# BACnet/IP Driver Help

# Table of Contents

# BACnet/IP Driver Help

Help version 1.034

## CONTENTS



## Overview

The BACnet/IP Driver provides an easy and reliable way to connect BACnet/IP devices to OPC Client applications, including HMI, SCADA, Historian, MES, ERP, and countless custom applications. It provides connectivity to equipment using the BACnet protocol over Ethernet (which is often referred to as "BACnet" or "Annex J").

### BACnet Resources

The official BACnet specification, ANSI/ASHRAE Standard 135-2001 BACnet A Data Communication Protocol for Building Automation and Control Networks, describes all aspects of the BACnet protocol. It is recommended that users of this driver be familiar with the standard BACnet objects and properties discussed in Clause 12 of the specification. In addition, users should be familiar with the particulars of BACnet/IP outlined in Annex J of the specification. The specification document, as well as many other useful resources, is available through the American Society of Heating, Refrigerating and Air-Conditioning Engineers, Inc. (ASHRAE) or the official BACnet website **www.bacnet.org**.

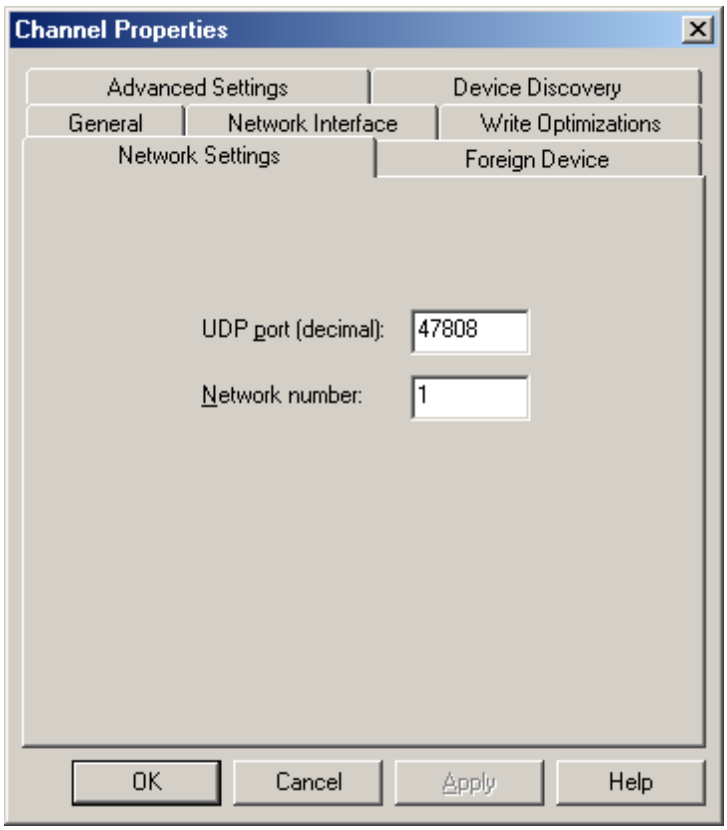
## Channel Setup

For more information on configuring a channel in the BACnet/IP Driver, select a link from the list below.



## Network Settings

The network settings are common to all devices on a channel.




Descriptions of the parameters are as follows:

- **UDP Port:** This parameter specifies the local UPD port that the driver will bind to for all communications on the channel. It is also the remote port to which all messages sent to devices on this channel will be addressed. The default setting is 47808 (0xBAC0).

  **Note:** Typically, all BACnet/IP devices on an Ethernet network use the same port.
- **Network Number:** This parameter specifies the local BACnet/IP network number on which the driver will be located. It should be set to the same network number as the local devices. The network number may range from 1 to 65534. The default setting is 1.

## Foreign Device

A foreign device is a BACnet/IP device (or software application) that resides on an IP subnet which is not part of a BACnet/IP network. BACnet/IP subnets are considered part of a larger BACnet/IP network if both directed and broadcasted messages can be forwarded to and from the other subnets by IP routers and BACnet Broadcast Management Devices (BBMD).

