inovonics®

Tech note

EN4080 IP Gateway BACnet/IP Quick Start

Introduction

This guide will walk you through the process of communicating with your EN4080 IP gateway using the BACnet protocol. It will assume that you already have a BACnet client and are familiar with its use. This BACnet reporter supports ISO 16484-5, ANSI/ASHRAE 135-2012, specifically revision 14.

Contact Information

For questions, contact Inovonics technical support:

- E-mail: support@inovonics.com.
- Phone: (800) 782-2709; (303) 939-9336.

Materials Needed

- An EN4080 IP gateway.
- A BACnet client.
- A USB flash drive.
- A computer.
- Compatible EchoStream transmitters.

Compatible Transmitters

Only the following transmitters are compatible with an IP gateway using the BACnet protocol.

EN1210 universal transmitter

EN1210W door/window transmitter with reed switch

EN1223D double-button water-resistant pendant

EN1235D double-button belt clip pendant

EN1260 wall mount motion detector

EN1261HT high traffic four element motion detector

EN1501 pulse meter transmitter

EN1501-EXT pulse meter transmitter board for integration

EN1501XL pulse meter transmitter with extended life battery

EN1550EP encoder pulse transmitter

EN1721 integrated temperature/humidity transmitter

EN1722 temperature/humidity transmitter

EN1723 temperature transmitter

EN1751 water detector

EN1752 temperature detector

EN1941 one-way binary RF module

Configure the EN4080 for BACnet

As a best practice, Inovonics recommends pre-configuring and testing your system in a controlled environment before arriving at the site. This will help ensure an efficient installation and provide a means of securing configuration and system logs data for future reference.

This procedure will configure your IP gateway for BACnet:

1. Power up and connect the gateway to the Internet per the *EN4080 IP Gateway Installation Instructions*, available at Inovonics.com.

2. When establishing Internet connectivity ensure the following for your router:
   - **a.** The router must assign addresses via DHCP.
   - **b.** The DHCP lease of the router must be set to a non-zero and non-infinite value, and should be set to a value greater than 30 seconds.
   - **c.** The router must allow outbound connections on the following ports if connecting through a firewall: DNS (UDP 53), HTTPS (TCP 443), NTP (UDP 123), TCP port 9000
   - **d.** Ensure the network is not double NATed, where traffic is flowing through multiple routers that are doing network address translation.

3. Once the green and orange LEDs are solidly lit, allow the IP gateway 20 minutes to check in with the Inovonics cloud to ensure it is using the most current firmware revision, and that your gateway functions properly.

4. Referring to the *EN4080 IP Gateway Installation Instructions*, load the configuration .yaml file and gateway license file onto a blank USB flash drive.

5. Find the en4080-*-config-*.yaml file and rename it to en4080-config.yaml.

6. Open the configuration file on the USB flash drive in Notepad++ or a comparable text editor.

Following is an example of a configuration file, with unrelated lines omitted for clarity:

```
txids:
    - {model: EN1723, name: 'Temp Sensor A', txid:
    10875838, appid: 4200}
    - {model: EN1210W, name: 'Contact Sensor B',
    txid: 2506323, appid: 4210}
```

```
   - {model: EN1501XL, name: 'Pulse Meter Sensor
   C', txid: 10792655, appid: 4220}
bacnet:
   device_instance: 30
   device_name: EN4080 BACnet Reporter Quick Start
   device_description: EchoStream IP Gateway with
   BACnet Interface
   device_location: Room 201, building 2.
   udp_port: 47808
   bbmd_ip: 0.0.0.0
   inactive_period: 6
network:
   ethernet:
      power: enabled
      address: 192.168.90.50
      netmask: 255.255.254.0
      gateway: 192.168.90.1
      dns: 192.168.1.35
   wifi:
      country: US
      power: disabled
      ssid: Guest
      passphrase: Welcome
      address: dhcp
```

7. Enter the BACnet device configuration information in the bacnet section.

- device_instance: This is the instance number of the IP gateway in BACnet.
- device_name: This is the name of the IP gateway in BACnet.
- device_description: A brief description for the gateway.
- device_location: Where the gateway is located.
- udp_port: This is the port number that the reporter will bind to.
- bbmd_ip: This is the IP address of the BACnet broadcast management device, if it exists.
- inactive_period: This is how long in hours since a transmitter has been last heard from before it is declared inactive.

