DNP 3.0 and PowerLogic® ION Technology

The Distributed Network Protocol Version 3.0 (DNP 3.0) is an open SCADA (system control and data acquisition) protocol used for communications and inter operability among substation computers, RTUs, IEDs (Intelligent Electronic Devices), and Master Stations. DNP is used for substation automation such as reclosing schemes automation, adaptive relaying, capacitor bank control, auto load transfer and bus tie control.

DNP 3.0 is managed by the DNP User's Group whose members represent utilities and equipment vendors.

In this document

Introduction 2
DNP Architecture Overview 2
DNP Protocol and ION Meters 6
Port Configuration 6
DNP Slave Options Module 7
DNP Slave Export Module 8
DNP Slave Import Module 10

The overview of DNP architecture is derived from *DNP3 Overview* found at www.TriangleMicroWorks.com.

Consult www.powerlogic.com for the following references:

- The online *ION Reference* for a detailed description of the DNP Slave Import, Export and DNP Options modules.
- The *Device Profile* for your meter.

Consult www.dnp.org for general DNP user's group information.

www.powerlogic.com

Schneider Electric

Introduction

DNP is a data protocol for communications used to monitor and control equipment, for example, monitoring current and voltage transducers, to close circuit breakers, open or close valves or regulate pressure.

You can use ION Enterprise software to configure ION meters to communicate over a DNP network. You should have a good understanding of DNP 3.0 to interpret the settings in the DNP Options module and the DNP Slave Export modules.

Consult the DNP User's Group (www.dnp.org), the online *ION Reference* and your meter’s *Device Profile* document. The device profile document is designed to address application and data link layer issues, identify deviations from subsets and provide an implementation table that lists the object variations, function codes and qualifier, that the device supports.

DNP Architecture Overview

DNP 3.0 is a layered protocol that adheres to a simplified three layer standard proposed by the IEC (International Electrotechnical Commission). The OSI (Open System Interconnection) is the more common seven-layer protocol standard. IEC calls this three layer standard, the Enhanced Performance Architecture, or EPA. DNP 3.0 adds a fourth layer to the EPA called a pseudo-transport layer that allows for message segmentation.

Physical Layer

The physical layer is the material media over which the protocol is communicated. Most commonly, DNP is specified over a simple serial physical layer such as RS-232 or RS-485. The media of the layer can be fiber, radio or satellite; support on Ethernet is also defined in the standard. The physical layer of DNP handles the state of the media (clear or busy), and synchronization across the media (starting and stopping). The maximum size of a physical layer is 292 bytes.

Data Link Layer

The data link layer manages the logical link between sender and receiver of information and it improves error characteristics. This is accomplished by starting each data link frame with a header and inserting a 16-bit CRC every 16 bytes of the frame (a frame is a portion of a complete message communicated over the physical layer). The maximum size of a data link frame is 256 bytes. Each frame has a 16-bit source address and a 16-bit destination address, which may be a broadcast address (0xffff). The address information, along with a 16-bit start code, the frame length, and a data link control byte, and two CRC bytes is contained in the 10-byte data link header.

Data Link Header:

Start Octets		Length	Data Link Control	Destination		Source		CRC		
05	64			LSB	MSB	LSB	MSB			10 bytes total

Data Block:

Transport Header	Application Header	Object Header #1	Data #1	Object header #2	Data #2	CRC used every 16 bytes		Data #n	CRC	
I BYTE	2 OR 4 BYTES	VARIABLE SIZE	VARIABLE SIZE	VARIABLE SIZE	VARIABLE SIZE			...		

The data link control byte indicates the purpose of the data link frame, and status of the logical link. Possible data link control byte values include: ACK, request data link confirm (ACK) of frame, NACK, request link status, link needs reset, link status reply, link is reset.

When a data link confirmation is requested, the receiver must respond with an ACK data link frame if the frame is received and passes CRC checks. If a data link confirmation is not requested, no data link response is required.

