PG-141-102-AB Notifier NFS 3030 to BACnet MSTP Protocol Converter

PG-141-102-AB is highly powerful, superior, completely configurable and productive Building & Industrial Automation gateway for integrators to effortlessly interface devices to networks in commercial buildings and industrial plants.

PG-141-102-AB Gateway model supports Notifier NFS 3030 and BACnet MSTP protocols. It is a Bi-directional Converter that can be configured as a Client on Notifier NFS 3030 side and a Server/Client on BACnet MSTP protocol interface.

When configured as a Notifier NFS 3030 client, the PG-141-102-AB can read data from your Notifier NFS 3030 fire panel and publish it as BACnet MSTP data. Also, it can write commands sent from the BACnet MSTP side to the Notifier NFS 3030 fire panel.

When configured as a BACnet MSTP client, the PG-141-102-AB can read data from your BACnet MSTP devices and publish it as Notifier NFS 3030 type data. Also, it can write commands sent from the Notifier NFS 3030 side to the BACnet MSTP devices.

The PG-141-102-AB can be configured to behave as a server on BACnet MSTP interfaces. This mode is useful when data exchange is required between a Notifier NFS 3030 client (for eg. SCADA) and a BACnet MSTP client (for eg. a Building Management System).

The PG-141-102-AB can be configured to behave as a client on both Notifier NFS 3030 and BACnet MSTP interfaces.

PG-141-102-AB gateways have benefitted system integrators worldwide with its powerful line of gateways. Additionally, PG-141-102-AB gateway runs the same protocol conversion software on a productive and cost efficient platform backed by the experience, engineering expertise and technically proven support that integrators have come to expect from PG-141-102-AB.

Features

- Ability to interface upto 1000 points
- DIN rail mount optional
- DIP switches to select baud rate or node ID on the fly
- Multi-configuration capability
- BACnet COV support for fast data communication while reducing the traffic over a BACnet network

Specifications

Environment	Operating Temperature: -40 to 75° C (-40 to 167°F)	
	Relative Humidity:5-90% RH non-condensing	
Power Requirements	9-30 VDC or 12-24 VAC	
	Current Draw @ 12V about 250Ma	
Physical Dimensions(HxWxD)	4.5x2.9x1.6 in. (11.5x7.4x4.1 cm)	
	0.4 lbs (0.2 Kg)	
Other	Configuration/Diagnostic utilities	
	Capacity: 1000 points	
	Table,Wall or DIN rail mount	
Communication Interfaces	RS-485	1
	RS-485 or RS-232	1
	Ethernet 10Base-T, 100BASE-T[2]	1
	Mbus	-
	KNX	-
	LonWorks	-
Approvals	TUV Approved to UL 916 and CSA C22.2 standards	
	BTL and LonMark certified	
	LonMark Certified	
	RoHS Compliant	
	GOST-R Certified	
	CE and FCC	

Notifier NFS 3030 Protocol Driver Description

Formal Driver Type	Serial	
	Passive Client	
Connection Information		
Connection Type:	RS-232 (Vendor Limitation)	
Baud Rates:	9600 (Vendor Limitation)	
Data Bits:	8 (Vendor Limitation)	
Stop Bits:	1 (Vendor Limitation)	
Parity:	None (Vendor Limitation)	
Multidrop Capability:	No	
Devices Tested		
Device	**Tested (Factory, SITE)**	
NFS-3030 Test Panel supplied by Notifier Corp	Factory	
BOOT: 002.003.002 APP: 002.003.014	Site	
BOOT: 002.012.006 APP: 002.013.002	Site	
Connection Facts		
Mode	**Nodes**	**Comments**
Client	1	Each port can connect to only 1 NFS3030 Panel
Server	0	The NFS3030 driver cannot be used as a Server

Data Types Supported

This driver was designed to be connected to the Notifier Onyx NFS-3030 printer port, and listen for incoming messages. The panel's default setting for the printer port is off. To utilize this driver, the printer port must be enabled to 80-columns, unsupervised, before this driver can be used.

