the master to synchronize its clock. There are two options—LAN and Serial. Only use LAN for a hardwired Ethernet connection. To get the true time for serial, radio, dial-up, or even wireless Ethernet, use Serial. When you use serial, the master will make a delay measurement, meaning it calculates lag time and sets the time in object 50, variation 1, to the lag corrected time.

Now we'll look at how to setup the polling at the device level and finish the timeout settings that we started earlier with the Response Timeout when we configured the channel.



First, you will configure the event polling. DNP has different event classes that you may use to classify points. You can configure each event class poll time separately. You may set each to a maximum of 24 hours (in seconds), or minimum of 0 (to turn off event polls.)


×

← Add Device Wizard

Specify the amount of time between each poll for event data changes.

Class 1 Poll Interval:


5



Specify the polling interval resolution.

Class 1 Poll Interval Resolution:


Seconds



Specify the amount of time between each poll for event data changes.


Class 2 Poll Interval:

5



Specify the polling interval resolution.

Class 2 Poll Interval Resolution:



Next

Cancel

Next, you will configure your integrity poll time. You can set this to a maximum of 30 days. Setting this to 0 turns off integrity polling, but we do not recommend this. We recommend this to be no more than once per hour. However, for most applications, very infrequent times of 8 to 24 hours or more is sufficient.



X

← Add Device Wizard

Specify the polling interval at which a complete data retrieval is requested from the DNP server device. To turn off integrity polling enter a value of zero (0).

Integrity Poll Interval (s):

 ?

Specify if integrity polls should be performed whenever the DNP client restarts.

Issue Integrity Poll On Restart:

 ?

Specify if integrity polls should be performed whenever the DNP server comes online.

Issue Integrity Poll On DNP Server Online:

 ?

Specify if integrity polls should be performed whenever the DNP server indicates it has an event buffer overflow.

^

v

The DNP protocol allows the slave to update the master with data changes without a poll from the master. This is fully controlled by the slave and most slaves allow you to configure delays or item deadbands for unsolicited messages. The deadband settings can prevent you from receiving every data change, which can be good or bad based on data reporting needs and bandwidth limitations.

The Unsolicited Messaging dialog is used to specify whether the DNP slave will send class 1, 2, and 3 unsolicited data updates. These parameters specify whether unsolicited messaging will be allowed. Options include “Automatic”, “Enable”, and “Disable”. “Automatic” takes no action and is at the slave's discretion. “Enable” permits the reporting of data updates for the selected classes. “Disable” turns off unsolicited messaging.

When “Disable unsolicited messaging during start up” is checked, unsolicited messaging will be prevented during start up. This option is only available when one or more classes have “Enable” selected, and no class has been set to “Automatic.” This setting applies to all Event classes.




×

← Add Device Wizard


Specify if unsolicited messaging is allowed for Class 1 data.

Unsolicited Mode Class 1:

Automatic ▼ 


Specify if unsolicited messaging is allowed for Class 2 data.

Unsolicited Mode Class 2:

Automatic ▼ 


Specify if unsolicited messaging is allowed for Class 3 data.

Unsolicited Mode Class 3:

Automatic ▼ 

Specify whether unsolicited messaging should be allowed during startup.

Use Unsolicited Messaging During Startup:

☐ 

Next

Cancel

So, what happens when communications fail between the master and slave? Most of the time you would lose that data however, DNP slaves can store event data for each point and then send the data when communications are re-established. Your slave determines if it can buffer, how much data it can buffer, and all other buffering settings. Configuration of these slave hardware settings is outside of the scope of TOP Server support.

Once we have good communications what happens? A properly configured DNP slave that fully supports the DNP protocol specification will send the events with timestamps to the master in the sequence with oldest timestamps sent first.

The TOP Server can then buffer these events up to the configurable maximum per tag and play these out to the client application at a user configurable playback rate.



Please do note that we replay these events in the order we receive them from the slave, so if the slave provides the events in the correct timestamp order, the events will be sent to the client, HMI or SCADA, in the same order.

The buffering feature only applies to the Objects listed that support it in the help file.

×

← Add Device Wizard

Specify if event reports received from the remote DNP device can be buffered and played back for OPC client collection.

