

BACnet On ThermaViewer 2

DRAFT 0.14

October 28~~17~~, 2020

Table of Contents

Introduction.....	4
1 Thermaviewer 2 (TV2).....	5
2 How to Use BACnet on the TV2.....	6
2.1 Overview.....	6
2.2 Querying TV2 Settings.....	6
2.3 Modifying TV2 Settings.....	6
2.4 TV2 Alarms.....	7
2.5 Real Time Clock.....	8
3 BACnet Objects.....	9
3.1 Device Object.....	9
3.2 Analog Input Objects.....	12
3.3 Binary Input Objects.....	15
3.4 Binary Value Object.....	17
3.5 Trend Log Objects.....	18
3.6 Notification Class Object.....	20
3.7 Network Port Object.....	21
4 BACnet Services.....	24
4.1 ReadProperty.....	24
4.2 ReadPropertyMultiple.....	24
4.3 ReadRange.....	24
4.3.1 Determining Time Based on Stored Data.....	26
4.3.2 Special Codes.....	27
4.4 WriteProperty.....	28
4.5 ReinitializeDevice.....	28
4.6 Who-Is/I-Am.....	29
4.7 Who-Has/I-Have.....	29
4.8 GetEventInformation.....	29
4.9 UnconfirmedEventNotification.....	30
4.10 UnconfirmedCOVNotification.....	30
4.11 TimeSynchronization.....	30
5 BACnet Interoperability Building Blocks (BIBBs) and Protocol Implementation Conformance Statements (PICS).....	32
5.1 Resources.....	33
5.2 TV2 Protocol Implementation Conformance Statement.....	33
6 Revisions.....	37

Introduction

This document will give an overview of BACnet/IP as implemented on the Thermaviewer 2.

BACnet (Building Automation Control network) is a data communication protocol developed by ASHRAE for the purpose of standardizing communications between devices in a building automation and control network.

BACnet/IP specifically is an Ethernet based protocol that uses UDP to transfer data between devices.

BACnet models each physical device as a collection of data structures called "objects," the properties of which represent various aspects of the hardware, software, and operation of the device. These objects provide a means of identifying and accessing information without requiring knowledge of the details of the device's internal design or configuration.

BACnet provides a collection of "services" which allows for the access and manipulation of the "objects" and their properties.

In BACnet parlance the Thermaviewer 2 is a BACnet server device that provides information and access to clients via the BACnet protocol. The Thermaviewer 2 mainly listens and responds. In some [instances](#), it will notify the devices on the network to which it is attached of events that have occurred if it is so configured.

This document will provide a description of the objects and services that are defined for the Thermaviewer 2 and how to use them.

1 Thermaviewer 2 (TV2)

The Thermaviewer 2 has four ports that each support up to two sensors used to monitor temperature, pressure, or humidity.

Each sensor supports an alarm that can signal that the reading has gone outside of a defined range or if the sensor has become disconnected.

The sensor readings are logged in an internal flash.

The Thermaviewer 2 supports the following function:

- The TV2 has 4 input ports. Each port has two channels that can support a single physical sensor. There is a single physical connection for the 2 sensor channels.
- The TV2 can support wired or wireless sensors.
- The TV2 supports the following types of sensors:
 - Temperature
 - Thermistor
 - Temperature/Humidity
 - Pressure
- Each sensor supports an Out of Range Alarm with the following programmable parameters:
 - High limit
 - Low limit
 - Time alarm must exceed limit before alarm is triggered
 - Duration of time for alarm to sound
- Each port supports an alarm that sounds if the attached sensors have become disconnected.
- The TV2 has an alarm that signals if the device has become disconnected from its main power source.
- The data for each sensor is sampled at a programmable rate and stored in an onboard flash.
- The TV2 supports wired Ethernet or Wi-Fi communication.

2 How to Use BACnet on the TV2

This section will describe how to perform various tasks using BACnet on the TV2. The following sections will go into more detail about BACnet objects and services.

2.1 Overview

BACnet allows the user to remotely query and control the TV2 device using a known standard interface. On the TV2 the flavor of BACnet used is BACnet/IP which communicates over wired or wireless ethernet using UDP. (For simplicity this document will refer to BACnet which implies BACnet/IP.)

A main consideration for adding BACnet function to the TV2 was to allow a remote user the same level of access as a user that had direct access to the TV2 device. That means that if there is a password set and enabled on the TV2 then a remote user would be limited to what they could change.

Note that despite the desire of providing the same access remotely, because of lack of security in BACnet and lack of resources in the TV2, some functionality is not available remotely. This function is limited to the network controls that, if exposed via BACnet, could give someone access to the network itself. Therefore, the TV2's network settings cannot be changed remotely via BACnet.

2.2 Querying TV2 Settings

Almost all of the TV2 settings that can be seen on the TV2 menus can be queried at any time without a password via BACnet. The exceptions to that are the WiFi settings and the Calibration Menu settings.

There are currently no WiFi properties defined in BACnet and the level of security is too low to allow those settings to be accessed at this time via BACnet.

There are no properties defined in BACnet that mirror the Calibration settings and with the limited resources available on the TV2 it was decided not to expose them as well.

The rest of the menu settings are available via BACnet for querying whether the password is enabled or disabled.

Most settings can be access using the ReadProperty or ReadPropertyMultiple service.

The one thing that cannot be accessed via those services is the sample data stored in the TV2 flash. That data is contained in the Trend Log objects in the Log Buffer which can only be read using the ReadRange service.

2.3 Modifying TV2 Settings

TV2 settings are modified in BACnet using the WriteProperty service.

If the TV2 password is enabled (Device Password Enabled property set to TRUE), then most settings cannot be modified using WriteProperty. To modify the TV2 settings using BACnet, the Device Password Enabled property must be set to FALSE.

The two exceptions to that are the Device Password Enabled and Device Password properties. These two properties can be provisionally written while the Device Password Enabled property is TRUE but are not activated until a proper ReinitializeDevice action is performed. The sequence to modify either of these properties is as follows:

1. Using WriteProperty change the value of Password Enabled or Password property.
2. Next initiate a ReinitializeDevice using the current (not the modified) password.
3. When the TV2 restarts, the targeted property will be changed.

Note that only one of these properties can be changed at a time. Performing a second WriteProperty after step 1 will invalidate the first pending change.

Once the Device Password Enabled property is FALSE, then the WriteProperty service can be used to modify all other writable properties.

2.4 TV2 Alarms

The TV2 has several alarms defined that can be enabled to alert the user of certain events that have occurred. In BACnet these TV2 alarms are referred to as "events".

BACnet supports both Alarms and Events but a reading of the BACnet specifications indicates that the TV2 alarms are best characterized as BACnet events. From the BACnet spec section 13, "...In general, "events" are changes of value of certain properties of certain objects, or internal status changes, that meet predetermined criteria. There are two mechanisms provided in BACnet for managing events: change of value reporting and event reporting." This would also describe TV2 alarms.

In BACnet the distinction between and Event and an Alarm is found in the following statement from the BACnet spec section 13.1,

"Events may be selectively identified as belonging to the category of "alarms" or "events." Event-initiating objects indicate this distinction through the Notify Type property. Conceptually, alarms are events that are intended to be seen and reacted to by human operators... In the BACnet protocol, the singular distinction is that alarms will be reported by the GetAlarmSummary service, while all other events will not. In every other respect, BACnet makes no distinction between an alarm and an event."

On the surface it would seem to be useful to implement TV2 alarms as BACnet alarms in order to make use of the GetAlarmSummary service, however that service has been deprecated and its use is discouraged.

The following TV2 alarms generate BACnet notifications if they are enabled:

- AC Disconnect Alarm – generates an UnconfirmedCOVNotification.
- Probe Missing Alarm – generates an UnconfirmedCOVNotification
- Sensor Out of Range Alarm – generates an UnconfirmedEventNotification

2.5 Real Time Clock

The TV2's real time clock (RTC) can be modified in BACnet using the TimeSynchronization service.