The primary purpose of this driver is to record the status of devices connected to the NFS-3030 system by interpreting the text messages sent to the printer port. Not all messages will be interpreted, as many messages do not directly pertain to device status, or are currently supported. The following subset of event messages is recognized:

Active Events:
FIRE ALARM
TROUBLE
PREALARM
SECURITY ALARM
SUPERVISORY
DISABLED
ON/OFF *detectors, modules, panels only*
ACTIVE

A detailed mapping of message interaction System Trouble messages provided by Notifier at the time this driver was written is tabulated in the NFS 3030 Driver Manual. Any changes or additions by Notifier will not be reflected in this driver unless specifically revised.

Zone Status

Information about zone status that is incorporated with point status messages will not be recorded by this driver. A device can belong to multiple zones; however, only the primary zone is listed in printer output. This limits, severely, the accuracy of zone data based on event generated messages, and therefore will not be recorded.

However, zone DISABLED messages will be recorded by the driver as there is no ambiguity in their status.

Panel Status: Data Array Mapping

The status of NFS 3030 devices will be recorded into a series of data arrays within the device, and are available for reading by any other connected device. The data from each loop will be recorded into a separate data array, and a single system array will record system troubles and disabled zones. The structure of the data arrays is provided below.

Most of these arrays will only contain binary information to represent an active or inactive state. However, there could be multiple troubles associated with a single device. For each trouble message, the data array register corresponding to a particular device will be incremented as a counter and decremented when a trouble is cleared.

Parameter	Registers (float)	
{per loop}		
Fire Alarm	0-199 200-399	detectors modules
Trouble *each point will increment/decrement the number of troubles recorded, system normal will reset the counter to zero*	500-799 700-899	detectors modules
PreAlarm	1000-1199 1200-1399	detectors modules
Security Alarm	1500-1799 1700-1899	detectors modules
Supervisory	2000-2199 2200-2399	detectors modules
Disabled	2500-2799 2700-2899	detectors modules
On/Off	3000-3199 3200-3399	detectors modules
Active	3500-3799 3700-3899	detectors modules
{system points only}		
System Troubles	0-1000	
Disabled Zones	1000-1999 2000-2099 2100-2199	General Zones Releasing Zones Trouble Zones
Panel *note: some of these Data Arrays are not appropriate for panels	3000-3099 3100-3199 3200-3299 3300-3399	Fire Alarm Trouble * Security
put arranged in this fashion for symmetry in message parsing	Alarm 3400-3499 3500-3599 3600-3699 3700-3799	* Disabled On/Off *

Port Supervision

The driver is able to process port supervision queries sent by the panel. It has several modes for achieving this.

Mode=1	Driver responds to port supervision queries.
Mode=2	Driver responds to port supervision queries unless it fails to process a message correctly (parsing error). In this case the driver starts a 7 second timer during which time it will not respond to port supervision queries.
Mode=3	Driver accepts the port supervision query but does not respond. This mode is useful for panels where supervision is enabled but no response should be sent.
Mode=4	This is an internal mode. It means the mode is in transition.
Mode=5	Similar to Mode 1 but can be made to transition between mode=3 and mode=5 based on the value in a Data Array. This mode is useful for Hot Standby.

Driver Limitations and Exclusions

- General zone disabling will be recorded, but zone information related to corresponding alarm, trouble, pre-alarm, security alarm, supervisory, and on/off will not be recorded.
- Synchronization between the NFS 3030 Panel and the device can only occur while the panel is in SYSTEM NORMAL mode. At this time the device can be reset.
- Read point status data will not be recorded as this information is not available at the printer port
- The printer port must be enabled on the unit and set to 80 columns with NO supervision unless port supervision is enabled in the driver configuration
- All data related to non-event driven printer reports will not be recorded by the device
- This driver was written specifically for the following Notifier 3030 firmware versions. Any changes or additions by Notifier will not be reflected in this driver unless specifically revised.

Boot:	001.001.001	Dec 03 2002	App:
	001.005.001	Feb 28 2003	

- Information about zone status incorporated with point status messages will not be recorded.
- There can only be one panel connected to any given port
- Data accuracy is dependent on data presented to the printer port by the Notifier NFS3030.

- The driver cannot send any messages (including Ack, Reset and Silence) to the 3030 Panel.