8. Enter configuration information for each of the registered transmitters in the txid section.

Note: When entering your configuration information make sure to follow the example formatting precisely, with no extra spaces, characters, punctuation or leading zeros, using exact model number designations.

- model: This is the Inovonics model number of the transmitter without any dashes. For example, enter the EN1501-EXT as EN1501EXT.
- name: This is the object name of the transmitter.

- txid: This is the eight-digit transmitter identification number (txid) shown on the barcode label inside the housing. Typically the first two digits are in a smaller font.
- appid: This is used at the BACnet object id, and should be any unique integer value for each transmitter.

Note: Some transmitters have multiple outputs, each of which is assigned an incrementing object id starting at the given appid; make sure to leave sufficient room before the next appid. For instance, use an appid that ends in zero, separated from other appids by at least ten, to avoid overlaps and easily map inputs to transmitters.

9. Enter the network configuration information in the network section.

Note: If the network section is not present, wired Ethernet is configured for DHCP IP address assignment by default, and the WiFi is disabled.

Note: If using WiFi only, set Ethernet to disabled. Otherwise the configuration file will attempt to make a connection via Ethernet as the primary, with WiFi as a backup.

- Configure the wired Ethernet interface in the ethernet section:
  – power: Controls if wired Ethernet is to be enabled or disabled.
  – address: This is the static IP address the gateway will use for its wired Ethernet port. To use DHCP for IP address assignment and network interface configuration, specify a value of dhcp; in this case, the following three settings can be omitted.
  – netmask: The network mask for the attached network.
  – gateway: The gateway of the attached network.
  – dns: The address of the desired DNS server.
- Configure the WiFi interface in the wifi section:
  – country: The two character ISO country code in which the gateway resides; e.g. US for the United States.

Note: See https://en.wikipedia.org/wiki/ISO_3166-1_alpha-2 for a full list of ISO country codes.

  – power: Controls if 2.4GHz WiFi is to be enabled or disabled.
  – ssid: This is the name of the WiFi network to connect to.
  – passphrase: The passphrase of the WPA/2 Wifi network to connect to. Leave empty for an open network.
  – address: The static IP address the gateway will use for its wired Ethernet port. To use DHCP for IP address assignment and network interface configuration, specify a value of dhcp; in this case, the following three settings can be omitted.
  – netmask: The network mask for the attached network.
  – gateway: The gateway of the attached network.
  – dns: The address of the desired DNS server.

10. Refer to the IP gateway’s installation instructions to save and reload the configuration file onto the IP gateway.

Note: If the gateway's red LED is not solidly lit after several minutes, the yaml file did not load. Please check formatting, ensure you are using Notepad ++ or a similar yaml editor and use a yaml validator such as yamllint.com.

11. Connect the gateway to your BACnet client.

12. Ensure the IP gateway and all transmitters are reporting in correctly.

- If transmitters are not reporting in as expected, review your configuration file to ensure you are using compatible model numbers, there are no duplicates, and that TXID, BACnet and network data are properly entered.
- If your configuration file is correct and you still don’t see transmitters reporting in correctly, please contact Inovonics technical support to assist you with retrieving systems logs.

13. Save the configuration file onto your PC or an appropriate shared drive for future reference.

14. To archive system logs for each site for future reference, complete the following:

a. Contact Inovonics technical support for a system-logs file.
b. Save the file onto a USB drive.
c. Reset the IP gateway.
d. When the IP gateway’s green LED starts flashing, put the USB drive in the gateway.
e. When the green LED turns solid, remove the USB drive.
f. Load the USB drive onto a PC and copy the log file onto the PC.

- The log file will have a name such as en4080-1e-logs-20200220-204510.zip, with the -1e- being the last two digits of the IP gateway's MAC address.

g. Rename the log file as desired to reflect the site name.