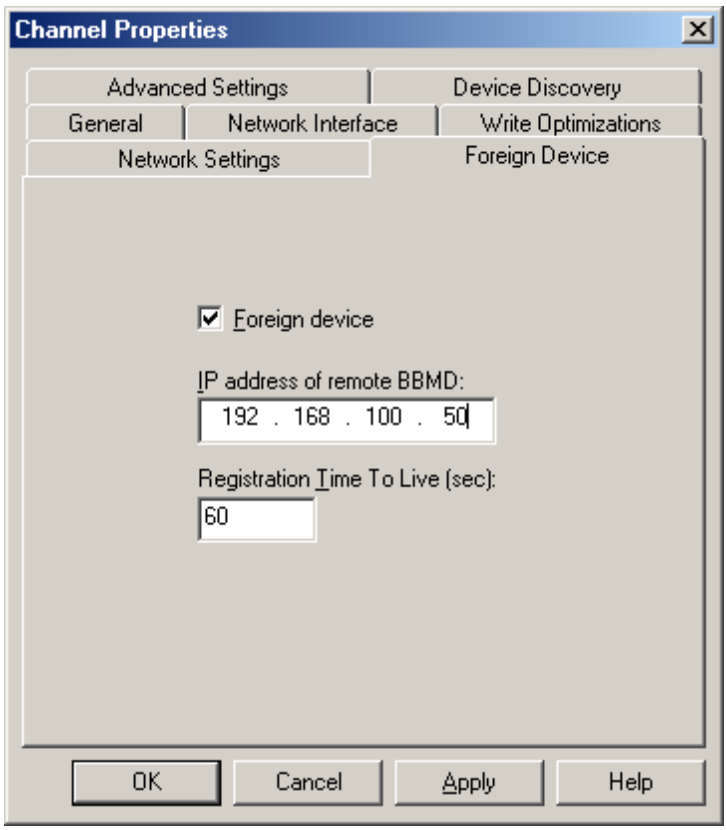
A foreign device may need to take special measures to discover devices on a BACnet network. For example, if the broadcast Who-Is/I-Am messages normally used for discovery will not be forwarded to and from the foreign device's subnet, the foreign device must work directly with a BBMD on the remote network to discover devices. The foreign device will send Who-Is messages directly to the BBMD for broadcast throughout the BACnet

network. I-Am messages broadcasted on the BACnet network will be forwarded back to the foreign device if it has registered with the BBMD.

A channel using this driver becomes a foreign device if the selected network interface is not connected to a BACnet/IP subnet. The settings on the channel's Foreign Device must be set to permit the discovery of devices on (or accessible from) a remote BACnet/IP network.

**Note:** If none of the devices configured on the channel will be using the driver's Device Discovery feature, ignore the the Foreign Device settings.