BACnet MS/TP Protocol Driver Description

Driver Name: BACnet/MSTP	Connection type:	RS-485 (Two wire, half-duplex)
	Baud Rates:	9600,19200,38400 and 76800[3]
	Data Bits:	7,8
	Stop Bits:	1,2
	Parity:	Odd, Even, None
	Multidrop Capability:	Yes

PG-141-102-AB AS A BACnet MS/TP CLIENT		
Read Operations Supported	**Properties Supported**	**Comments and Limitations**
Read Property	Present Value	Store value in Data Array location after scaling has been applied
	Out_Of_Service	When using a Complex Data Object, the OOS property is fully supported. Return FALSE when not OOS or when using standard Data Arrays
	Units	Returns Units as specified in the Map Descriptor
	Reliability	When using a Complex Data Objects, returns "Unreliable Other" when the Node is offline, or when the data is old. Returns

		FALSE if the Node is online or when using Standard Data Arrays
	Priority_Array	Returns Priority_Array of Map Descriptor
	Unsupported	This property is supported
	Protocol_Object_Type_Supported	This property is supported
	Protocol_Services_Supported	This property is supported
	Database_Revision	This property is supported and will change if a new configuration is downloaded to the FS
	Max_Master	This property is supported for the BACnet /MSTP DLL option
	Max_Info_Frames	This property is supported for the BACnet/MSTP DLL option
	Relinguish_Default	Returns Relinguish _Default
	Mode	This property is supported.
	Tracking_Value	This property is supported.
Read Property Multiple	As for Read Property	Transactions can be defined to read multiple objects and properties in a single ReadPropertyMultiple operation.

	ALL	Read Property Multiple of the ALL property is NOT supported
Write Operations Supported	**Properties Supported**	**Comments and Limitations**
Write Property	Present Value	Send value in Data Array location after scaling has been applied
Write Property Multiple		
PG-141-102-AB AS A BACnet MS/TP SERVER		
DEVICE OBJECT		
Read Operations Supported	**Properties Supported**	**Comments and Limitations**
Read Property	Object_Identifier	Returns Object _ID with Node_ID as Object Instance
	Object_Name	Returns Node Name
	Object_Type	Returns Device Object type
	System_Status	Returns Normal
	Vendor_Name	Returns PG-10XX Technologies
	Vendor_Identifier	Returns 37
	Model_Name	Returns PG-10XX model
	Firmware_revision	Returns Kernel Version
	Application_sw_version	Returns DCC version
	Protocol_Version	Returns version 1

	Protocol_Revision	Returns version 1
	Protocol_Services_Supported	This property is supported
	Protocol_Object_Type_Supported	This property is supported
	Protocol_Object_List	Returns a list of objects defined in the PG-10XX
	Max_APDU_Length_Accepted	For PG-10XX,the MAX APDU length for BACnet MSTP is 480 bytes and for BACnet IP/BACnet Eth 1497 bytes
	Segmentation_Supported	Returns Segmantation NOT Supported
	APDU_Timeout	Returns the value as defined by the Node's "Timeout" paramater
	APDU_Retries	Returns the value as defined by the Node's "Retries" parameter
	Device_Address_Bindings	Returns an empty list
	Max_Master	This property is supported for the BACnet/MSTP DLL option
	Max_info_Frames	This property is supported for the BACnet/MSTP DLL option
	Description	This property is supported
	Database_Revision	This property is supported and will change if a new configuration is downloaded to the

		PG-10XX
Read Property Multiple	Same properties as Read Property	Read Property Multiple is fully supported. Multiple objects with multiple properties can be specified
Write Operations Supported	**Properties Supported**	**Comments and Limitation**
Write Property	Max_Master	This Property is supported for the BACnet /MSTP DLL option
	Max_info_Frames	This Property is supported for the BACnet /MSTP DLL option
Write Property Multiple	Max_Master	This Property is supported for the BACnet /MSTP DLL option
	Max_info_Frames	This Property is supported for the BACnet /MSTP DLL option
Analog Input Object		
Read Operations Supported	**Properties Supported**	**Comments and Limitations**
Read Property	Object_Identifier	No Limitations
	Object_Name	Returns Map Descriptor Name
	Object_Type	Returns Analog Input Object Type
	Present_Value	Returns value in Data_Array after