Note: Unzipping the file will create a /data directory with /supervisor and /syslog.

Note: Inovonics can assist in determining that your configuration file is correct and the gateway is outputting BACnet compatible data. For questions on how data from Inovonics hardware appears in your BACnet client, please contact your BACnet client vendor.

15. Label transmitters, the IP gateway and, if used, any repeaters to assist with installation. For example label transmitters with install location and what they are monitoring.

16. Install the hardware on site, making sure to adjust any network settings per each sites' specific network requirements.

Adding to an Existing Site

Due to build out schedules, remodels and other factors, you may have to modify an existing site. To do so, please follow the steps below:

1. Open the saved configuration file for the site in question.
2. Add new transmitters to the configuration file, being careful to not alter existing configuration file data.
   - Swap out a transmitter by overwriting the old TXID with a new TXID.
   - Remove transmitters by deleting specific configuration data.
3. Save the configuration file.
4. Referring to the IP gateway's installation instructions, load the configuration file.

Mapping EchoStream Status to BACnet

Following is a description of how the information provided by EchoStream transmitters to the IP gateway is presented in BACnet properties.

This information is provided to help properly identify Inovonics devices in your BACnet client. Please contact your BACnet client vendor with questions about how Inovonics hardware appears in your BACnet client.

Note: In larger sites which require repeaters, the repeaters will only rebroadcast messages from transmitters, and will not be mapped to your BACnet client. Monitoring of system health within your BACnet client will occur through supervision of the transmitters and the IP gateway.

Standard Properties

Each standard BACnet object instance will support all of the required properties of those objects, in addition to Description and Reliability.

Property values other than Present_Value are derived as follows:

- Object_Identifier: The object's standard object type and instance.
- Object_Type: The object's standard object type.
- Object_Name: Writable by user via BACnet or USB config.
- Description: Not user writable, as used to communicate EchoStream specific values: <MODEL> | <TXID> | <LAST HEARD>.
  – MODEL: This is the Inovonics model number for the transmitter.

  - – TXID: This is the id of the transmitter, as shown in large text on the barcode label inside the transmitter.
  - – LAST HEARD: This is an ISO8601 timestamp, UTC timezone, of when the transmitter was last heard from; or 1970-01-01T00:00:00.000000 if never heard from.
  - – An example description: EN1723 | 10875838 | 2019-01-04T23:16:50.050552
- Status_Flags: Always {FALSE, Y, FALSE, FALSE}; see below.
- Event_State: Always NORMAL.
- Reliability: See below.
- Out_Of_Service: Always FALSE.
- Property_List: As required.

Status Flags

Status_Flags IN_ALARM(X) bit is FALSE because we do not support event reporting. Reliability is NO_FAULT_DETECTED unless the status associated with each transmitter is something other than NORMAL. In these cases, the status will correspond to these values for Reliability.

- If the transmitter or receiver inactive status bit is set, then the status bit will be set to 7 (unreliable-other).
- If the transmitter lowBattery status bit is set, then the status bit will be set to 7 (unreliable-other).
- If the transmitter tamper status bit is set, then the status bit will be set to 7 (unreliable-other).
- If the transmitter or receiver reset status bit is set, then the status bit will be set to 7 (unreliable-other).
- If the receiver jammed status bit is set, then the status bit will be set to 7 (unreliable-other).
- If all the above status bits are cleared, regardless if any other transmitter or receiver status bits not listed above are set, then the status bits will be set to (no-fault-detected).

Status_Flags FAULT(Y) bit will generally be FALSE unless Reliability is not NO_FAULT_DETECTED.

To differentiate which status bits are set or cleared, consult the BitString Value (BSV) object associated with each device. This object includes an array of status bit states and map to an array of status bit string names. For example:

If the Present_Value of a BSV object is: [0, 1, 0, 0, 0, 1, 1, 0] and the Bit_Text of a BSV object is: ['inactive', 'rapidTransmissionMode', 'deltaTotalizer', 'shipping', 'lowBattery', 'tamper', 'supervision', 'reset'] then the device is in rapid transmit mode, tampered, and the last message heard was a supervision message.