Because changing the TV2 RTC can cause data to be erased, this service can only change the TV2 time if the Device Password Enabled property is FALSE.

The process to change the TV2 time is as follows:

1. Disable the password by:
 - a. Using WriteProperty to change the value of Password Enabled to FALSE.
 - b. Next initiate a ReinitializeDevice using the current (not the modified) password.
 - c. When the TV2 restarts, the password should be disabled.
2. Send a TimeSynchronization request to synchronize the TV2 with the network time.
3. Reenable the password using the same procedure as used in step 1 above but changing the Password Enabled value to TRUE.

If the Real Time Clock is advanced by less than an hour, then a Special Code is inserted into the flash data for each sample period missed.

If the Real Time Clock is advanced more than an hour, then all of the sample data is erased.

If the Real Time Clock is set back, then all of the sample data is erased.

23 BACnet Objects

The Thermaviewer 2 implementation of BACnet contains the following objects:

- Device – like all BACnet implementations, the TV2 has a single Device object that describes general characteristics.
- Analog Input – the TV2 has 8 Analog Input objects. There is one objects for each sensor channel.
- Binary Input – the TV2 has 5 Binary Input objects. There is one object for each sensor port and one object for the AC power connection.
- Trend Log – the TV2 supports 8 Trend Log objects. There is one object for each sensor channel.
- Binary Value – there is a single Binary Value object that controls whether the buzzer sounds when an alarm is triggered.
- Network Port – there is a single Network Port object that describes the TV2 network connection.
- Notification Class – there is a single Notification Class object that describes how event notifications are distributed.

Each object contains a collection of properties that provide a means for accessing and manipulating information.

Every object contains the following properties:

- Object Identifier ([75](#)) – this property is a unique 32-bit unsigned integer identifier assigned to each object. It consists of the object type and an object instance. The object instance is a 22-bit value unique to the object type in a device.
- Object Type ([79](#)) – this property is an enumeration that defines the type of an object which defines its set of properties.
- Object Name ([77](#)) – this character string is a name that is used to refer to the object. Each object name must be unique for the device in which they exist.
- Property List ([371](#)) – this property is a list of all of the properties that exist within an object. The Object Identifier, Object Type, Object Name and Property List properties are not included in this list.

2.13.1 Device Object

The device object contains properties that describe the general characteristics of the TV2. All device properties except one are readable. Writable properties are indicated in their descriptions below.

There is only one device object.

There is no event reporting within the device object.

The device object contains the following properties:

• [Object Identifier \(75\) – For the device object, the instance of the object is determined by default from the MAC address of the TV2. The instance is the lower 22-bits of the MAC address. The TV2 allows this value to be modified, if needed, using the Network Menu.](#)

- Object Name (77) – the name of this device. This value is the character string “TV2” for all Thermaviewer 2 devices.
- Object Type (79) – this property is the BACnetObjectType enumeration, device (8).
- System Status (112) – this is the current status of the TV2. For the TV2 this value is always the enumeration value, operational (0).
- Vendor Name (121) – this is the character string “Two Dimensional Instruments, LLC”.
- Vendor Identifier (120) – this is a unique unsigned integer value, 1236.
- Model Name (70) – this is the character string “Thermaviewer 2”.
- Firmware Revision (44) – this is a character string that is the current firmware loaded on the TV2. A typical sting is “9.b2d.06”.
- Application SW Revision (12) – this is required by BACnet but has no real meaning for the TV2. Its value is the character string “0”.
- Protocol Version (98) – this unsigned integer is the version number of the BACnet protocol supported by the TV2. This value is currently ‘1’.
- Protocol Revision (139) – this unsigned integer is the revision level of BACnet supported by the TV2. This value is currently ‘19’.
- Protocol Services Supported (97) – this bit string describes the BACnet services supported by the TV2. The current value is 0x948108000000080.
- Protocol Object Types Supported (96) – this bit string describes the BACnet objects supported by the TV2. The current value is 0x000B083CF100.
- Segmentation Supported (107) – this enumeration value indicates the type of segmentation supported by this device. For the TV2 this value is no-segmentation (3).
- Object List (76) – this is a list of all of the supported BACnet objects in the Thermaviewer 2.
- Serial Number (372) – this character string contains the TV2’s serial number.
- Local Time (57) – this property is the current time.
- Local Date (56) – this property is the current date.
- Restart Notification Recipients (202) – this property indicates the devices that are sent notifications of a device restart. For the TV2 this property indicates that notification is sent to all devices on the local network.
- Last Restart Reason (196) – this enumeration indicates the reason for the last device restart. Valid values for the TV2 are:
 - UNKNOWN (0)
 - COLSTART (1) – indicates a Reset to Factory Defaults.
 - WARMSTART (2) – indicates a reboot of the system under software control.
 - DETECT_POWER_LOST (3) – indicates that the device power was lost causing the restart.
 - DETECT_POWERED_OFF (4) – indicates the user powered the TV2 off.
 - SUSPENDED (7) – indicates that the TV2 was put into deep sleep mode.

- ACTIVATE_CHANGES (8) – indicates that the TV2 was rebooted to instantiate a change in the device password.
- Time of Device Restart (203) – this BACnetTimeStamp property is the time at which the device was last restarted.
- Max APDU Length Accepted (62) – this unsigned integer is the maximum size of APDU accepted by the TV2. This value is '1476'.
- APDU Timeout (11) – this property is an unsigned representing the milliseconds between APDU retransmissions. The TV2 does not retransmit, therefore this value is '0'.
- Number of APDU Retries (73) – this property is an unsigned integer that specifies how many retries the device will attempt to send an APDU. The TV2 does not attempt retries and this value is '0'.
- Device Address Binding (30) – this property has no meaning for the TV2 and is an empty list as allowed by the BACnet spec.
- Database Revision (155) – this unsigned integer has no meaning for the TV2 and always has a value of '0'.

The following list contains custom device properties defined only for the TV2:

- Device Password (1200) – Custom property. This character-string is the current password for the TV2.
 - This property is not readable. All attempts to read this property will result in a Read Access Denied error code for the Property error class.
 - This property is writable using the Write Property service.
 - A valid password contains 4 numeric digits.
- Device Password Enabled (1201) – Custom property. This boolean is the current state of the Password Enabled control on the System Menu.
 - This property is writable using the Write Property service.
- Temperature Units (1205) – This enumerated value specifies the units used for temperature values.
 - Valid values are:
 - degrees-fahrenheit (64)
 - degrees-celsius (62)
 - This property is writable.
- Pressure Units (1206) – This enumerated value specifies the units used for pressure values.
 - Valid values are:
 - inches-of-water (58)
 - hectopascals (133)
 - This property is writable.

2.23.2 Analog Input Objects

The analog input object is used to model the sensors that can be connected to the TV2.

There are 8 analog input objects defined for the TV2. These are:

- Analog Input 11 – Port 1, Channel 1 sensor.
- Analog Input 12 – Port 1, Channel 2 sensor.
- Analog Input 21 – Port 2, Channel 1 sensor.
- Analog Input 22 – Port 2, Channel 2 sensor.
- Analog Input 31 – Port 3, Channel 1 sensor.
- Analog Input 32 – Port 3, Channel 2 sensor.
- Analog Input 41 – Port 4, Channel 1 sensor.
- Analog Input 42 – Port 4, Channel 2 sensor.

These objects always exist whether a sensor is attached to the channel or not.

These objects are not completely independent. The two channels on each port share a physical connection. So, for instance, if one becomes disconnected, both are disconnected. Additionally, if the alarm for one channel is enabled or disabled, the alarm for the other channel is likewise enabled or disabled. Properties that are shared between objects will be called out.

The analog input objects employ intrinsic reporting using the OUT_OF_RANGE event algorithm.