		scaling has been applied
	Status_Flags	When using Complex Data Objects returns the FAULT and OUT_OF_SERVICE fields as indicated in section 12.2.7 of the BACnet specification. When using standard Data Arrays returns FALSE for all bits.
	Event_State	No Limitations
	Reliability	When using a Complex Data Objects, returns Unreliable Other when the Node is offline, or when the data is old. Returns FALSE if the node is online or when using Standard Data Arrays
	Out_Of_Service	Fully supported when using a Complex data Object. Returns FALSE when not OOS or when using standard Data Arrays
	Description	This property is supported
	Units	Returns Units as specified in the Map Descriptor
Read Property Multiple	Same properties as Read Property	Read Property Multiple is fully supported. Multiple objects with Multiple properties can be

		specified
Write Operations Supported	**Properties Supported**	**Comments and Limitations**
Write Property	Present_Value	Writing to the Present Value is allowed if the Object is OOS
Write Property Multiple		
Data Sharing Operations Supported	**Properties Supported**	**Comments and Limitations**
SubscribeCOV	Present_value	Subscription storage is non-volatile
COVNotification	Present_value	Confirmed and Unconfirmed
Alarm and Event Operations Supported	**Properties Supported**	**Comments and Limitations**
EventNotification	Present_Value,Status	Confirmed and Unconfirmed
AcknowledgeAlarm		No limitations
Analog Output Object, Analog Value Object		
Read Operations Supported	**Properties Supported**	**Comments and Limitations**
Read Property	Object_Identifier	No Limitations
	Object_Name	Returns “Map Descriptor Name”
	Object_Type	Returns Analog Output Object type
	Present_Value	Returns value in Data Array after scaling has been applied

	Status_Flags	When using Complex Data Objects returns the FAULT and OUT_OF_SERVICE fields as indicated in section 12.2.7 of the BACnet specification. When using standard Data Arrays returns FALSE for all bits
	Event_State	No Limitations
	Reliability	When using a Complex Data Objects, returns “Unreliable Other” when the Node is offline, or when the data is old. Returns FALSE if the Node is online or when using Standard Data Arrays
	Out_Of_Service	Fully supported when using a Complex Data Object. Returns FALSE when not OOS or when using standard Data Arrays
	Units	Returns Units as specified in the Map Descriptor
	Priority_Array	Returns Priority_Array of Map Descriptor
	Description	This property is supported
	Relinguish_Default	Returns Religuish _Default
Read Property Multiple	Same properties as Read Property	Read Property Multiple is fully supported. Multiple

		objects with multiple properties can be specified
Write Operations Supported	Properties Supported	Comments and Limitations
Write Property	Present_Value	When using Complex Data Objects and OOS is TRUE, then the write will not cause a write-through operation to the Server side. If the OOS is FALSE or when using standard Data Arrays then writes will always cause a write-through operation to the Server side
Write Property Multiple		
Data Sharing Operations Supported	**Properties Supported**	**Comments and Limitations**
SubscribeCOV	Present_Value	Subscription storage is non-volatile
COVNotification	Present_Value	Confirmed and Unconfirmed
Alarm and Event Operations Supported	**Properties Supported**	**Comments and Limitations**
Event Notification	Present_Value, Status	Confirmed and Unconfirmed
AcknowledgeAlarm		No Limitations
Binary Input Object		

Read Operations Supported	Properties Supported	Comments and Limitations
Read Property	Object_Identifier	No Limitations
	Object_Name	Returns “Map Descriptor Name”
	Object_Type	Returns Analog Input Object type
	Present_Value	Returns the binary value in the data array
	Status_Flags	When using Complex Data Objects returns the FAULT and OUT_OF_SERVICE fields as indicated in section 12.2.7 of the BACnet specification. When using standard Data Arrays returns FALSE for all bits
	Event_State	No Limitations
	Reliability	When using a Complex Data Objects, returns “Unreliable Other” when the Node is offline, or when the data is old. Returns FALSE if the Node is online or when using Standard Data Arrays
	Out_Of_Service	Fully supported when using Complex Data Object. Returns FALSE when not OOS or when using standard Data Arrays
	Polarity	Always returns “Normal”

	Active_Text	Returns Active Text as specified on the Map Descriptor
	Description	This property is supported
	Inactive_Text	Returns Inactive Text as specified on the Map Descriptor
Read Property Multiple	Same properties as Read Property	Read property Multiple is fully supported. Multiple objects with multiple properties can be specified
Write Operations Supported	**Properties Supported**	**Comments and Limitations**
Write Property	Present_Value	Writing to the Present Value is allowed if the Object is OOS
Write Property Multiple		
Data Sharing Operations Supported	**Properties Supported**	**Comments and Limitations**
SubscribeCOV	Present_Value	Subscription storage is non-volatile
COVNotification	Present_Value	Confirmed and Unconfirmed
Alarm and Event Operations Supported	**Properties Supported**	**Comments and Limitations**