Event Buffer:

Disable

?

Specify the maximum number of events to be collected per point.

Max Events Per Point:

100

?

Specify the rate at which event reports are played back.

Playback Rate (msec):

2000

?

Next

Cancel

Using event buffering does come at some opportunity cost. When enabled, it will introduce latency into the tags of the affected objects. After the initial group of events is sent, new updates from the slave will only be released to the client at the playback rate. Therefore, it is wise to use this feature only when capturing all events is more important than fast delivery of the events to the client.

There are three settings related to event buffering. First, you must enable the feature using the checkbox. Then you will configure the maximum events per tag to be buffered. The range available is 1 to 1000. Finally, you will configure the rate at which events are played out to the client. It is important to note that



if you want to insure retrieval of all events, the client scan rate must be at least twice as fast as the playback rate configured here.


Next you will be given the opportunity to import tags.

✕

← Add Device Wizard


Controls whether or not this device will automatically generate new tags whenever certain properties are changed.

On Property Change:

Yes 


Select the automatic tag generation action to be taken on device startup.

On Device Startup:

Do Not Generate on Startup 


Indicate the preferred method of avoiding creation of duplicate tags.

On Duplicate Tag:

Delete on Create 


Indicate a tag group name for new generated tags. If empty, generated tags are added at the device level.

Parent Group:



Instruct the server to automatically create sub groups for automatically generated tags.

Allow Automatically Generated Subgroups:

Enable 

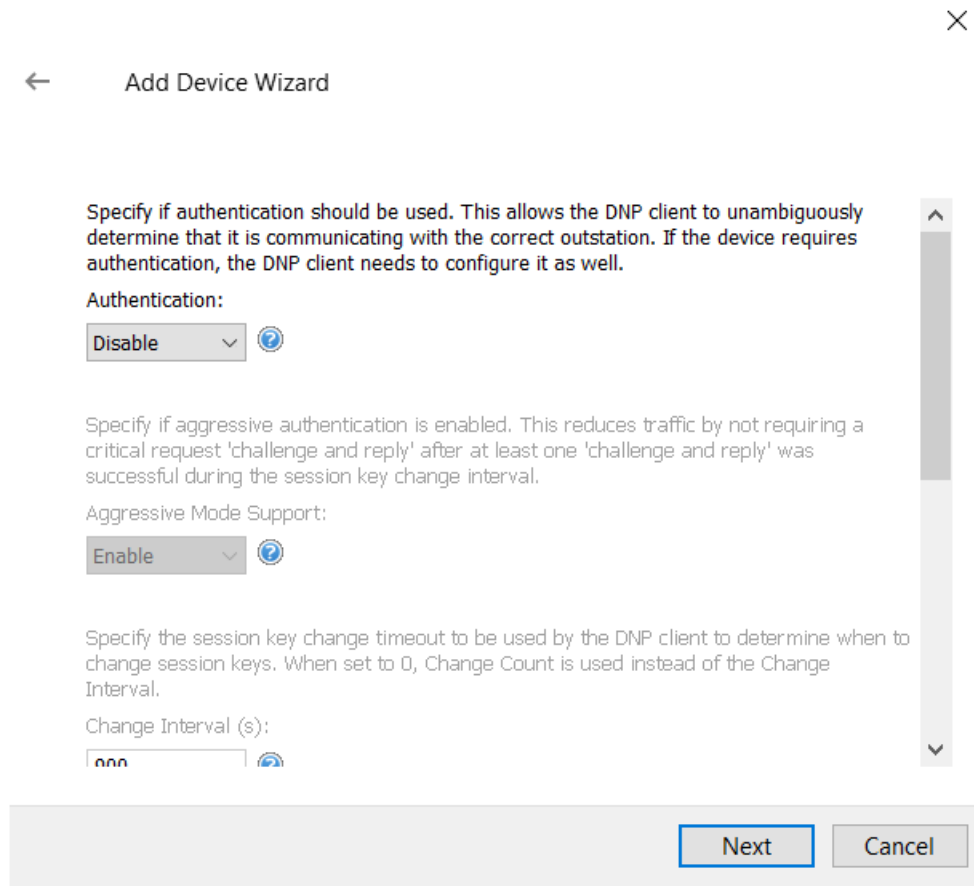
Next

Cancel

The driver will create all tag groups through communication with the device after it has been determined that tags are available in the target device. For accurate tag import, the communication settings must be correct.