Each analog input object supports the following properties:

- Object Name (77) – This character string property represents a unique name for each analog input object.
- Object Type (79) – this property is the BACnetObjectType enumeration, analog input (0).
- Present Value (85) – This property, of type REAL, indicates the current value, in Units (see below), of the input being measured.
- Status Flags (111) – This property, of type BACnetStatusFlags, represents four Boolean flags that indicate the general "health" of an analog input. These flags are:
 - IN_ALARM – this flag is FALSE (0) if the Event State property has a value of NORMAL, otherwise this is TRUE (1).
 - FAULT – this flag is TRUE (1) if the Reliability property is present and does not have a value of NO_FAULT_DETECTED, otherwise it is FALSE (0).
 - OVERRIDDEN – on the TV2 this flag is FALSE (0).
 - OUT_OF_SERVICE – on the TV2 this value is FALSE (0).
- Event State (36) – This property, of type BACnetEventState, shall indicate the event state of the object. For analog input objects the possible values are:
 - NORMAL (0) – if the input is operating normally.
 - OFFNORMAL (2) – if the input is disconnected.
 - HIGH LIMIT (3) – if the alarm is enabled and the high limit has been exceeded.

- LOW LIMIT (4) – if the alarm is enabled and the low limit has been exceeded.
- Out of Service (81) – This property on the TV2 is FALSE (0).
- Units (117) – This property, of type BACnetEngineeringUnits, indicates the measurement units of this object. Valid units for the TV2 are:
 - DEGREES_CELSIUS (62) – valid for temperature sensors, thermistors, and thermocouples.
 - DEGREES_FAHRENHEIT (64) – valid for temperature sensors, thermistors, and thermocouples.
 - NO_UNITS (95) – only valid if no sensor is connected.
 - INCHES_OF_WATER (58) – valid for pressure sensors.
 - HECTOPASCALS (133) – valid for pressure sensors.
 - PERCENT_RELATIVE_HUMIDITY (29) – valid for humidity sensors.
- Reliability (103) – This property, of type BACnetReliability, provides an indication of whether the Present Value or the operation of the physical input in question is "reliable" as far as the BACnet device or operator can determine and, if not, why. Valid values for this property on the TV2 are:
 - NO_FAULT (0) – sensor is enabled and operating.
 - NO_SENSOR (1) – no sensor connected to this channel.
 - COMMUNICATION_FAILURE (12) – the sensor connected to this port has lost its connection.
- Update Interval (118) – This property, of type Unsigned, indicates the maximum period of time between updates to the Present Value in hundredths of a second.
- Min Pres Value (69) – This property, of type REAL, indicates the lowest number in engineering units that can be reliably obtained for the Present Value property of this object.
- Max Pres Value (65) – This property, of type REAL, indicates the highest number in engineering units that can be reliably obtained for the Present Value property of this object.
- Resolution (106) – This read-only property, of type REAL, indicates the smallest recognizable change in Present Value in engineering units.
- Serial Number (372) – This read-only property, of type CharacterString, is assigned by the vendor to represent the serial number of the input device. This is a custom property on the TV2.
- Time Delay (113) – This property, of type Unsigned, is the pTimeDelay parameter for the object's event algorithm. It indicates how long, in seconds, a value must exceed a limit to cause the alarm to trip.
 - This property can be found in the TV2 Sensor Alarm Menu.
 - This property is writable.
- Notification Class (17) – This property, of type Unsigned, shall specify the instance of the Notification Class object to use for event-notification distribution. For the TV2 this value is '0'.
- High Limit (45) – This property is the pHighLimit parameter for the object's event algorithm.
 - This property can be found in the TV2 Sensor Alarm Menu as the "greater than" value.
 - This property is writable.
- Low Limit (59) – This property is the pLowLimit parameter for the object's event algorithm.
 - This property can be found in the TV2 Sensor Alarm Menu as the "less than" value.

- This property is writable.
- Deadband (25) – This property is the pDeadband parameter for the object's event algorithm. For the TV2 this value is '0.0'.
- Limit Enable (52) – This property, of type BACnetLimitEnable, is the pLimitEnable parameter for the object's event algorithm. For the TV2 analog input object this property consists of two flags. Each flag indicates if the associated limit is enabled or disabled. On the TV2 the limits cannot be individually controlled. So, these flags are either both TRUE or both FALSE depending on the value of the Event Algorithm Inhibit property.
- Event Enable (35) – This property, of type BACnetEventTransitionBits, shall convey three flags that separately enable and disable the distribution of TO_OFFNORMAL, TO_FAULT, and TO_NORMAL notifications. The values of these flags depend on the value of the Event [Detection Enable](#) property. If that property is TRUE then the flags are TRUE, FALSE, TRUE. If that property is FALSE then all three flags are FALSE.
- Acked Transitions (0) – This read-only property, of type BACnetEventTransitionBits, shall convey three flags that separately indicate the acknowledgment state for TO_OFFNORMAL, TO_FAULT, and TO_NORMAL events. On the TV2 these flags are all TRUE. The TV2 does not require an ACK for its event notifications.
- Notify Type (72) – This property, of type BACnetNotifyType, shall convey whether the notifications generated by the object should be Events or Alarms. On the TV2 this property is EVENT_TYPE.
- Event Time Stamps (130) – This read-only property, of type BACnetARRAY[3] of BACnetTimeStamp, shall convey the times of the last TO_OFFNORMAL, TO_FAULT, and TO_NORMAL events [recorded following the last device restart. A timestamp of 0xFFFFFFFF indicates that event has not yet occurred. These timestamps are all cleared if Event Detection Enable is set to FALSE.](#)
- Event Detection Enable (353) – This property, of type BOOLEAN, indicates whether (TRUE) or not (FALSE) intrinsic reporting is enabled in the object and controls whether (TRUE) or not (FALSE) the object will be considered by event summarization services. [On the TV2 this property controls whether the TV2 limit alarm is enabled. A FALSE value indicates that the alarm is enabled, and a TRUE value indicates that it is disabled.](#)
 - [This property can be found in the TV2 Sensor Alarm Menu as the Alarm Enabled/Disabled control.](#)
 - [This property controls the alarm for both channels of a port.](#)
 - [This property is writable.](#)
- Averaging Type (1202) – This property is an ENUMERATED value that specifies how the samples
 - The enumerations are:
 - Slow (0)
 - Medium (1)
 - Fast (2)
 - No Averaging (255)
 - This property is writable.

- Alarm Duration (1203) – This UNSIGNED property indicates how long, in seconds, an alarm will sound if there is no user intervention.
 - The default value is 900 (15 minutes).
 - The valid range is 0 <= value <= 3600
 - This property is writable.
- Clear Data (1204) – This BOOLEAN specifies if the sample data for this input should be erased.
 - A TRUE value means to erase the data.
 - This property is not readable. Attempting to read this property will generate a read-access-denied property error.
 - This property is writable.

2.33.3 Binary Input Objects

The binary input object is used to model connections within the TV2 device.

There are 5 binary input objects defined for the TV2. These are:

- Binary Input 0 – this object refers to the AC power connection of the TV2.
- Binary Input 10 – this object refers to the port 1 connection.
- Binary Input 20 – this object refers to the port 2 connection.
- Binary Input 30 – this object refers to the port 3 connection.
- Binary Input 40 – this object refers to the port 4 connection.

These objects always exist.

The binary input objects employ intrinsic reporting using the CHANGE_OF_STATE event algorithm.