Event Notification	Present_Value, Status	Confirmed and Unconfirmed
AcknowledgeAlarm		No Limitations
Binary Output Object, Binary Value Object		
Read Operations Supported	**Properties Supported**	**Comments and Limitations**
Read Property	Object_Identifier	No Limitations
	Object_Name	Returns “Map Descriptor Name”
	Object_Type	Returns Analog Input Object type
	Present_Value	Returns the binary value in the data array
	Status_Flags	When using Complex Data Objects returns the FAULT and OUT_OF_SERVICE fields as indicated in section 12.2.7 of the BACnet specification. When using standard Data Arrays returns FALSE for all bits
	Event_State	No Limitations
	Reliability	When using a Complex Data Objects, returns “Unreliable Other” when the Node is offline, or when the data is old. Returns FALSE if the Node is online or when using Standard Data Arrays
	Out_Of_Service	Fully supported when using Complex Data

		Object. Returns FALSE when not OOS or when using standard Data Arrays
	Priority_Array	Returns Priority_Array of Map Descriptor
	Religuish_Default	Returns Current Relinguish_Default
	Description	This property is supported
	Active_Text	Returns Active Text as specified on the Map Descriptor
	Inactive_Text	Returns Inactive Text as specified on the Map Descriptor
Read Property Multiple	Same properties as Read Property	Read property Multiple is fully supported. Multiple objects with multiple properties can be specified
Write Operations Supported	**Properties Supported**	**Comments and Limitations**
Write Property	Present_Value	When using Complex Data Objects and OOS is TRUE, then the write will not cause a write-through operation to the downstream side. If the OOS is FALSE or when using standard Data Arrays when writes will always cause a write-through operation to the

		downstream side
Write Property Multiple		
Data Sharing Operations Supported	**Properties Supported**	**Comments and Limitations**
SubscribeCOV	Present_Value	Subscription storage is non-volatile
COVNotification	Present_Value	Confirmed and Unconfirmed
Alarm and Event Operations Supported	**Properties Supported**	**Comments and Limitations**
Event Notification	Present_Value, Status	Confirmed and Unconfirmed
AcknowledgeAlarm		No Limitations
Multiple State Input Object		
Read Operations Supported	**Properties Supported**	**Comments and Limitations**
Read Property	Object_Identifier	No Limitations
	Object_Name	Returns “Map Descriptor Name”
	Object_Type	Returns Analog Input Object type
	Present_Value	Returns unsigned Integer value in the data array
	Status_Flags	When using Complex Data Objects returns the FAULT and OUT_OF_SERVICE fields as indicated in section 12.2.7 of the BACnet specification. When using standard

		Data Arrays returns FALSE for all bits
	Event_State	No Limitations
	Reliability	When using a Complex Data Objects, returns “Unreliable Other” when the Node is offline, or when the data is old. Returns FALSE if the Node is online or when using Standard Data Arrays
	Description	This property is supported
	Out_Of_Service	When using a Complex Data Object, the OOS property is fully supported. Returns FALSE when not OOS or when using standard Data Arrays
	Number_Of_State	When using a Complex Data Object, returns the number of states defined. When using Standard Data Arrays returns the value of 5
	State_Text	When using Complex Data Objects returns the State Text Strings defined. When using Standard Data Arrays return “State_X” where “X” is the value stored in Data_Array and could be 0 to 4

Read Property Multiple	Same properties as Read Property	Read property Multiple is fully supported. Multiple objects with multiple properties can be specified
Write Operations Supported	**Properties Supported**	**Comments and Limitations**
Write Property	Present_Value	Writing to the Present Value is allowed if the Object is OOS
Write Property Multiple		
Data Sharing Operations Supported	**Properties Supported**	**Comments and Limitations**
SubscribeCOV	Present_Value	Subscription storage is non-volatile
COVNotification	Present_Value	Confirmed and Unconfirmed
Alarm and Event Operations Supported	**Properties Supported**	**Comments and Limitations**
Event Notification	Present_Value, Status	Confirmed and Unconfirmed
AcknowledgeAlarm		No Limitations
Multi-State Output Object, Multi-State Value Object		
Read Operations Supported	**Properties Supported**	**Comments and Limitations**
Read Property	Object_Identifier	No Limitations
	Object_Name	Returns “Map Descriptor Name”
	Object_Type	Returns Analog Input Object type