Next, you will come to an Authentication dialog. When checked, this parameter enables Authentication. If the device requires Authentication, this must also be configured in TOP Server.



← Add Device Wizard

Specify if authentication should be used. This allows the DNP client to unambiguously determine that it is communicating with the correct outstation. If the device requires authentication, the DNP client needs to configure it as well.

Authentication:

Disable

Specify if aggressive authentication is enabled. This reduces traffic by not requiring a critical request 'challenge and reply' after at least one 'challenge and reply' was successful during the session key change interval.

Aggressive Mode Support:

Enable

Specify the session key change timeout to be used by the DNP client to determine when to change session keys. When set to 0, Change Count is used instead of the Change Interval.

Change Interval (s):

000

Next Cancel

A tag import will be performed when this property changes. This ensures that the Authentication Object internal Statistics Tags will be automatically generated when Authentication is enabled. These tags are pre-defined, and may be imported without communication with the device. When Authentication is disabled, a tag import will be performed in order to remove the Authentication Object internal Statistics Tags.

Aggressive Mode Support reduces traffic by not requiring a critical request "challenge and reply" after at least one "challenge and reply" was successful during the session key change interval. The Session Key Change Interval specifies the session key change timeout that will be used by the master to determine when to change session keys. When a value of 0 is entered, Session Key Change Count will be used

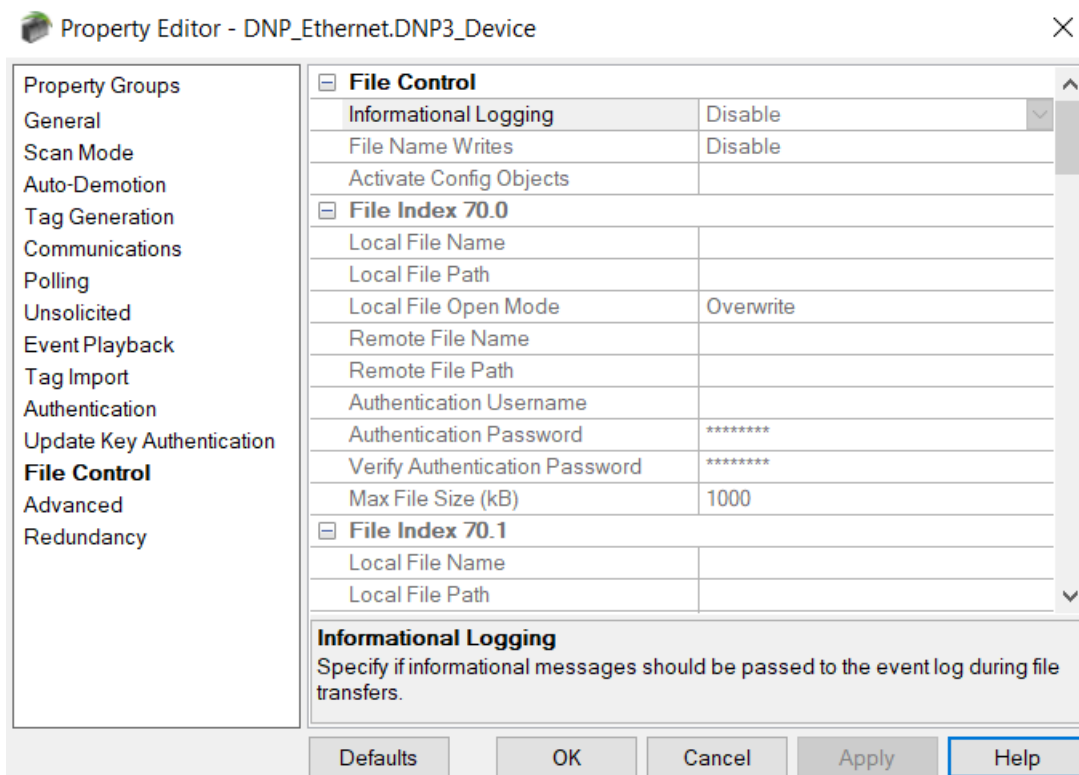


instead. The valid range is 0 to 7200 seconds. The Session Key Change Count specifies the number of transmitted authentication messages at which the master will change session keys. The messages may have been transmitted in either direction. The valid range is 0 to 65535.