Each binary input object supports the following properties:

- Object Name (77) – This character string property represents a unique name for each binary input object.
- Object Type (79) – this property is the BACnetObjectType enumeration, binary input (3).
- Present Value (85) – This enumerated property indicates the current state of the input being monitored. This value may be inactive (0) or active (1).
- Status Flags (111) – This property, of type BACnetStatusFlags, represents four Boolean flags that indicate the general "health" of an binary input. These flags are:
 - IN_ALARM – this flag is FALSE (0) if the Event State property has a value of NORMAL, otherwise this is TRUE (1).
 - FAULT – this flag is TRUE (1) if the Reliability property is present and does not have a value of NO_FAULT_DETECTED, otherwise it is FALSE (0).
 - OVERRIDDEN – on the TV2 this flag is FALSE (0).
 - OUT_OF_SERVICE – on the TV2 this value is FALSE (0).

- Event State (36) – This property, of type BACnetEventState, shall indicate the event state of the object. For binary input objects the possible values are:
 - NORMAL (0) – if the input is operating normally.
 - OFFNORMAL (2) – if the input is disconnected.
- Out of Service (81) – This property on the TV2 is FALSE (0).
- Reliability (103) – This property, of type BACnetReliability, provides an indication of whether the Present Value or the operation of the physical input in question is "reliable" as far as the BACnet device or operator can determine and, if not, why. Valid values for this property on the TV2 are:
 - NO_FAULT (0) – sensor is enabled and operating.
 - NO_SENSOR (1) – no sensor connected to this channel.
 - COMMUNICATION_FAILURE (12) – the sensor connected to this port has lost its connection.
- Polarity (84) – This property, of enumerated type BACnetPolarity, indicates the relationship between the physical state of the input and the logical state. Polarity of NORMAL (0) means that the ACTIVE state of Present Value is also the ACTIVE state of the physical input. Polarity of REVERSE (1) means that the ACTIVE state of Present Value is the INACTIVE state of the physical input.
 - The Polarity of the TV2 inputs is always NORMAL.
- Time Delay (113) – This property, of type Unsigned, is the pTimeDelay parameter for the object's event algorithm. It indicates how long, in seconds, a value must exceed a limit to cause the alarm to trip. On the TV2 this value is '0' for all binary input objects.
- Notification Class (17) – This property, of type Unsigned, shall specify the instance of the Notification Class object to use for event-notification distribution. For the TV2 this value is '0'.
- Event Enable (35) – This property, of type BACnetEventTransitionBits, shall convey three flags that separately enable and disable the distribution of TO_OFFNORMAL, TO_FAULT, and TO_NORMAL notifications. The values of these flags depend on the value of the Event [Detection Enable](#) property. If that property is TRUE then the flags are TRUE, FALSE, TRUE. If that property is FALSE then all three flags are FALSE.
- Alarm Value (6) – This property is the pAlarmValue of the object's event algorithm. For binary inputs this is the Present Value inactive (0) value.
- Acked Transitions (0) – This read-only property, of type BACnetEventTransitionBits, shall convey three flags that separately indicate the acknowledgment state for TO_OFFNORMAL, TO_FAULT, and TO_NORMAL events. On the TV2 these flags are all TRUE. The TV2 does not require an ACK for its event notifications.
- Notify Type (72) – This property, of type BACnetNotifyType, shall convey whether the notifications generated by the object should be Events or Alarms. On the TV2 this property is EVENT_TYPE.
- Event Time Stamps (130) – This read-only property, of type BACnetARRAY[3] of BACnetTimeStamp, shall convey the times of the last TO_OFFNORMAL, TO_FAULT, and TO_NORMAL events [recorded following the last device restart. A timestamp of 0xFFFFFFFF indicates that event has not yet occurred. In the case of the AC Connection, instance 0, the](#)

[timestamps are not affected by a restart. These timestamps are all cleared if Event Detection Enable is set to FALSE.](#)

- [Event Detection Enable \(353\)](#) – This property, of type BOOLEAN, indicates whether (TRUE) or not (FALSE) intrinsic reporting is enabled in the object and controls whether (TRUE) or not (FALSE) the object will be considered by event summarization services. [. On the TV2 this property controls whether the TV2 limit alarm is enabled. A FALSE value indicates that the alarm is enabled, and a TRUE value indicates that it is disabled.](#)
 - [This property can be found in the TV2 Sensor Alarm Menu as the Probe Missing Alarm control for binary inputs 10, 20, 30 and 40.](#)
 - [This property can be found in the Main Menu as the AC Disconnect Alarm for binary input 0.](#)
 - [This property is writable.](#)

2.43.4 Binary Value Object

The binary value object is used to model an on/off control in the TV2.

The single binary value object, Binary Value 0, controls the audible buzzer associated with the TV2 alarms.

The binary value object always exists.

There is no event reporting within the binary value object.

The binary value object supports the following properties:

- **Object Name (77)** – This character string property represents a unique name for each binary input object.
- **Object Type (79)** – this property is the BACnetObjectType enumeration, binary input (3).
- **Present Value (85)** – This enumerated property indicates the current state of the value being monitored. This value may be inactive (0) or active (1).
 - This value controls whether the Alarm Buzzer is audible (active) or not (inactive) when an alarm is triggered.
 - The default for the TV2 is active.
 - This value is also on the System Setup Menu as Alarm Beeper.
 - This value is writable.
- **Status Flags (111)** – This property, of type BACnetStatusFlags, represents four Boolean flags that indicate the general "health" of an binary value. These flags are all FALSE (0) for the binary value object on the TV2.
- **Event State (36)** – This property, of type BACnetEventState, shall indicate the event state of the object. For the binary value object on the TV2 there is no event reporting, so this value is NORMAL (0).
- **Out of Service (81)** – This property on the TV2 is FALSE (0).

2.53.5 Trend Log Objects

The trend log object is used to provide access to the sensor sample data stored in the TV2's internal flash memory.

There are 8 trend log objects defined for the TV2. These are:

- Trend Log 11 – Port 1, Channel 1 sensor sample data.
- Trend Log 12 – Port 1, Channel 2 sensor sample data.
- Trend Log 21 – Port 2, Channel 1 sensor sample data.
- Trend Log 22 – Port 2, Channel 2 sensor sample data.
- Trend Log 31 – Port 3, Channel 1 sensor sample data.
- Trend Log 32 – Port 3, Channel 2 sensor sample data.
- Trend Log 41 – Port 4, Channel 1 sensor sample data.
- Trend Log 42 – Port 4, Channel 2 sensor sample data.

These objects always exist whether a sensor is attached to the channel or not or whether the log contains any data.

Like analog inputs, the trend logs are not completely independent. In the TV2 sample data is stored per port. So, the data for each channel is stored as pairs each time the data is sampled and stored. Therefore, some properties affect both trend logs for a port. For instance, the Log Interval, is a shared property which means that if the Log Interval for Trend Log 11 is changed, it also is changed for Trend Log 12. Those properties that are shared will be called out.

There is no event reporting within the trend log objects.

Each trend log object supports the following properties:

- Object Name (77) – This character string property represents a unique name for each binary input object.
- Object Type (79) – this property is the BACnetObjectType enumeration, trend log (20).
- Enable (133) – This Boolean is used to turn logging on or off. On the TV2 logging cannot be turned off, so this value is always TRUE.
- Log Device Object (132) – This property, of type BACnetDeviceObjectPropertyReference, specifies the Device Identifier, Object Identifier and Property Identifier of the property whose data a log contains.
 - For each Trend Log the property referenced is the Present Value property.
 - Each Trend Log object references an Analog Input object. Each of these objects has a similar name. For example, Trend Log 11 contains data for Analog Input 11.
 - On the TV2 the data in a Trend Log is stored in the TV2 flash. The data is stored per port, not per sensor, so some properties are controlling two Trend Log objects. For example,

Trend Log 11 and Trend Log 12 sample data is stored interleaved in the same flash blocks and cannot be physically separated. Properties that are shared include:

- Log Interval
 - Log Buffer
 - Record Count
 - Total Record Count
 - Buffer Size
- Since all objects that are being logged are on this device, the Device Identifier is not necessary.
 - Log Interval (134) – This UNSIGNED property indicates the time between stored samples in 1/100 seconds.
 - This property is writable.
 - Changing this property causes the sample data for this Trend Log and its companion Trend Log, if it exists, to be cleared.
 - This property is also found on the Sensor Setup menu as the “Log Data once every” control.
 - This is the same value as the Analog Input Update Interval property. (?) Is this true?
 - Stop When Full (144) – This boolean value is intended to control how the log should handle things when it becomes full. On the TV2 when a buffer becomes full it overwrites the oldest data in the buffer, therefore this value is always FALSE.
 - Buffer Size (126) – This UNSIGNED property is the maximum number of samples that can be contained in a log. This number is dependent on the type of sensor and the number of active channels on a port.
 - Log Buffer (131) – This is a reference to the actual log data for this object.
 - This property is not readable using ReadProperty or ReadPropertyMultiple services. Attempting to read this property results in a property-not-readable error.
 - The only way to read the sample data contained in this object is by using the ReadRange service.
 - Record Count (141) – This UNSIGNED property contains the number of samples currently stored in the buffer.
 - Total Record Count (145) – This UNSIGNED property contains the total number of samples that have been stored since this buffer was created/instantiated. This value includes samples that have been overwritten.
 - Logging Type (197) – This property specifies whether the Trend Log acquires its samples using polling, COV or triggered acquisition. The TV2 uses polling so this value is the enumeration POLLED (0).
 - Status Flags (111) – This property, of type BACnetStatusFlags, represents four Boolean flags that indicate the general "health" of a trend log. These flags are all FALSE (0) for the binary value object on the TV2.

- Event State (36) – This property, of type BACnetEventState, shall indicate the event state of the object. For the trend log object on the TV2 there is no event reporting, so this value is NORMAL (0).

2.63.6 Notification Class Object

The notification class object represents and contains information required for the distribution of event notifications within BACnet systems.

There is no event reporting within the notification class object.

Each notification class object supports the following properties:

- Object Name (77) – This character string property represents a unique name for each binary input object.
- Object Type (79) – this property is the BACnetObjectType enumeration, notification class (15).
- Notification Class (17) – this property, of type UNSIGNED, shall specify the instance of the Notification Class.
- Priority (86) – this property is a BACnetArray of 3 UNSIGNED values that conveys the priority to be used for event notification for TO_OFFNORMAL, TO_FAULT and TO_NORMAL events, respectively. Each can be a value in the range 0-255, where 0 is the highest priority. For the TV2 these values are all 128.
- Ack Required (1) – this property is a BIT-STRING that contains 3 flags that represent whether acknowledgement shall be required in notifications generated for TO_OFFNORMAL, TO_FAULT and TO_NORMAL event transitions. For the TV2 these bits are all '0' or false, no ACK is required. All TV2 notifications are “unconfirmed”.
- Recipient List (102) – this property is a BACnetList of BACnetDestination object values. It specifies a list of zero or more recipient destinations to which notifications shall be sent when event-initiating objects using this class detect the occurrence of an event.
 - On the TV2 there is a single class object, Notification Class 0.
 - The recipient list specifies that event notification will be broadcast to the local network all days of the week from 00:00:00 to 23:59:59.99, Process Identifier set to 0, Issue Confirmed Notifications set to FALSE (all notifications will be unconfirmed), and all Transitions set to TRUE.
 - This list is not writable.

2.73.7 Network Port Object

The network port object describes the TV2's network interface.

There is no event reporting within the network port object.

On the TV2 none of the network port properties are writable. To modify any of the network settings, the TV2 Network Menu must be used. One of the reasons for this is when the TV2 is using WiFi, changing some of these properties could knock the TV2 off the network and there is not a secure method to login to the network. So, changing network parameters requires physical access to the TV2 or access via TCP.

Each network port object supports the following properties:

- Object Name (77) – This character string property represents a unique name for each binary input object.
- Object Type (79) – this property is the BACnetObjectType enumeration, network port (56).
- Status Flags (111) – This property, of type BACnetStatusFlags, represents four Boolean flags that indicate the general "health" of the network port. These flags are all FALSE (0) for the binary value object on the TV2.
- Out Of Service (81) – This BOOLEAN indicates if the network port is out of service (TRUE) or not (FALSE).
- Reliability (103) – This ENUMERATED value provides an indication of whether the Network Port object, the network port, and the network connected to the port are "reliable" as far as the BACnet device can determine and, if not, why.
- Network Type (427) – This ENUMERATED value indicates the type of network this object is representing. For the TV2 this value is IPV4 (5).
- Protocol Level (482) – This property, of type BACnetProtocolLevel, indicates whether the object represents a physical network interface (PHYSICAL), a non-BACnet protocol (PROTOCOL), the BACnet use of the protocol (BACNET_APPLICATION), or a non-BACnet use of the protocol (NON_BACNET_APPLICATION). For the TV2 this value is PHYSICAL (0).
- BACnet IP Mode (408) – This ENUMERATED property on the TV2 has a value of NORMAL (0) indicating that this device is operating as neither a foreign device nor a BBMD over the network port.
- Network Number (425) – This property, of type Unsigned16, represents the BACnet network number associated with this network.
- Network Number Quality (426) – This read-only ENUMERATED property represents the current quality of the Network Number property. Valid values for the TV2 are:
 - UNKNOWN (0) – the TV2 has never received a network number value.
 - LEARNED (1) – the TV2 received a learned value from the network.
 - LEARNED_CONFIGURED (2) – the TV2 received a configured value from the network.
- Changes Pending (416) – this This property, of type BOOLEAN, indicates whether the configuration settings in the Network Port object map to the current configuration settings. A value of FALSE indicates that the configuration settings reflect the current port configuration information. A value of TRUE indicates the configuration settings have been modified but have not been activated on the port. On the TV2 none of the network port properties are writable therefore this value is always FALSE.

- MAC Address (423) – this OCTET STRING contains the BACnet MAC address used on this network.
- APDU Length (399) – This property, of type Unsigned, is the maximum number of octets that may be contained in a single indivisible application protocol data unit sent or received on this port. For the TV2 this value is 1476.
- Link Speed (420) – This property, of type REAL, represents the network communication rate as the number of bits per second. A value of 0 indicates an unknown communication rate. On the TV2 this value is 0.
- Link Speed Autogenerate (422) – This property, of type BOOLEAN, represents the auto negotiation setting of the network port. On the TV2 this value is TRUE.
- IP Address (400) – This OCTET STRING indicates the IP address of the network port. This value can also be found in the TV2 Network Menu.
- BACnet IP UDP Port (412) – This property, of type Unsigned16, indicates the UDP port number of this network port.
- IP Subnet Mask (411) – This property, of type OCTET STRING, indicates the subnet mask for this network. This value can also be found in the TV2 Network Menu.
- IP Default Gateway (401) – This property, of type OCTET STRING, indicates the IP address of the default gateway for this network. This value can also be found in the TV2 Network Menu.
- BACnet IP Multicast Address (409) – This property, of type OCTET STRING, contains the BACnet/IP multicast group address to be used for the distribution of BACnet broadcast messages. On the TV2 this value is 0xFFFFFFFF (which indicates 255.255.255.255).
- IP DNS Server (406) – This property, of type BACnetARRAY[N] of OCTET STRING containing at least one entry, indicates the DNS server used by this network port for Internet host name resolution. This value can also be found in the TV2 Network Menu.
- IP DHCP Enable (402) – This BOOLEAN indicates whether this network port was configured via DHCP. This value can also be found in the TV2 Network Menu.
- IP DHCP Lease Time (403) – This read-only property, of type Unsigned, indicates the lease time in seconds of the last DHCP lease obtained for the port.
- IP DHCP Lease Time Remaining (404) – This read-only property, of type Unsigned, indicates the lease time in seconds remaining of the last DHCP lease obtained for the port.
- IP DHCP Server (406) – This read-only property, of type OCTET STRING, indicates the address of the DHCP server from which the last DHCP lease was obtained for the port. If the address of the DHCP server cannot be determined, the value of this property shall be 0x00000000.