	Present_Value	Returns unsigned Integer value in the data array
	Status_Flags	When using Complex Data Objects returns the FAULT and OUT_OF_SERVICE fields as indicated in section 12.2.7 of the BACnet specification. When using standard Data Arrays returns FALSE for all bits
	Event_State	No Limitations
	Reliability	When using a Complex Data Objects, returns “Unreliable Other” when the Node is offline, or when the data is old. Returns FALSE if the Node is online or when using Standard Data Arrays
	Out_Of_Service	Fully supported when using a Complex Data Object. Returns FALSE when not OOS or when using standard Data Arrays
	Number_Of_State	When using a Complex Data Object, returns the number of states defined. When using Standard Data Arrays returns the value of 5
	State_Text	When using Complex Data Objects returns the State Text Strings defined. When using Standard Data Arrays

		return “State_X” where “X” is the value stored in Data_Array and could be 0 to 4
	Description	This property is supported
	Priority_Array	Returns Priority_Array of Map Descriptor
	Religuish_Default	Returns Relinguish_Default
Read Property Multiple	Same properties as Read Property	Read property Multiple is fully supported. Multiple objects with multiple properties can be specified
Write Operations Supported	**Properties Supported**	**Comments and Limitations**
Write Property	Present_Value	When using Complex Data Objects and OOS is FALSE or when using standard data arrays, writes will trigger a write through operation to client side
Write Property Multiple		
Data Sharing Operations Supported	**Properties Supported**	**Comments and Limitations**
SubscribeCOV	Present_Value	Subscription storage is non-volatile
COVNotification	Present_Value	Confirmed and Unconfirmed

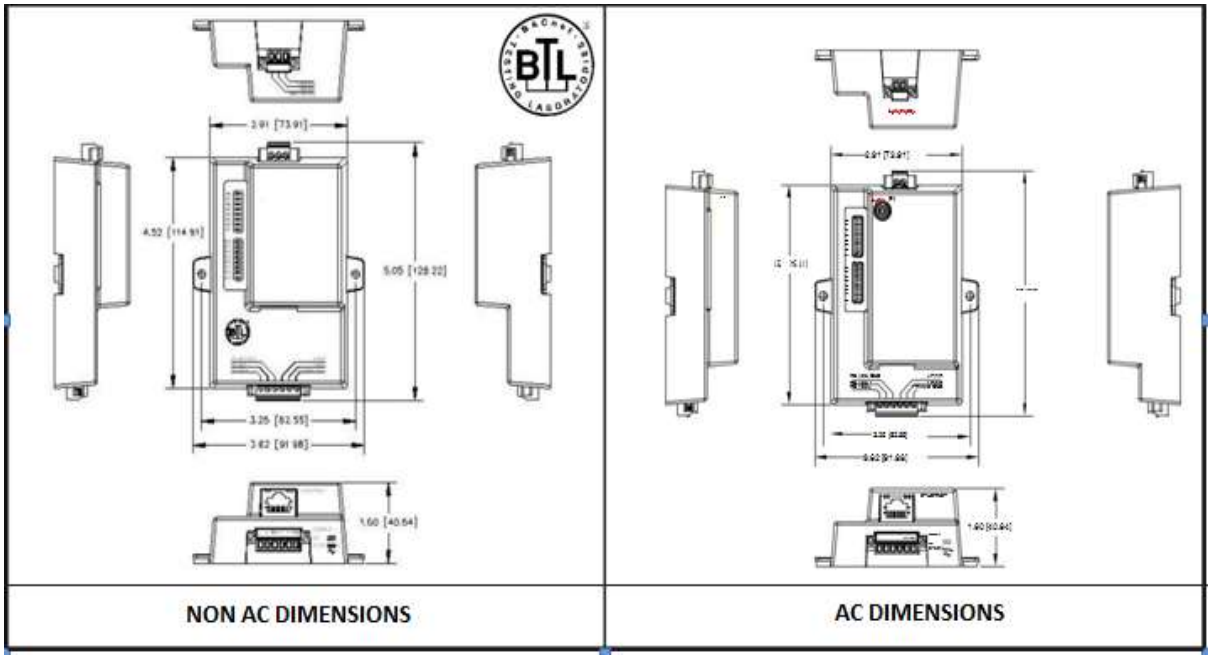
Alarm and Event Operations Supported	Properties Supported	Comments and Limitations
Event Notification	Present_Value, Status	Confirmed and Unconfirmed
AcknowledgeAlarm		No Limitations
Notification Class Object		
Read Operations Supported	**Properties Supported**	**Comments and Limitations**
Read Property	Object_Identifier	No Limitations
	Object_Name	Returns “Map Descriptor Name”
	Object_Type	Returns Notification Class Object type
	Description	No Limitations
	Notification_Class	No Limitations
	Priority	No Limitations
	Ack_Required	No Limitations
	Description	This Property is supported
	Recipient List	No Limitations
Read Property Multiple	Same properties as Read Property	Read property Multiple is fully supported. Multiple objects with multiple properties can be specified
Write Operations Supported	**Properties Supported**	**Comments and Limitations**
Write Property	Recipient_List	RecipientList storage is non-volatile