Pseudo-Transport Layer

The pseudo-transport layer segments application layer messages into multiple data link frames. For each frame, it inserts a single byte function code that indicates if the data link frame is the first frame of the message, the last frame of a message, or both (for single frame messages). The function code also includes a rolling frame sequence number which increments with each frame and allows the receiving transport layer to detect dropped frames.

Application Layer

The application layer responds to complete messages received and builds messages based on the need for, or availability of user data. Once messages are built, they are passed down to the pseudo-transport layer where they are segmented and passed to the data link layer and communicated over the physical layer. The total length of received messages is indicated by pseudo-transport layer as it appends data link layer frames, each with their own indicated length.

When the data is too large for a single application layer message, multiple messages may be built and transmitted sequentially. Each message, except the last, indicates that more messages follow. Because the application data can be fragmented, each message is referred to as a fragment that may either be single-fragment or a multi-fragment.

Application layer fragments from master DNP stations are typically requests for operations on data objects. Application layer fragments from Slave DNP stations are typically responses to those requests.

As in the data link layer, application layer fragments may be sent with a request for confirmation. An application layer confirmation indicates that a message is received and parsed without error. (A data link layer confirmation (ACK) indicates that the data link frame is received and it passed CRC error checks.)

Each application layer fragment begins with a header followed by one or more object header/object data combinations. The header contains an application control code and an application function code. The application control code contains the following information:

- identifies if the message is multi- or single fragment
- specifies if an application layer confirmation is requested for the fragment
- identifies if the fragment was unsolicited
- provides a rolling application layer sequence number

The application layer sequence number allows the receiving application layer to detect fragments that are out of sequence or dropped. The application layer header function code indicates the purpose, or requested operation of the message. DNP allows multiple data types in a single message but it only allows a single requested operation on the data types within the message. The header function code applies to all object headers, and therefore all data within the message fragment.

Point Map Organization

In DNP data is organized into data types. Each data type is an object group, including: binary inputs (single-bit read-only values), binary outputs (single-bit values whose status can be read, pulsed or latched directly, or through SBO type operations), analog inputs (multiple-bit read-only values), analog outputs (multiple-bit values whose status may be read or controlled directly, or through SBO type operations), binary counters (multiple-bit values used to store increasing parameters such as energy counts), time and date, file transfer objects.

One or more data points, or data types, exist for each object group. A data point is a single data value that is specified by its object group. Variations exist within each object group. An object group variation is typically used to indicate a different method of specifying data within the object group. For example, variations of analog inputs allow for transfer of the data as 16-bit signed integer values, or 32-bit signed integer values.

Since an application layer message may contain multiple object headers, object header that specifies an object group, a variation of the object group, and a range of points within that object group variation. Some application layer header function codes indicate that object data follows each object header. Other function codes indicate that there is no object data in the message - instead multiple object headers, if present, follow each other contiguously. For example, a read request message fragment only contains object headers that describe the object groups, variations, and point ranges that are requested to be read and responded; a read response message fragment contains object headers and the request object data.

DNP allows object point ranges to be specified in a variety of ways. For request messages, object point ranges may consist of a request for: all points of the specified object group, a contiguous range of points beginning with a specified starting point and ending with a specified stopping point, a maximum quantity of points, a list of requested points.

For response messages, object point ranges typically consist of either a contiguous range of points, with a specified starting point and stopping point, or with a list of points. For response object point ranges that consist of a list of points, a point number precedes each data object. The number of points in the list is specified as part of the object point range.

Reporting Model

Many of the object groups have corresponding object groups that contain change data. Change data represents points that have changed for a specifically corresponding object group. For example, object group 1 represents binary inputs (considered static data), and object group 2 represents binary input change data. When a point in object group 1 is detected to have changed, a change event is created in object group 2 for the same point number (if binary input change events are enabled). Including only points that have changed in response messages allows smaller, efficient messages. Such reporting schemes are called report-by-exception, or RBE.