The Reply Timeout specifies how long the device will wait for an Authentication reply. The valid range is 0 to 300000 milliseconds. The Max Error Count specifies the number of error messages that will be sent before error message transmission is disabled. It is also used to limit the number of Authentication attempts when there is no reply from the slave. With proper timeout settings, the maximum number of authentication retries per response timeout will be Max Error Count + 2. The valid range is 0 to 10.

The DNP Master Ethernet Driver will automatically match the HMAC algorithm as configured in the slave.

The File Control feature set is intended to be used as a mechanism for transferring log and configuration files between DNP masters and slaves. The DNP Master Ethernet Driver supports the transfer of files to and from a DNP slave. In the File Control tab of Device Properties, users can change a setting by clicking in the second column of the parameter. In most cases, this will invoke a drop-down menu that displays the available options. For the path properties, this will invoke a browse button instead.




There are a number of advanced DNP settings to configure next. More information regarding these settings can be found in the help file.

Device Addressing

When setting up DNP tags, there are a few important things to be aware of. It is essential to have the DNP slave profile document for your device. This will let you know what the DNP addresses are used for, what event class they belong to, and how to access them. To use this document, you also need to understand the basic naming syntax for DNP standard items.

DNP items are addressed with a four part address:

object.variation.index.subtype or sub attribute.

The object is the data object group to which the item belongs. Variation defines the default data type for the item. Index is the data object within the group. For example, the 5th point in the group would have an index of 4, because indexes and variations start with 0. The sub attribute or subtype is the attribute you wish to read for the specific index or point. Most often, you will use .Value to read the index Value, but, depending on the object and variation, there are a number of attributes available such as .Timestamp, .Explicit and .Restart.

Alternative to .Value, you would use the .Explicit sub attribute to cause the tag to become demand polled, which as we discussed earlier will cause the tag to work like tags in traditional polling PLC drivers.

For example, if we wish to address the value of the 52 analog input, we would use 30.0.51.Value. 30 defines the Analog Input object, 0 defines the default variation or data type, 51 is the index of the 52nd analog input (remembering that indexing starts at 0), and Value reports the Value of the input.

Likewise, if we wish to know the timestamp of the event reported on the 3rd binary input we would define the address 1.0.2.Timestamp where 1 defines the binary input object, 0 defines the default variation or data type (in this case, date), 3 indicates the fourth binary input (once again, with indexing starting at 0), and Timestamp tells us to report the date/time of the event Value currently reported by the .Value sub attribute.

There are some other subtleties to DNP Addressing we need to understand. In a proper DNP implementation, the number of tags in the client or server will have no effect on what points are scanned



or updated from the DNP slave. The driver abstracts many DNP protocol and slave profile details on its own. The important part is to focus on finding the addressing lists in your slave profile. This process could take some time. If you understand DNP addressing and your DNP slave device vendor has provided a clear and concise document, this process will not be too difficult.

If you do not see an object number from your slave profile list in the addresses supported by the server, this does not mean it is not supported. Some objects are “reflected” or addressed under the hood using a base object. There is great detail in the help file about these objects. For example, if you wish to read from object 32, you would address that by using the correct variation of object 30 to cause object 32 to be used.

It is also important to note that in some devices, you will need to write the Latch On and Latch Off commands to different bits. If you see that writing Latch On commands are successful, but Latch Off commands aren't, this is likely to case.



The TOP Server DNP Suite can connect your HMI/SCADA/MES system to your DNP 3 Slave Devices via serial, Ethernet encapsulated serial, or true TCP/IP Ethernet connections. The driver acts as a DNP Master and supports unsolicited messaging with DNP devices, user configurable polling rates, timeouts, and more.

Contact Us

If you have any questions or seek further information and help:

Online Support: <http://support.softwaretoolbox.com>

Email Support: support@softwaretoolbox.com

Phone Support: +1 (704) 849-2773

Fax: +1 (704) 849-6388