**See Also:** **Discovery** and **Configuring Multiple Channels**.

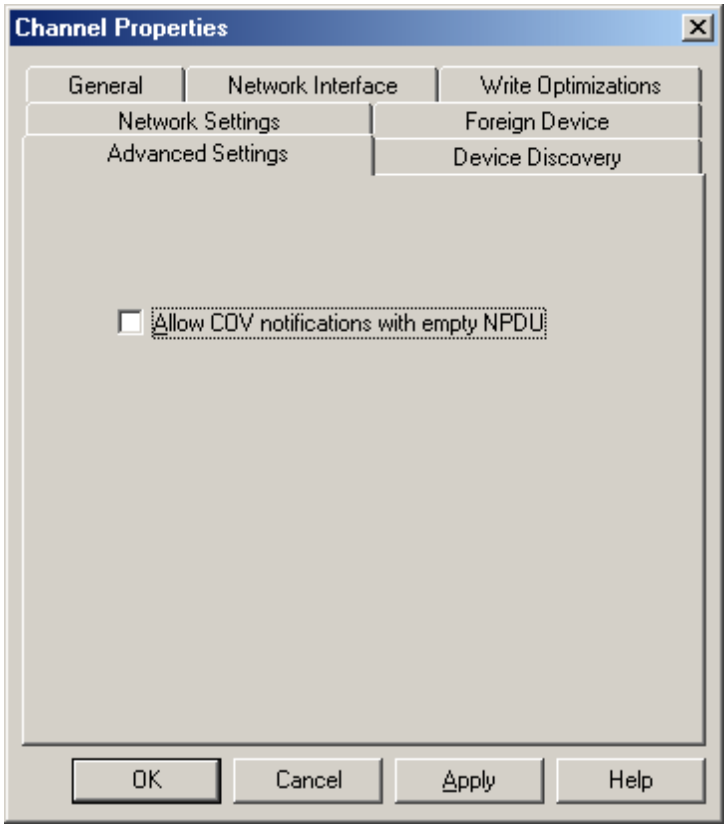



Descriptions of the parameters are as follows:

- **Foreign Device:** When checked, this parameter will enable the Device Discovery functionality. This box should be checked if the channel is not connected to a BACnet/IP subnet.
- **IP Address of Remote BBMD:** This parameter specifies the IP address of the remote BBMD with which the driver will work during Device Discovery.
- **Registration Time To Live (Sec):** This parameter specifies the length of time the BBMD should forward broadcast messages to the driver. The driver only requires an active registration during Device Discovery, and will automatically renew the registration as needed. Users should specify a time long enough for the driver to discover all configured devices (in order to prevent the need for renewals and optimize startup performance). A few seconds is generally sufficient unless import on startup is enabled or discovery timeouts are anticipated. Very long times should be avoided to reduce the load on the driver and BBMD after discovery is complete. This setting may range from 10 to 3600 seconds. The default setting is 60.

**Note:** For more information, refer to Connection Timeout under **Device Setup**.

## Advanced Settings




Description of the parameter is as follows:

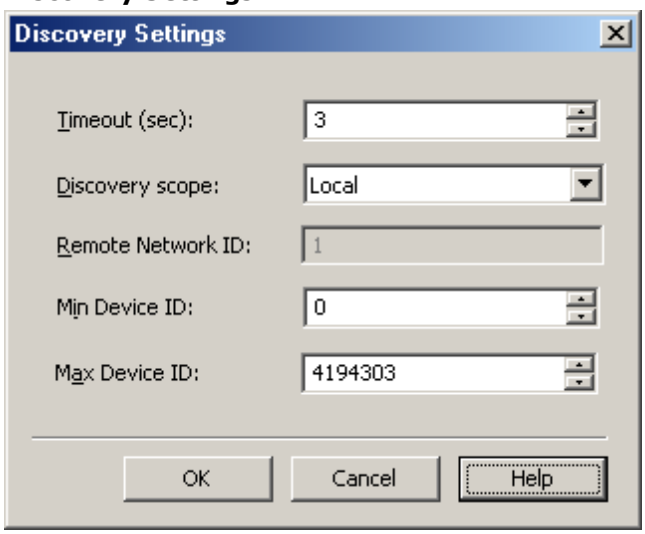
- **Allow COV notifications with empty NPDU:** When checked, this parameter allows COV notifications to be processed from a BACnet device on a different BACnet network (whose NPDUs do not contain the source address). The default setting is unchecked.

  **Note:** This setting is not commonly used, and may decrease performance on BACnet networks with many COV notifications and/or broadcast requests.

## Device Discovery

This channel-level dialog is used to specify parameters for locating devices on the network. Once devices are found, they may be added to the channel. The maximum number of devices that can be discovered at once is 65535. For more information on device discovery, refer to the server help file.