Write Property Multiple		
AddList	RecipientList	Used to subscribe to Alarm and Event Notifications
Life Safety Point Object		
Read Operations Supported	**Properties Supported**	**Comments and Limitations**
Read Property	Object_Identifier	No limitations.
	Object_Name	Returns "Map Descriptor Name".
	Object_Type	Returns Analog Input Object type.
	Present_Value	Returns unsigned integer value in the Data Array.
	Status_Flags	When using Complex Data Objects returns the FAULT and OUT_OF_SERVICE fields as indicated in the Reference Section of the BACnet specification. When using standard Data Arrays returns FALSE for all bits.
	Event_State	No limitations.
	Reliability	When using a Complex Data Objects, returns “Unreliable Other” when the Node is offline, or when the data is old. Returns FALSE if the Node is online or when using Standard Data Arrays.
	Description	This property is supported.
	Out_Of_Service	When using a Complex Data Object, the OOS property is fully supported. Return FALSE when not OOS or when using standard Data Arrays.
	Mode	Operating Mode. Only 'ON' mode is supported.
	Accepted_Modes	List of Operating Modes
	Silenced	Represents silenced state, but only "All Silenced" supported.
	Operation_Expected	List of LifeSafety Operations, only 'None' operation is supported.
	Property_List	Returns the list of supported properties.
Read Property multiple	Same properties as Read Property	Read Property Multiple is fully supported. Multiple objects with multiple properties can

		be specified.
Write Operations Supported	**Properties Supported**	**Comments and Limitations**
Write Property	Object_Name	Sets Object_Name.
	Present_Value	Writing to the Present Value is allowed if the Object is OOS.
	Mode	Operating Mode. Only 'ON' mode is supported.
	Operation_Expected	List of LifeSafety Operations, only 'None' operation is supported.
Write Property Multiple	Present_Value	Writing to the Present Value is allowed if the Object is OOS.
Data Sharing Operations Supported	**Properties Supported**	**Comments and Limitations**
SubscribeCOV	Present_Value	Subscription storage is non-volatile.
COVNotification	Present_Value	Confirmed and Unconfirmed.
Alarm and Event Operations Supported	**Properties Supported**	**Comments and Limitations**
EventNotification	Present_Value, Status	Confirmed and Unconfirmed.
AcknowledgeAlarm	No Limitations.	
Unsupported Functions and Data Types		
BACnet Object Type not Supported		
Averaging Object		
Calendar Object		
Command Object		
Event Enrollment Object		
File Object		
Group Object		
Life Safety Zone Object		

Loop Object
Notification Class Object unsupported on Client side only
Program Object
Schedule Object
BACnet Services not Supported
Alarm and Event Services unsupported on Client side only
File Access Services
Virtual Terminal Services
COV and EventNotification services are not supported for BACnet
MSTP on the ProtoCessor
For BACnet MSTP , PTP and Arcnet , COV services are disabled by default and may be enabled by setting the Node_Option property to COV_Enable in the Nodes section configuration file.

Dimensions:



NON AC DIMENSIONS	AC DIMENSIONS