EN1210, EN1210W, EN1223D, EN1235D, EN1260 and EN1261HT Transmitters

Objects for these transmitters generate from 1 to 4 Binary Input (BI) object instances, depending on the number of alarms supported by the device. For example, the EN1210W supports two alarm flags. For these objects, the Present_Value of the object corresponds with the respective alarm status bit. Each BI will support these standard properties where different from those described above:

- Present_Value: Tracks alarm status returning 0 (restore) or 1 (alarm).
- Tracking_Value: Same as Present_Value.
- Polarity: Always NORMAL.
- Inactive_Text: Normal.
- Active_Text: Alarm.

Each BI instance will have a corresponding BitString Value (BSV) object instance. The purpose of the BSV object is to indicate the collection of statuses associated with the BI transmitter. Each BSV will support these standard properties where different from those described above:

- Object_Name: The same as BI with Status appended.
- Present_Value: Indicates the collection of status bits.
- Bit_Text: An array of strings that varies according to the relevant status bits returned by each BI:
  - – inactive: Set if no message has been heard from the transmitter recently.
  - – alarm0: Set if the primary alarm input is activated.
  - – alarm1, ...: Set if the additional alarm inputs are activated (EN1210W reed switch).
  - – lowBattery: Set if the transmitter's battery is nearly depleted.
  - – tamper: Set if the transmitter's housing has been opened.
  - – supervision: Set if the last message received indicates no change in status; in this case, it was simply a check-in.
  - – reset: Set if the last message received indicates the transmitter was just reset or powered up.

EN1501, EN1501-EXT, EN1501XL and EN1550EP Transmitters

Objects for the pulse meter transmitter (PMT) generate one Analog Value (AV) object instance. For these objects, the Present_Value of the object corresponds with the returned pulse count. The counts from these PMTs are positive integer values. AV objects, a float point value, are used for broad compatibility with BACnet client and controllers. For the range of PMT counts, zero to one million, the single precision floating point number is sufficient to represent counts without loss of accuracy (i.e. there are never more than six significant decimal digits). Each AV shall support these standard properties where different from those described above:

- Present_Value: Tracks pulse count.
- COV_Increment: Writeable with zero or greater integer or float numerical value; delta number of counts required before COV notification sent.

Each AV instance shall have a corresponding BitString Value (BSV) object instance. The purpose of the BSV object shall be to indicate the collection of statuses associated with the BI transmitter. Each BSV shall support these standard properties where different from those described above:

- Object_Name: The same as AV with Status appended.
- Present_Value: Indicates the collection of status bits.
- Bit_Text: An array of strings that varies according to the relevant status bits returned by each AV.
  - inactive: Set if no message has been heard from the transmitter recently.
  - rapidTransmissionMode: Set if transmitter is sending a check-in message often. This is useful during installation. After 72 hours, this clears and check-in messages revert to once per hour.
  - deltaTotalizer: Set if ten counts or more have been recorded since the last message sent.
  - shipping: Set if this is the last message before the transmitter goes to sleep and stops transmitting. The transmitter will not wake until the reset button is pressed.
  - lowBattery: Set if the transmitter's battery is nearly depleted.
  - tamper: Set if the transmitter's housing has been opened.
  - supervision: Set if the last message received indicates no change in status; in this case, it was simply a check-in.
  - reset: Set if the last message received indicates the transmitter was just reset or powered up.

Note: While the EN1501-EXT contains a dash in the model number, do not include the dash when specifying this model number in configuration, entering it instead as EN1501EXT.

EN1721, EN1722 and EN1723 Transmitters

Objects for these transmitters generate two Analog Input (AI) object instances, depending on the number of inputs supported by the device. For example, the EN1723 supports two temperature sensors, while the EN1722 supports one temperature sensor and one relative humidity sensor. For these objects, the Present_Value of the object corresponds with the sensor reading. Each AI will support these standard properties where different from those described above:

- Present_Value: The sensor value reported in the transmitter message; either Fº or Cº, relative humidity percentage or ohms resistance.
- Tracking_Value: Same as Present_Value.
- Units: This is writeable, and defaults to Fº for integrated temperature sensors (EN1723 internal sensor, EN1721 and EN1722 internal and external sensor), percentage for relative humidity and ohms for external temperature sensors (EN1723 external sensor).
- Min_Pres_Value: This is writable, but is not used by the reporter; use for reference only.
- Max_Pres_Value: This is writable, but is not used by the reporter; use for reference only.