For each change-data point, a time can be associated with the change. For binary input, each detection of a data value that changes is considered a change event. For analog input and binary counter, an event is created when the change is by a certain amount (deadband). At any given time, it is possible to have multiple change events for some points and no change events for other points.

Object groups and the data points within them, are further organized into classes. This provides an efficient method of requesting data; a simple (and small) message is sent to request all data in a specific class. This is referred to as scanning for class data. There are four classes defined in DNP 3.0. Class 0 represents all static (not event data). Classes 1, 2, and 3, represent different priorities of change event data. By associating different event data with different classes, the classes are requested with varying periodic rates.

NOTE

Frozen statics and frozen events are specific to analog inputs and binary counters.

If class 1 contains the highest priority change-event data and class 3 contains the lowest priority change-event data, then a class 1 poll should be performed as often as possible, a class 2 poll performed less often, and a class 3 poll performed even less often. For each class data response, only the class data that has changed is returned. Finally, to acquire data not associated with either class 1, 2, or 3, an integrity poll, consisting of a class 0 scan, is performed. Since a large amount of data may be returned in a class 0 scan, it should not be performed often.

DNP Protocol and ION Meters

You integrate your ION meter into a DNP network using ION Enterprise software. In Designer, use the DNP Slave Import module to take the value of a DNP object (written by a DNP master device) and import it into an ION register. ION Import modules are then linked to other ION modules (such as a digital output module). The Export module takes the value from the ION register and creates a DNP object that can be read by a DNP master device.

The DNP Options module lets you specify global options for all DNP Slave Import and Export modules. You can define the variants that are returned for various classes of polling, the maximum number of events stored on the device and communications options such as time synchronization periods and data link layer settings. The output registers on the DNP Options module indicate how much event buffer space is available.

NOTE

Complete DNP documentation is available at www.dnp.org. DNP 3.0 is a complex protocol please take the time to learn more or contact Technical Services for help.

Port Configuration

You can use the DNP protocol for real-time data communications on the serial, modem and infrared ports on ION meters.

These hardware ports are controlled by ION modules that exist on the meter. First, choose the hardware port through which you want to transmit data in DNP format. Set the protocol in any of these modules to DNP 3.0.

NOTE

The ION8600 Series meters support DNP on two ports at a time - all other ION meters only support DNP on one port at a time. Also, data logs and waveform logs must be communicated through the ION protocol.

DNP Slave Options Module

The DNP Options module provides global settings that affect all DNP Slave Export and DNP Slave Import modules.

Setup Register	Sample Setting*	Function - see the online *ION Reference*
BinInStatic	Single-bit Binary Input	variant for Binary Input Static objects
BinInEvents	Binary Input Change w/o time	variant for Binary Input Event objects
BinInEvDepth	100	maximum number of Binary Input Events that can be stored
BinCntStatic	16-bit Binary Counter w/o flag	variant for Binary Counter Static objects
FrzCntStatic	16-bit Frozen Counter w/o flag	variant for Frozen Counter Static objects
FrzCntEvents	16-bit Frozen Counter Event w/o time	variant for Frozen Counter Event objects
FrzCntEvDepth	100	maximum number of Frozen Counter Events that can be stored
CntChangeEvents	16-bit Counter Change Event w/o time	variant for Counter Change Event objects
CntChangeEvDepth	100	maximum number of Counter Change Events that can be stored
AIStatic	16-bit Analog Input w/o flag	variant for Analog Input Static objects
FrzAIStatic	16-bit Frozen Analog Input w/o flag	variant for Frozen Analog Input Static objects
FrzAIEvents	16-bit Frozen Analog Event w/o time	variant for Frozen Analog Input Event objects
FrzAIEvDepth	100	maximum number of Frozen Analog Input Events that can be stored
AIChangeEvents	16-bit Analog Input Change Event w/o time	variant for Analog Input Change Event objects
AIChangeEvDepth	200	maximum number of Analog Input Change Events that can be stored
AOStatic	16-bit Analog Output Status	variant for Analog Output Block objects
SelectTimeout	10	Select Before Operate timeout period (in seconds)
TimeSynchPeriod	86400	time (in seconds) between IED requests for time syncs
ALFragSize	2048	maximum application layer message size (in octets) that IED can send
DLAck	Never	when device will request data link layer acknowledgements
DLTimeout	2	how long data link layer waits (secs) for acknowledgement from Master
DLNumRetries	0	how many times a data link layer packet is re-sent after failing