34 BACnet Services

The Thermaviewer 2 BACnet implementation supports the following services:

- ReadProperty and ReadPropertyMultiple
- ReadRange
- WriteProperty
- ReinitializeDevice
- Who-Is/I-Am
- Who-Has/I-Have
- GetEventInformation
- UnconfirmedEventNotification/UnconfirmedCOVNotification
- TimeSynchronization

3.14.1 ReadProperty

The ReadProperty service is used by a client BACnet-user to request the value of one property of one BACnet Object. This service allows read access to any property of any object, whether a BACnet-defined object or not.

3.24.2 ReadPropertyMultiple

The ReadPropertyMultiple service is used by a client BACnet-user to request the values of one or more specified properties of one or more BACnet Objects. This service allows read access to any property of any object, whether a BACnet-defined object or not. The user may read a single property of a single object, a list of properties of a single object, or any number of properties of any number of objects. A 'Read Access Specification' with the property identifier ALL can be used to learn the implemented properties of an object along with their values.

3.34.3 ReadRange

The ReadRange service is used by a client BACnet-user to read a specific range of data items representing a subset of data available within a specified object property. The service may be used with any list or array of lists property.

This service is especially useful for Trend Log objects. It is the only way to read the Trend Log's Log Buffer data.

Using ReadRange on the Trend Log's Log Buffer property is the only way using BACnet to download sample data from the TV2.

When reading sample data from a Trend Log's Log Buffer, there is a limit to how many samples can be sent to the requester. That limit depends upon the size of the APDU that the requestor advertised in the request and the type of sensor for which the samples are being requested. If the sensor is a Temperature or Pressure sensor, then the data returned is of type REAL which is a 32-bit value. If the sensor is a Humidity sensor then the data samples should be UNSIGNED values between 0 and 100 which should be an 8-bit value.

The header and footer of the Complex-ACK that returns the sample data consist of 32 bytes and 7 bytes, respectively. For simplicity we will use 40 bytes in our calculation.

The TV2 network code utilizes a frame size of 1476 bytes. The UDP frame that contains the BACnet data consists of:

- Ethernet II header – 14 bytes
- IPV4 header – 20-60 bytes, we will use 60 bytes.
- UDP Header – 8 bytes

This results in an overhead before the BACnet packet is considered of $14 + 60 + 8 = 82$ bytes. That leaves $1476 - 82 = 1394$ bytes available for the BACnet packet.

A BACnet packet consists of the following pieces:

- BACnet Virtual Link Control (BVLC) – this is 4 bytes in length.
- BACnet NPDU – this has a variable size. For a Complex-ACK, the size we will use is 15 bytes.
- BACnet APDU – this contains all of the data defined for Result(+) in section 15.8 of the BACnet spec. This includes:
 - Object Identifier – this is the Object Identifier of the Object whose data is being read. For sample data, this is one of the Trend Log objects. (5 bytes)
 - Property Identifier – this is the ENUMERATION for the Property Id of the list being read. For the Trend Log data, this is the Log Buffer property. (2 or 3 bytes)
 - Result Flags – this is a BIT-STRING with flags that indicate if the Item Data contains the first sample, the last sample or if there are more samples to be requested to fulfill the original request. (3 bytes)
 - Item Count – this is an UNSIGNED value indicating how many samples are in the Item Data list. (2 bytes)
 - Item Data – this is a list of the samples. This list consists of:
 - Open Tag 5 (1 byte)
 - Data Samples – the size of each sample depends on the type of data. For the TV2 there are 3 types of data possible:
 - REAL – for Temperature and Pressure sensor data (19 bytes per sample)
 - UNSIGNED – For Humidity sensor data (13 bytes per sample)
 - ENUMERATED – for Special codes and Error codes (14 bytes per sample)
 - Close Tag 5 (1 byte)

- First Sequence Number (Optional) – this is required if the ReadRange is by Timestamp or Sequence Number. (5 bytes)

From the above list we have ~39 bytes of fixed data (we will use 40) and 13-19 bytes per sample of data (we will use 19).

That results in $(1394 - 40) / (19 \text{ bytes per REAL}) = 71$ REAL samples in a UDP Ethernet frame of 1476 bytes. If we assume that the data samples are from a Humidity sensor, then we could theoretically return $(1394 - 40) / (14 \text{ bytes per ENUMERATED}) = 96$ samples. (We use 14 bytes because it is the larger of the two possible value types that could be contained in the response.)

4.3.1 Determining Time Based on Stored Data

The data stored in flash consists of synchronous and asynchronous samples that is found in the Item Data read from the Log Buffer of a Trend Log. As described above, the Item Data can be one of three types of data:

- REAL – for Temperature and Pressure sensor data (19 bytes per sample)
- UNSIGNED – For Humidity sensor data (13 bytes per sample)
- ENUMERATED – for Special codes and Error codes (14 bytes per sample)

REAL and UNSIGNED data are synchronous sensor samples. Some ENUMERATED data are also synchronous. This data is a type of error that replaces a sensor sample. All synchronous data is used to determine time.

The other ENUMERATED data are asynchronous data that signify events that occur during logging. Asynchronous data does not take part in determining time. The timestamp associated with an asynchronous code is the same as the timestamp of the previous sample or synchronous code.

When a sensor is first connected a “setup menu” appears on the TV2 that allows the user to define certain parameters used for logging. One of those parameters is the time between samples, this is the Log Interval property in the Trend Log object. When this menu is exited and the settings saved, a timestamp is stored in flash that signifies the beginning of time for this sensor.

Time can then be calculated by taking the initial timestamp and counting the number of synchronous samples from the start of data collection to the desired sample. The timestamp of that sample is then the number of samples times the Log Interval plus the starting timestamp.

$$\text{Timestamp}_{\text{Sample}} = \text{Timestamp}_{\text{Start}} + (\text{Number of samples from start to target} * \text{Log Interval})$$

4.3.2 Special Codes

In this section we will describe the ENUMERATED “special” codes. These codes signify synchronous. The following are the known asynchronous “special” codes:

- [0x8000 \(32,768\) – Not used.](#)
- [0x8001 \(32,769\) – POR](#)
- [0x8002 \(32,770\) – Alarm Start Channel 1](#)
- [0x8003 \(32,771\) – Alarm Start Channel 2](#)
- [0x8004 \(32,772\) – Alarm End Channel 1](#)
- [0x8005 \(32,773\) – Alarm End Channel 2](#)
- [0x8006 \(32,774\) – Sensor Changed](#)
- [0x8007 \(32,775\) – Reviewed](#)
- [0x8008 \(32,776\) – AC Off](#)
- [0x8009 \(32,777\) – AC On](#)

These next codes are the synchronous “special” or error codes:

- [0x800A \(32,778\) – Sensor Error None](#)
- [0x800B \(32,779\) – Sensor Error No New Sample](#)
- [0x800C \(32,780\) – Not used](#)
- [0x800D \(32,781\) – Not used](#)
- [0x800E \(32,782\) – Not used](#)
- [0x800F \(32,783\) – Not used](#)
- [0x8010 \(32,784\) – Sensor Error Unknown](#)
- [0x8011 \(32,785\) – Sensor Error Sample Not Valid](#)
- [0x8012 \(32,786\) – Not used](#)
- [0x8013 \(32,787\) – Not used](#)
- [0x8014 \(32,788\) – Not used](#)
- [0x8015 \(32,789\) – Not used](#)
- [0x8016 \(32,790\) – Not used](#)
- [0x8017 \(32,791\) – Not used](#)
- [0x8018 \(32,792\) – Sample Missed \(Used when time is advanced forward\)](#)

3.44.4 WriteProperty

The WriteProperty service is used by a client BACnet-user to modify the value of a single specified property of a BACnet object. This service potentially allows write access to any property of any object, whether a BACnet-defined object or not. Some implementors may wish to restrict write access to certain properties of certain objects. In such cases, an attempt to modify a restricted property shall result in the return of an error of 'Error Class' PROPERTY and 'Error Code' WRITE_ACCESS_DENIED.