### Discovery Settings



Descriptions of the parameters are as follows:

- **Timeout:** This parameter specifies the length of time that the driver will wait for all "I-Am" responses to the initial "Who-Is" discovery request. It is also used to timeout non-responsive devices when requesting device names. The default setting is 3 seconds.
- **Discovery Scope:** This parameter controls how the driver will broadcast "Who-Is" messages. It can also be used to limit the list of devices that will be discovered. Options include Local, Global, and Remote. The default setting is Local. Descriptions of the options are as follows:

    - **Local:** When selected, "Who-Is" messages will be broadcast over the local Ethernet subnet. Devices on remote Ethernet subnets will not see these messages. BACnet gateways that are visible from local subnets can forward these messages to non-BACnet/IP subnets.
    - **Global:** When selected, "Who-Is" messages will be broadcast over the entire Ethernet network. Devices on remote Ethernet subnets will see these messages unless network routers have been configured to block broadcasts between subnets. In this case, a BBMD must be placed on each Ethernet subnet to forward broadcast BACnet messages.
    - **Remote:** When selected, "Who-Is" messages will be sent with the global broadcast IP 255.255.255.255, but will contain information so that BACnet routers and BBMDs will forward them to a single destination network. The destination BACnet network will be given in the Remote Network ID.
- **Remote Network ID:** This parameter specifies the remote Network ID that will be used for remote discovery scope. The default setting is disabled. When enabled, the default value is 1.
- **Min Device ID:** This parameter specifies the lower range for device discovery. It is used to reduce the number of discovered devices. The valid range is 0 to 4191302. The default setting is 0.

  **Note:** The specified value must be lower than the Max Device ID.
- **Max Device ID:** This parameter specifies the upper range for device discovery. It is used to reduce the number of discovered devices. The valid range is Min Device ID +1 to 4194303. The default setting is 4194303.

  **Note:** The specified value must be higher than the Min Device ID.

## Device Setup

### Supported Devices

The BACnet/IP Driver can be used successfully with devices that use the BACnet protocol, are visible on an Ethernet network, and support the objects, properties, and services supported by this driver. For more information, refer to the Protocol Implementation Conformance Statement (PICS), which is available from the hardware vendor. Conformance data for this driver is provided in **Supported Objects and Services**.

### Communications Protocol

BACnet/IP (Annex J)

**Note:** This driver requires Winsock V1.1 or higher.

### Maximum Number of Channels and Devices

The maximum number of channels supported by this driver is 128. The maximum number of devices is 100.

### Connection Timeout

For this driver, a connection is the process of verifying the presence of a BACnet/IP device on the network and successfully reading some basic communications parameters from its device object. This is accomplished by sending a "Who-Is" service request, and then processing the "I-Am" response. Since UDP is used, this does not involve the actual creation of a socket connection. The connection timeout setting is the amount of time that the driver will wait for the I-Am response. If an I-Am message is not received during this time, the driver will assume the local communications settings. Communication with the device may still be possible if the Who-Is/I-Am exchange fails. The accepted range is 1 to 30 seconds. The default setting is 3 seconds. For more information, refer to **APDU Settings**.

### Request Timeout

This parameter specifies the time that the driver will wait for an expected response from the device before retrying or going on to the next request. The valid range is 100 to 9999 milliseconds. The default setting is 1000 milliseconds.

### Retry Attempts

This parameter specifies the number of times that the driver will retry a confirmed request before giving up. The valid range is 1 to 10. The default setting is 3 retries.

### Device ID

Each device on a BACnet inter-network is uniquely identified by its network number and its device object instance. The Device ID has the form *<network number>.<device instance>*. For example, to communicate with device 100 on network 1, users would enter "1.100". The network number may range from 1 to 65534 and the device instance may range from 0 to 4194303. The IP address of the device or BACnet gateway/router device is discovered on communications startup by an Who-Is/I-Am exchange and is transparent to the user.

**Note:** Although each device on a channel must have a unique Device ID, users may address the same device from separate channels. If a device is configured with the same Device ID as another device already on that channel, a message box similar to the one shown below will be displayed. For more information, refer to **Configuring Multiple Channels**.


If an invalid Device ID is written to the "_DeviceId" device system tag, this message will not be received. Such on-the-fly configuration changes will cause communications with that device to fail.

## Supported Objects and Services

The following summarizes the portions of the BACnet protocol that are supported by this driver. It should be compared with the hardware's Protocol Implementation Conformance Statement (PICS), which is available from the hardware vendor.