*See *ION Device Templates* tables in the documentation section at www.powerlogic.com for default settings for your meter.

⚠ **CAUTION**

Do not make any changes to the DNP Options module setup register unless you understand the effects each change causes. Consult the *ION Reference* or contact Technical Services for help.

Changing the default DNP configuration:

1. Use Designer software to link the parameters that you want to access through DNP. You can add additional DNP Slave modules and link the desired ION parameters to them.
2. Right-click on the module in Designer to access the registers. If your DNP network requires data in a different format than the factory DNP configuration, you can edit the setup registers in the DNP Slave modules and the DNP Options module as necessary.

DNP Slave Export Module

The DNP Slave Export modules are pre-configured to communicate numeric or Boolean values in a specific format. If these settings do not suit your needs, you can modify the setup registers or re-link the modules to other power system data. Or, you can add additional DNP Slave Export modules and link them to ION parameters.

The DNP Slave Export module takes values from ION registers and exports them into DNP objects. The DNP Slave Export module supports four categories of DNP data objects:

- Static: real-time value of a data point (i.e. phase A voltage)
- Event: generated when a static object exceeds a deadband threshold
- Frozen Static: value represented/generated at the time the DNP master issues a "freeze" command
- Frozen Event: value represented/generated at the time the DNP master issues a "freeze" command

NOTE

The 7330/7350 meters do not support all of the DNP data objects. Consult the Device Profile document.

The possible settings for DNP Slave Export modules are outlined in the following table.

Setup Register	Settings in a default framework
BasePoint	Varies – each analog input or binary counter has a different BasePoint
StaticObj	11 modules are *Analog Input*; three are *Binary Counter*
EventObj	Disable Event Objects
Deadband	0
FrozStaObj	Disable Frozen Static Objects
FrozEvtObj	Disable Frozen Event Objects
EventClass	Class 1
Scaling	OFF (excluding *Unbalx10* and *Freqx10* which are ON)
IONZero	0
IONFull	0 (1000 for *Unbalx10* and 100 for *Freqx10*)
DNPZero	0
DNPFull	0 (10000 for *Unbalx10* and 1000 for *Freqx10*)

*See *ION Device Templates* tables in the documentation section at www.powerlogic.com for default settings for your meter.

NOTE

The supported range or options varies depending on the ION meter.

The *BasePoint* register maps the first value in a DNP Slave Export module into a DNP point number. The DNP master can then read the static, frozen, and event objects associated with that DNP point. Each subsequent value, and its related DNP point, is addressable by the appropriate offset from that BasePoint.

The *Scaling* register determines whether or not the meter's measurements are scaled for output to a DNP master. You can use this to scale up to include decimal components. *IONZero* and *IONFull* specify the range of source values while *DNPZero* and *DNPFull* specify the range of output values.

StaticObj is a Binary Input, Binary Counter or Analog input from an ION register. You can enable *EventObj,* to trigger an event when a value exceeds a specified deadband threshold. These events are returned to the master in a class 1-3 poll as specified in the *EventClass* register.

When *FrozStaObj, FroEvtObj* are enabled and a freeze occurs, the static object makes two things:

- A Frozen Static object that gets returned in a class 0 poll (a Frozen static object is overwritten each time a freeze happens)
- A Frozen Event object is returned in a class 1-3 poll (frozen events are not overwritten as more event objects are created)

EventClass defines the class of event that is created by the Export Slave module. A class 1, 2, 3 poll is then used to retrieve events as appropriate.