On the TV2 [settings are modified by using the menus. Those menus are only accessible if the Password feature is disabled or if a Password is provided. Using BACnet we follow that precedent by requiring the following:](#)

- [Device Password Enabled property is required to be set to FALSE to use the WriteProperty service except when writing the Device Password and Device Password Enabled properties.](#)
- [To write the Device Password and Device Password Enabled properties see the description of ReinitializeDevice service.](#)

3.54.5 ReinitializeDevice

The ReinitializeDevice service is used by a client BACnet-user to instruct a remote device to reboot itself (cold start), reset itself to some predefined initial state (warm start), to activate certain system changes, or to control the backup or restore procedure. Resetting or rebooting a device is primarily initiated by a human operator for diagnostic purposes.

Due to the sensitive nature of this service, a password may be required by the responding BACnet-user prior to executing the service.

A BACnet device may support the ReinitializeDevice service by supporting only the restart choices COLDSTART and WARMSTART. Support for the backup and restore features of this service is claimed separately.

The TV2 extends the restart choice ACTIVATE_CHANGES to support changing the system password and password enabled properties in the device object. The process for changing either of these properties is as follows:

1. [Using WriteProperty change the value of Password Enabled or Password.](#)
2. [Next initiate a ReinitializeDevice using the current \(not the modified\) password.](#)
3. [When the TV2 restarts, the targeted property will be changed.](#)

Note, that if there is any other action taken between steps 1 and 2, then the property targeted in step 1 will not be changed.

In order to modify any of the other writable properties in the TV2, the Password Enabled property in the device object must be FALSE when the WriteProperty requested is attempted.

3.64.6 Who-Is/I-Am

The Who-Is service is used by a sending BACnet-user to determine the Device object identifier, the network address, or both, of other BACnet devices that share the same internetwork. The Who-Is service is an unconfirmed service. The Who-Is service may be used to determine the Device object identifier and network addresses of all devices on the network, or to determine the network address of a specific device whose Device object identifier is known, but whose address is not.

The I-Am service is also an unconfirmed service. The I-Am service is used to respond to Who-Is service requests. However, the I-Am service request may be issued at any time. It does not need to be preceded by the receipt of a Who-Is service request. In particular, a device may wish to broadcast an I-Am service request when it powers up.

The TV2 responds to Who-Is requests with I-Am responses. The TV2 also sends out I-Am responses when it powers up and periodically to let the network know that it is still active.

3.74.7 Who-Has/I-Have

The Who-Has service is used by a sending BACnet-user to identify the Device object identifiers and network addresses of other BACnet devices whose local databases contain an object with a given Object_Name or a given Object_Identifier. The I-Have service is used to respond to Who-Has service requests or to advertise the existence of an object with a given Object_Name or Object_Identifier. The I-Have service request may be issued at any time and does not need to be preceded by the receipt of a Who-Has service request. The Who-Has and I-Have services are unconfirmed services.

The TV2 will send I-Have responses if it receives an appropriate Who-Has request.

3.84.8 GetEventInformation

The GetEventInformation service is used by a client BACnet-user to obtain a summary of all "active event states". The term "active event states" refers to all event-initiating objects that

- a) have an Event_State property whose value is not equal to NORMAL, or
- b) have an Acked_Transitions property, which has at least one of the bits (TO_OFFNORMAL, TO_FAULT, TO_NORMAL) set to FALSE.

This service is intended to be implemented in all devices that generate event notifications.

3.94.9 UnconfirmedEventNotification

The UnconfirmedEventNotification service is used by a notification-server to notify a remote device that an event has occurred. Its purpose is to notify recipients that an event has occurred, but confirmation that the notification was received is not required.

[The TV2 sends an UnconfirmedEventNotification for the following:](#)

- [Sensor Out of Range event](#)

3.104.10 UnconfirmedCOVNotification

The UnconfirmedCOVNotification Service is used to notify subscribers about changes that may have occurred to the properties of a particular object, or to distribute object properties of wide interest (such as outside air conditions) to many devices simultaneously without a subscription.

The TV2 does not support subscriptions.

[The TV2 sends an UnconfirmedCOVNotification for the following:](#)

- [Device restart](#)
- [AC Connection disconnect and reconnect](#)
- [A Sensor is disconnected or reconnected](#)
- [Change of Trend Log's Log Interval property – this is done because a side effect is the clearing of data sample memory.](#)

3.114.11 TimeSynchronization

The TimeSynchronization service is used by a requesting BACnet-user to notify a remote device of the correct current time. This service may be broadcast, multicast, or addressed to a single recipient. Its purpose is to notify recipients of the correct current time so that devices may synchronize their internal clocks with one another.

The TV2 will process TimeSynchronization requests by altering its internal clock to conform with the network time.

This service is only allowed if the system password, [Device object Device Password Enabled property](#), is disabled.

If normally the system has a password enabled, to synchronize the time the following steps must be taken:

- 1.4. Disable the password by:
 - a. Using WriteProperty [to](#) change the value of Password Enabled to FALSE.
 - b. Next initiate a ReinitializeDevice using the current (not the modified) password.
 - c. When the TV2 restarts, the password should be disabled.
- 2.5. Send a TimeSynchronization request to synchronize the TV2 with the network time.
- 3.6. Reenable the password using the same procedure as used in step 1 above but changing the Password Enabled value to TRUE.

Note that changing the TV2 time can cause the data stored in its flash memory to be erased. Therefore, it is recommended that before updating the time, the data should be downloaded. [If the Time is moved forward by more than an hour or moved backwards, the data stored in the flash memory will be erased.](#)

5 BACnet Interoperability Building Blocks (BIBBs) and Protocol Implementation Conformance Statements (PICS)

From a reading of the BACnet Specification Annex K, BACnet Interoperability Building Blocks (BIBBs), and Annex L, Descriptions And Profiles of Standardized BACnet Devices, the TV2 can be best categorized as a BACnet Smart Sensor (B-SS) in the Controller profiles grouping, Annex L.4.

As a Smart Sensor, the TV2 is required to support the following:

- Data Sharing-ReadProperty-B (DS-RP-B)
 - ReadProperty – Execute
- Device Management-Dynamic Device Binding-B (DM-DDB-B)
 - Who-Is – Execute
 - I-Am – Initiate
- Device Management-Dynamic Object Binding-B (DM-DOB-B)
 - Who-Has – Execute
 - I-Have – Initiate

Additionally, the TV2 supports the following BIBBs:

- Data Sharing-ReadPropertyMultiple-B (DS-RPM-B)
 - [ReadPropertyMultiple](#) – Execute
- [Data Sharing-WriteProperty-B \(DS-WP-B\)](#)
 - [WriteProperty](#) – Execute
- [Data Sharing-Change of Value Unsubscribed-B \(DS-COVU-B\)](#)
 - [UnconfirmedCOVNotification](#) – Initiate
- [Alarm and Event Management-Notification-Internal-B \(AE-N-I-B\)](#)
 - [UnconfirmedEventNotification](#) – Initiate
- [Alarm and Event Management-Information-B \(AE-INFO-B\)](#)
 - [GetEventInformation](#) – Execute
- [Trending-Viewing and Modifying Trends-Internal-B \(T-VMT-I-B\)](#)
 - [ReadRange](#) – Execute
- Device Management-TimeSynchronization-B (DM-TS-B)
 - TimeSynchronization – Execute
- Device Management-ReinitializeDevice-B (DM-RD-B)
 - ReinitializeDevice - Execute
- Device Management-Restart-B (DM-R-B)
 - UnconfirmedCOVNotification – Initiate

Annex L.4.5 describes the BACnet Smart Sensor as:

A B-SS is a simple sensing device with very limited resources.