### Supported Objects

Most properties of all of the standard object types are supported. For more information, select a link from the list below.

**Analog Input**
**Analog Output**
**Analog Value**
**Averaging**
**Binary Input**
**Binary Output**
**Binary Value**
**Calendar**
**Command**
**Device**
**Event Enrollment**
**File**
**Group**
**Life Safety Point**
**Life Safety Zone**
**Loop**
**Multi-state Input**
**Multi-State Output**
**Multi-State Value**
**Notification Class**
**Program**
**Schedule**
**Trend Log**

### Supported Services

| BACnet Service | BIBB* | Initiate | Execute |
|---|---|---|---|
| Who-Is | DM-DDB-A | X | |
| I-Am | DM-DDB-A | | X |
| ReadProperty | DS-RP-A | X | |
| ReadPropertyMultiple | DS-RPM-A | X | |
| WriteProperty | DS-WP-A | X | |
| WritePropertyMultiple | DS-WPM-A | X | |
| SubscribeCOV | DS-COV-A | X | |
| SubscribeCOVProperty | DS-COVP-A | X | |
| ConfirmedCOVNotification | DS-COV-A | | X |
| UnconfirmedCOVNotification | DS-COV-A | | X |

**Note:** BACnet Interoperability Building Block (BIBB). This describes the services supported by a BACnet/IP device or application. For more information, refer to Annex K of the BACnet specification.

### Data Link Layer Support

BACnet/IP (Annex J)

### Segmentation Support

The BACnet/IP Driver supports both segmented requests and segmented responses. Both requests and responses support window sizes between 1 and 127 bytes.

### Character Set Support

ISO 10646 (UTF-8)
ISO 10646 (UCS-2)
ISO 8859-1

## Cable Diagrams

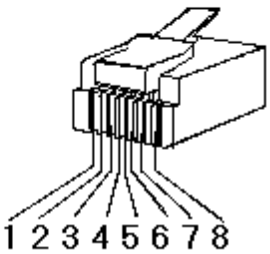
Patch Cable (Straight Through)

| | | | | |
|---|---|---|---|---|
| TD + 1 | OR/WHT | OR/WHT | 1 | TD + |
| TD - 2 | OR | OR | 2 | TD - |
| RD + 3 | GRN/WHT | GRN/WHT | 3 | RD + |
| 4 | BLU | BLU | 4 | |
| 5 | BLU/WHT | BLU/WHT | 5 | |
| RD - 6 | GRN | GRN | 6 | RD - |
| 7 | BRN/WHT | BRN/WHT | 7 | |
| 8 | BRN | BRN | 8 | |
| RJ45 | | | RJ45 | |

Crossover Cable

| | | | | |
|---|---|---|---|---|
| TD + 1 | OR/WHT | GRN/WHT | 1 | TD + |
| TD - 2 | OR | GRN | 2 | TD - |
| RD + 3 | GRN/WHT | OR/WHT | 3 | RD + |
| 4 | BLU | BLU | 4 | |
| 5 | BLU/WHT | BLU/WHT | 5 | |
| RD - 6 | GRN | OR | 6 | RD - |
| 7 | BRN/WHT | BRN/WHT | 7 | |
| 8 | BRN | BRN | 8 | |
| RJ45 | | | RJ45 | |