- COV_Increment: Writeable with zero or greater integer or float numerical value; represents the delta change in measured sensor value required before COV notification is sent.

Each AI instance will have a corresponding BitString Value (BSV) object instance. The purpose of the BSV object is to indicate the collection of statuses associated with the BI transmitter. Each BSV will support these standard properties where different from those described above:

- Object_Name: The same as AI with Status appended.
- Present_Value: Indicates the collection of status bits.
- Bit_Text: An array of strings that varies according to the relevant status bits returned by each AI.
  - inactive: Set if no message has been heard from the transmitter recently.
  - deltaChannel1: Set if the measured sensor one (EN1723 internal sensor, EN1721 and EN1722 temperature sensor) value has changed by more than the transmitter’s configured threshold; defaults to 0.5.
  - deltaChannel2: Set if the measured channel two value (EN1723 internal sensor, EN1721 and EN1722 humidity sensor) has changed by more than the transmitter’s configured threshold; defaults to 0.5.
  - lowBattery: Set if the transmitter's battery is nearly depleted.
  - supervision: Set if the last message received indicates no change in status; in this case, it was simply a check-in.
  - reset: Set if the last message received indicates the transmitter was just reset or powered up.

EN1751 and EN1752 Transmitters

Objects for these transmitters generate two or three Binary Input (BI) object instances, depending on the number of alarms supported by the device. For example, the EN752 supports two alarm flags, one for when a high or low temperature threshold is exceeded. For these objects, the Present_Value of the object corresponds with the respective alarm status bit. Each BI will support these standard properties where different from those described above:

- Present_Value: Tracks the alarm status, returning 0 (restore) or 1 (alarm).
- Tracking_Value: The same as Present_Value
- Polarity: Always NORMAL.
- Inactive_Text: Normal.
- Active_Text: Alarm.

Each BI instance will have a corresponding BitString Value (BSV) object instance. The purpose of the BSV object is to indicate the collection of statuses associated with the BI transmitter. Each BSV will support these standard properties where different from those described above:

- Object_Name: The same as BI with Status appended.
- Present_Value: Indicates the collection of status bits.

- Bit_Text: An array of strings that varies according to the relevant status bits returned by each BI.
  - inactive: Set if no message has been heard from the transmitter recently.
  - alarm0, alarm 1, ...: Set if the respective alarm inputs are activated: EN1751 alarm0 if water detected, alarm1 if temperature below 4º F, alarm 2 if temperature above 140º F; EN1752 alarm if temperature below 40º F by default, alarm1 if temperature above 80º F by default.
  - lowBattery: Set if the transmitter's battery is nearly depleted.
  - tamper: Set if the transmitter's housing has been opened.
  - supervision: Set if the last message received indicates no change in status; in this case, it was simply a check-in.
  - reset: Set if the last message received indicates the transmitter was just reset or powered up.

EN1941 Transmitters

Objects for EN1941 transmitters generate two Binary Input (BI) object instances. For these objects, the Present_Value of the object corresponds with the respective alarm status bit. Each BI will support these standard properties where different from those described above:

- Present_Value: Tracks the alarm status, returning 0 (restore) or 1 (alarm).
- Tracking_Value: The same as Present_Value
- Polarity: Always NORMAL.
- Inactive_Text: Normal.
- Active_Text: Alarm.