Data Sharing

- Ability to provide values for any of its BACnet objects upon request

Alarm and Event Management

- No requirement

Scheduling

- No requirement

Trending

- No requirement

Device and Network Management

- Ability to respond to queries about its status
- Ability to respond to requests for information about any of its objects

5.1 Resources

For more information about BIBBs and PICS, the following links may be helpful:

- <https://store.chipkin.com/articles/bacnet-bibbs>
- <https://store.chipkin.com/articles/bacnet-what-are-bacnet-bibbs/>
- <http://www.bacnet.org/DL-Docs/>

5.2 TV2 Protocol Implementation Conformance Statement

The following section is taken from a sample statement from the BACnet site. Do not edit any of the yellow highlighted text in this section. It is meant for illustrating how the statement might be filled out for the TV2.

ANNEX A - PROTOCOL IMPLEMENTATION CONFORMANCE STATEMENT (NORMATIVE)

(This annex is part of this Standard and is required for its use.)

BACnet Protocol Implementation Conformance Statement

Date: _____

Vendor Name: Two Dimensional Instruments

Product Name: Thermaviewer 2

Product Model Number: TV-2

Application Software Version: _____ Firmware Revision: _____ BACnet Protocol Revision:

1

Product Description:

BACnet Standardized Device Profile (Annex L):

- BACnet Operator Workstation (B-OWS)
- BACnet Advanced Operator Workstation (B-AWS)
- BACnet Operator Display (B-OD)
- BACnet Building Controller (B-BC)
- BACnet Advanced Application Controller (B-AAC)
- BACnet Application Specific Controller (B-ASC)
- BACnet Smart Sensor (B-SS)
- BACnet Smart Actuator (B-SA)

List all BACnet Interoperability Building Blocks Supported (Annex K): DS-RP-B, DS-RPM-B, DM-DDB-B, DM-DOB-B, DM-TS-B, DM-RD-B, DM-R-B, DS-WP-B, DS-COVU-B, AE-N-I-B, AE-INFO-B, T-VMT-I-B

Segmentation Capability:

- Able to transmit segmented messages Window Size _____
- Able to receive segmented messages Window Size _____

Standard Object Types Supported:

- **Device**
 - Optional Properties: Serial Number, Local Time, Local Date, Last Restart Reason, Restart Notifications Recipients, Time of Device Restart
 - Writable Properties:
 - Device Password – see below.
 - Device Password Enabled – see below.
 - Temperature Units – see below.
 - Pressure Units – see below.
 - Proprietary Properties:
 - Device Password (1200), Character-String, not readable, writable in conjunction with ReinitializeDevice ACTIVATE_CHANGES. String must contain 4 numeric characters.
 - Device Password Enabled (1201), Boolean, writable in conjunction with ReinitializeDevice ACTIVATE_CHANGES. Except for Device Password and Device Password Enabled, all writable properties are writable only if Device Password Enabled is FALSE.
 - Temperature Units (1205), Enumerated, writable if Device Password Enabled is FALSE. Allowed values: DEGREES_CELSIUS (62), DEGREES_FAHRENHEIT (64).
 - Pressure Units (1206), Enumerated, writable if Device Password Enabled is FALSE. Allowed values: INCHES_OF_WATER (58), HECTOPASCALS (133).
- **Analog-Input**
 - Optional Properties: Reliability, Update Interval, Min Pres Value, Max Pres Value, Resolution, Update Interval, Event Algorithm Inhibit, Time Delay Normal
 - Writable Properties:
 - High Limit
 - Low Limit

- Time Delay – Max value: 24 hours.
- Event Algorithm Inhibit
- Averaging Type – see below.
- Alarm Duration – see below.
- Clear Data – see below.
- Proprietary Properties:
 - Serial Number (372), Unsigned. This property contains the serial number for the sensor associated with this object.
 - Averaging Type (1202), Enumerated, writable if Device Password Enabled is FALSE. This enumerated value allows the user to determine how samples are averaged to arrive at a final sample. 0-slow, 1-medium, 2-fast.
 - Alarm Duration (1203), Unsigned, writable if Device Password Enabled is FALSE. This value is how many seconds the alarm is allowed to sound before it is automatically turned off by the device. Max value=3600.
 - Clear Data (1204), Boolean, writable if Device Password Enabled is FALSE. If TRUE, then all data for the sensor associated with this object will be erased and reset to defaults by a following ReinitializeDevice ACTIVATE_CHANGES action.
- Binary-Input
 - Optional Properties: Reliability, Event Algorithm Inhibit
 - Writable Properties: Event Algorithm Inhibit
- Binary-Value
 - Optional Properties: none
 - Writable Properties: Present Value
- Trend Log
 - Optional Properties: none
 - Writable Properties:
 - Log Interval – Min value: 30 seconds; Max value: 24 hours.
 - Record Count – can only write 0 if Device Password Enabled is FALSE.
- Notification Class
 - Optional Properties: none
- Network Port
 - Optional Properties: Description, BACnet IP Multicast Address, IP DHCP Lease Time, IP DHCP Lease Time Remaining, IP DHCP Server
- None of these objects may be created.
- None of these objects may be deleted.
- There are no range restrictions.

An object type is supported if it may be present in the device. For each standard Object Type supported provide the following data:

- 1) Whether objects of this type are dynamically creatable using the CreateObject service
- 2) Whether objects of this type are dynamically deletable using the DeleteObject service
- 3) List of the optional properties supported
- 4) List of all properties that are writable where not otherwise required by this standard
- 5) List of all properties that are conditionally writable where not otherwise required by this standard
- 6) List of proprietary properties and for each its property identifier, datatype, and meaning
- 7) List of any property range restrictions

Data Link Layer Options:

- BACnet IP, (Annex J)
- BACnet IP, (Annex J), Foreign Device
- ISO 8802-3, Ethernet (Clause 7)
- ATA 878.1, 2.5 Mb. ARCNET (Clause 8)
- ATA 878.1, EIA-485 ARCNET (Clause 8), baud rate(s) _____

- MS/TP master (Clause 9), baud rate(s): _____
- MS/TP slave (Clause 9), baud rate(s): _____
- Point-To-Point, EIA 232 (Clause 10), baud rate(s): _____
- Point-To-Point, modem, (Clause 10), baud rate(s): _____
- LonTalk, (Clause 11), medium: _____
- BACnet/ZigBee (ANNEX O)
- Other: _____

Device Address Binding:

Is static device binding supported? (This is currently necessary for two-way communication with MS/TP slaves and certain other devices.) Yes No

Networking Options:

- Router, Clause 6 - List all routing configurations, e.g., ARCNET-Ethernet, Ethernet-MS/TP, etc.
- Annex H, BACnet Tunneling Router over IP
- BACnet/IP Broadcast Management Device (BBMD)
 - Does the BBMD support registrations by Foreign Devices? Yes No
 - Does the BBMD support network address translation? Yes No

Network Security Options:

- Non-secure Device - is capable of operating without BACnet Network Security
- Secure Device - is capable of using BACnet Network Security (NS-SD BIBB)
 - Multiple Application-Specific Keys:
 - Supports encryption (NS-ED BIBB)
 - Key Server (NS-KS BIBB)

Character Sets Supported:

Indicating support for multiple character sets does not imply that they can all be supported simultaneously.

- ISO 10646 (UTF-8) IBM™/Microsoft™ DBCS ISO 8859-1
- ISO 10646 (UCS-2) ISO 10646 (UCS-4) JIS X 0208

If this product is a communication gateway, describe the types of non-BACnet equipment/networks(s) that the gateway supports:

6 Revisions

The following list will describe what revisions have been made.

- October 17, 2020: original draft.