DNP Slave Export modules support the following DNP functions:

Function		Description
DNP Slave Response Functions	Confirm	Message fragment confirmation used in meter responses. No response to this message is required.
	Response	Meter responds to a master request message.
DNP Master Transfer Functions	Confirm	Message fragment confirmation used in meter responses. No response to this message is required.
	Read	Master requests particular objects from meter; meter responds with requested objects that are available.
DNP Master Freeze Functions	Immediate Freeze	Meter copies the specified objects to a freeze buffer and responds with status of the operation.
	Immediate Freeze-No Acknowledge	Meter copies the specified objects to a freeze buffer but does not respond with a status message.
	Freeze and Clear	Meter copies the specified objects to a freeze buffer then clears the objects and responds with status of the operation.
	Freeze and Clear-No Acknowledge	Meter copies the specified objects to a freeze buffer then clears the objects but does not respond with a status message.
DNP Master Time Synchronization Functions	Delay Measurement	Master requests data from the meter in order to calculate the meter's communication delay and use it for time synchronization. The meter responds with the Time Delay object.

DNP Slave Import Module

The DNP Slave Import module takes DNP objects and imports them as values into ION registers. The DNP Slave Import module takes a DNP Analog Output or Binary Output object and converts it into an ION numeric or Boolean value. The DNP Slave Import module allows the meter to react to control commands from a DNP master.

Various parameters are available through DNP, however importing DNP data to the meter is not a factory-configured option. The possible settings for DNP Slave Import modules are:

Setup Register	Supported Range or Options	Default Setting
DNPPoint	0 to 15	0
DNPObjGrp	Analog Output or Binary Output	Binary Output
RelayMode	1 or 2 points per address	1 point per address

DNPPoint specifies the point number of the DNP object that you want to map to the ION module. *DNPObjGrp* specifies the group of the DNP object and *RelayMode* specifies the method of accessing trip or close relays.

The DNP Slave Import module supports the following functions:

Function		Description
DNP Master Transfer Functions	Confirm	Message fragment confirmation used in master requests. No response to this message is required
	Read	Master requests particular objects from meter; meter responds with requested objects that are available. For Analog Outputs, the value of the AnalogOut output register is returned as the status. For Binary Outputs, the OR'd value of the Relay 1 & Relay 2 output registers are returned as the status. For class 0 polls, status of all Analog and Binary Outputs are returned.
DNP Master Control Functions	Select	Select (or arm) output points (controls, setpoints, analog outputs) at meter but do not activate them; meter responds with the status of the output points selected. When it receives the Select message, the meter starts a timer and must receive an Operate message before the timer expires to activate these outputs.
	Operate	Activate the meter's outputs that were previously selected with the Select function; respond with the status of the outputs.
	Direct Operate	Activate the meter's outputs without a preceding Select message; meter responds with the status of the outputs.
	Direct Operate-No Acknowledge	Activate the meter's output without a preceding Select message; meter does not respond with the status of the outputs.

NOTE

Class 0 polls return DNP status objects for both Analog and Binary Output objects.

Importing DNP data into ION registers:

1. In Designer, use a DNP Slave Import module to take a DNP Analog output or Binary output object and map them into ION registers.
2. There are no input registers for the Import module. You must configure the setup registers to define a DNP object. Use the *ION Reference* to take the setup information and output it to ION registers.

Sample DNP 3.0 Configuration

There is no factory-configured functionality for importing DNP 3.0 data into the meter (although with the DNP Slave Import modules, data can be imported into the meter).

ION modules are linked to DNP Slave Export modules which convert the ION data into the appropriate DNP objects. These objects are available through the meter communications port that is configured to use the DNP 3.0 protocol. The DNP Options module sets global options for all of the DNP Slave Export modules.