



Tech note

---

E\*4080 IP Gateway BACnet/IP Quick Start

## Introduction

---

This guide will walk you through the process of communicating with your EE4080 or 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](mailto:support@inovonics.com).
- Phone: (800) 782-2709; (303) 939-9336.

## Materials Needed

---

- An EE4080 or EN4080 IP gateway.
- A BACnet client.
- A USB flash drive.
- A computer.
- Compatible EchoStream transmitters.

## Compatible Transmitters

---

The following transmitters are compatible with an IP gateway using the BACnet protocol. Up to 100 end devices can be supported.

“EN” indicates EchoStream products that operate at 900 MHz only; “EE” indicates EchoStream products that operate at 868 MHz only; “E\*” represents products available in both EE and EN versions. For example, E\*4080.

EN1210 universal transmitter

EN1210W door/window transmitter with reed switch

E\*1223D double-button water-resistant pendant

E\*1235D double-button belt clip pendant

EN1260 wall mount motion detector

EE1261 four element motion detector

EN1261HT high traffic four element motion detector

EN1501 pulse meter transmitter

EN1702 current/voltage data transmitter

EN1721 integrated temperature/humidity transmitter

E\*1722 temperature/humidity transmitter  
E\*1723 temperature transmitter  
EN1751 water detector  
EN1752 temperature detector  
E\*1941 one-way binary RF module

## Configure the E\*4080 for BACnet

---

This procedure will configure your IP gateway for BACnet:

1. Referring to the *EE4080 IP Gateway Installation Instructions* or *EN4080 IP Gateway Installation Instructions*, load the configuration .yaml file and gateway license file onto a blank USB flash drive.
2. Find the en4080-\*-config-\*.yaml file and rename it to en4080-config.yaml.
3. Open the configuration file on the USB flash drive in any 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: EN1501, name: 'Pulse Meter Sensor C',
    txid: 10792655, appid: 4220}
bacnet:
  device_instance: 42
  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
```

4. 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.
5. Enter configuration information for each of the registered transmitters in the txid section.
  - model: This is the Inovonics model number of the transmitter.
  - name: This is the object name of the transmitter.
  - txid: This is the txid of the transmitter as shown in large text on the barcode label inside the housing.
  - appid: This is used at the BACnet object id, and should be unique 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.

---

6. 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.

---

- 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](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.
7. Referring to the IP gateway's installation instructions, use the USB flash drive to boot the IP gateway.
- The IP gateway should now respond to BACnet communications.

## 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.

### 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. For example, Ex1723 for both EN1723 and EE1723 transmitters, or Ex1261 for both EN1261HT and EE1261 transmitters.
  - 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: E\*1723 | 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.

## EN1210, EN1210W, E\*1223D, E\*1235D, EN1260, EE1261 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 Transmitters

Objects for the EN1501 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.

## EN1702 Transmitters

Objects for the EN1702 transmitter generate two Analog Input (AI) object instances. For these objects, the Present\_Value of the object corresponds with the channel reading. Each AI will support these standard properties where different from those described above:

- Present\_Value: This is the measured channel value reported in the transmitter message, in either volts or milliamps.
- Tracking\_Value: This is the same as Present\_Value.
- Units: This is writable; it defaults to volts, and should be updated if the transmitter is configured via jumpers to measure milliamps.
- 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 channel value required before COV notification.

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 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.
  - deltaChannel1: Set if the measured channel one value has changed by more than the transmitter's configured threshold; defaults to 0.5.
  - deltaChannel2: Set if the measured channel two value 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.

## EN1721, E\*1722 and E\*1723 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 E\*1723 supports two temperature sensors, while the E\*1722 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 F° or C°, relative humidity percentage or ohms resistance.
- Tracking\_Value: Same as Present\_Value.
- Units: This is writeable, and defaults to F° for integrated temperature sensors (E\*1723 internal sensor, EN1721 and E\*1722 internal and external sensor), percentage for relative humidity and ohms for external temperature sensors (E\*1723 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 (E\*1723 internal sensor, EN1721 and E\*1722 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 (E\*1723 internal sensor, EN1721 and E\*1722 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 4° F, alarm 2 if temperature above 140° F; EN1752 alarm if temperature below 40° F by default, alarm1 if temperature above 80° 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.

## E\*1941 Transmitters

Objects for E\*194x 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 is to 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.

## **E\*4080 Receivers**

The object for the E\*4080 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](http://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.

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.