Each BI instance will have a corresponding BitString Value (BSV) object instance. The purpose of the BSV object isto indicate the collection of statuses associated with the BI transmitter. Each BSV supports these standard properties where different from those described above:

- Object_Name: The same as BI with Status appended.
- Present_Value: Indicates the collection of status bits.
- Bit_Text: An array of strings that varies according to the relevant status bits returned by each BI.
  - inactive: Set if no message has been heard from the transmitter recently.
  - alarm0, alarm 1: Set if the respective alarm inputs are activated.
  - lowBattery: Set if the transmitter's battery is nearly depleted.
  - tamper: Set if the transmitter's housing has been opened.
  - supervision: Set if the last message received indicates no change in status; in this case, it was simply a check-in.
  - reset: Set if the last message received indicates the transmitter was just reset or powered up.

EN4080 IP Gateway

The object for the EN4080 generates one Binary Input (BI) object instance. For the receiver, the Present_Value of the object corresponds with the respective jammed status bit. Each BI will support these standard properties where different from those described above:

- Object _Identifier: Always 0.
- Present_Value: Tracks the alarm status, returning 0 (restore) or 1 (alarm).
- Tracking_Value: The same as Present_Value
- Polarity: Always NORMAL.
- Inactive_Text: Normal.
- Active_Text: Alarm.

Each BI instance will have a corresponding BitString Value (BSV) object instance. The purpose of the BSV object will be to indicate the collection of statuses associated with the BI transmitter. Each BSV will support these standard properties where different from those described above:

- Object_Name: The same as BI with Status appended.
- Present_Value: Indicates the collection of status bits.
- Bit_Text: An array of strings that varies according to the relevant status bits returned by each BI.
  - inactive: Set if no message has been heard from a transmitter recently.
  - jammed: Set if all channels contain interfering signals above a predetermined level.
  - supervision: Set if the last message received indicates no change in status; in this case, it was simply a check-in.
  - reset: Set if the last message received indicates a transmitter was just reset or powered up.

Change Of Value Subscription

The BACnet reporter maintains a pool of up to 2000 simultaneous COV subscriptions. The BinaryInput, AnalogInput, and AnalogValue objects can be subscribed to with a confirmed or unconfirmed notification. A notification is then sent if the Prevent_Value of the subscribed object changes by more than the current COV_Increment value.

The COV_Increment value is stored as a float and is checked on update to be greater than zero; hence it can be compared with the difference between the new and old values in the RF message received, to determine if a notification needs to be sent. The COV_Increment property can be written to with a float or an integer data type, but the numerical value must be zero or greater.

Note: Subscribing to BitString Value objects, or any specific property of any object, is not supported.

Inactive Status

Periodically, the last heard date/time stamp of each transmitter is evaluated, and, if too far in the past, the inactive status bit is set. A transmitter that has never been heard from is marked as inactive at startup of the IP gateway. When a message is heard from the transmitter, its inactive status bit is cleared.

Known Limitations

This release of the BACnet reporter for the EN4080 provides common functionality, though not all features of BACnet are supported. Limitations to be aware of include:

- After writing to device or object properties via BACnet, allow 20 minutes for changes to be reflected before power cycling or otherwise resetting the gateway; otherwise changes may be lost.
- It is not possible to set the IP gateway's clock with BACnet commands; the IP gateway's clock is sync'd using NTP to inovonics.pool.ntp.org.
- It is not possible to subscribe to BitString Value objects, or any specific property of any object, for COV notifications.
- It is not possible to configure Units, Min_Pres_Value, Max_Pres_Value nor COV_Increment via the USB config file; they can only be configured via BACnet property writes.
- For submetering applications, the TXID used in USB config file en4080-config.yaml refers to the transmitter’s serial number, which is the number printed above the barcode on the label inside the transmitter housing. The submetering ten-digit TXID on the label on the outside of the housing of the EN1501 will be ignored if used in the config file.

The TXID used in the USB config file en4080-config.yaml refers to the transmitter's serial number above the barcode on the printed circuit board. Users can disregard the submetering 10-digit TXID label affixed to the outside top of the case.

The model used in the USB config file must match that of the associated transmitter and is limited solely to the compatible transmitters. If such a mismatch occurs, a message will be written to the system log to assist in diagnostics.

If you encounter further issues, limitations or missing features with the IP gateway BACnet reporter, please contact Inovonics technical support. They will be taken under consideration and used to shape the future production release of the BACnet reporter.