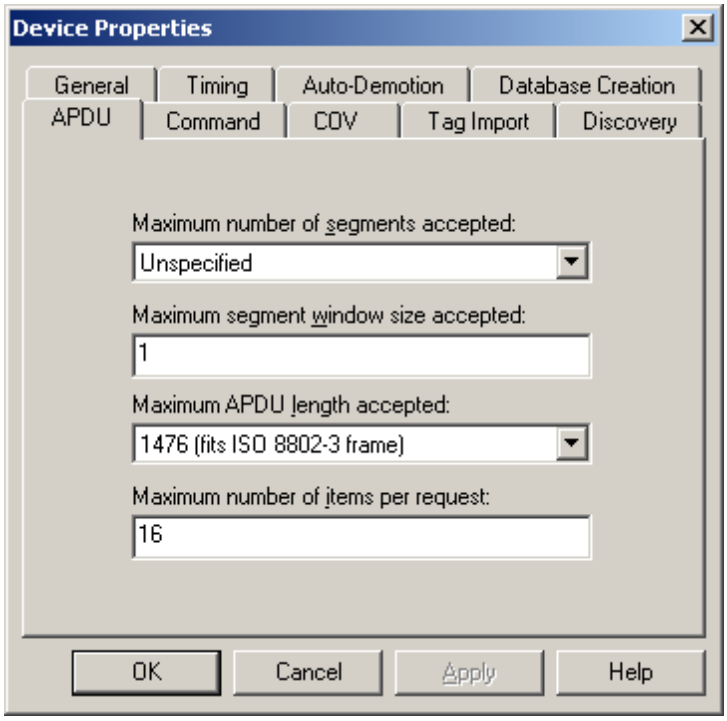
10 BaseT




8-pin RJ45

## APDU Settings

The Application Protocol Data Unit (APDU) settings affect message segmentation. These limits will be imposed by the driver, not the target device. Lower values will automatically be used if so constrained by the target device. It is generally beneficial to send messages using the largest frame and the fewest segments possible. In most cases, the default settings are acceptable.



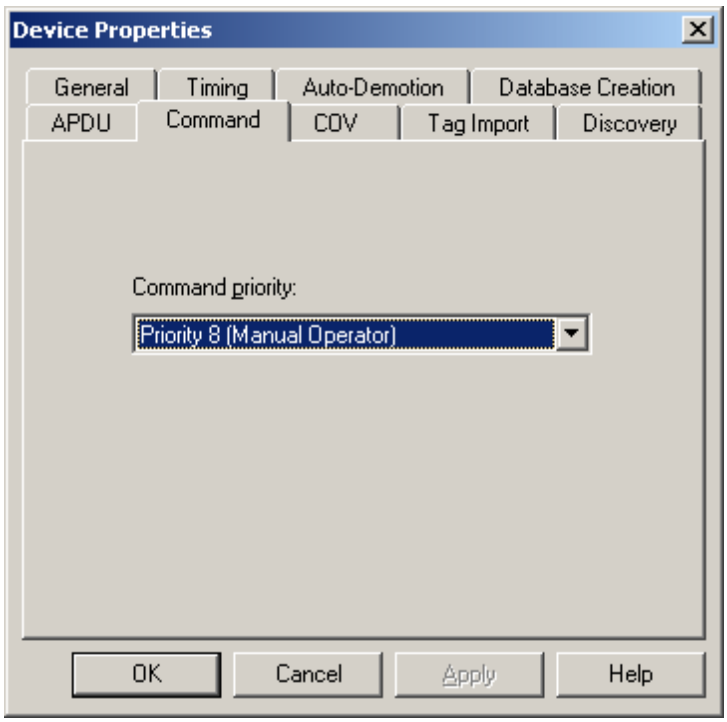

Descriptions of the parameters are as follows:

- **Maximum Number of Segments Accepted:** Although the driver is not limited in the number of response message segments it can handle, it must specify a limit when making requests. Options include 2, 4, 8, 16, 32, 64, Unlimited, and Unspecified. The default setting is Unspecified.
- **Maximum Segment Window Size Accepted:** This parameter specifies the number of message segments that can be sent before a segment acknowledge message must be returned by the receiving party. The sender proposes a window size, and the receiver determines the actual size (which will be no larger than the proposed size). Thus, the driver will use this setting as the proposed window size for requests, and as the actual window size limit for responses from the device. Larger settings will likely increase performance on a reliable network, though smaller settings will allow communications problems to be detected earlier and corrected with fewer segments being resent. The valid range is 1 to 127.
- **Maximum APDU Length Accepted:** This parameter specifies the overall length or number of bytes of message segments that the driver will accept. The largest value is generally the best choice. The driver will attempt to read the maximum APDU length allowed by the target device on startup, and will use the smallest of the local or remote limits when sending requests. A smaller setting may be needed in order to accommodate the limitations of hardware between the driver and target device. The driver will not attempt to determine the framing limits of intermediate network devices such as routers and gateways. Options include 50, 128, 206 (fits LonTalk frame), 480 (fits ARCNET frame), 1024, and 1476 (fits ISO 8803-3 frame). The default setting is 1476 bytes, which is the largest length allowed for BACnet/IP.
- **Maximum Number of Items Per Request:** This parameter limits the number of items that can be packed into read property multiple and write property multiple service requests. The actual number of items packed into a request can vary depending on how many items are due for reads or writes at a given time. Generally, the higher the value, the better the performance. For large requests or responses, however, performance gain may be diminished by message segmentation. Unfortunately, there are no general rules for determining the optimum setting. To refine a particular application, users should experiment with this setting. Devices that do not support read property multiple or write property multiple services should be set to 1. The valid range is 1 to 16. The default setting is 16.

**Note:** For supported services, refer to the device's PICS statement.

## Command Settings

BACnet/IP devices prioritize write requests to certain commandable properties according to a command priority. Once a write to a commandable property has been executed, the sending application acquires command over that property. Write requests from other applications using a lower priority will not be executed until the commanding application relinquishes its command over the property. Writes from applications using a higher priority will be executed, and command transferred to the higher priority application. For more information on creating special tags, refer to **Address Descriptions**.




### Command Priority

This setting specifies the priority level of write commands to the device. The priority level can range from 1 (the highest) to 16 (lowest). The default setting is 8. The following priority levels have accepted uses as outlined in the BACnet specification:

- **Priority 1:** Manual-Life Safety.
- **Priority 2:** Automatic-Life Safety.
- **Priority 5:** Critical Equipment Control.
- **Priority 6:** Minimum On/Off.
- **Priority 8:** Manual Operator.

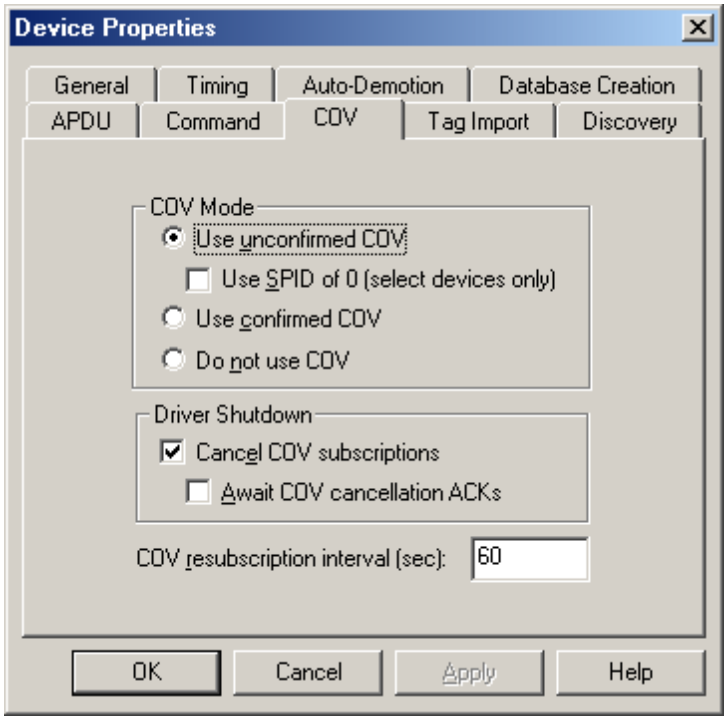
### Standard Commandable Properties and Objects

| Object | Commandable Property |
|---|---|
| Analog Output | Present Value |
| Analog Value | Present Value |
| Binary Output | Present Value |
| Binary Value | Present Value |
| Multi-state Output | Present Value |
| Multi-state Value | Present Value |

**Note:** Devices may implement additional commandable properties. For more information, refer to the hardware's PICS statement.

## COV Settings

BACnet allows applications to subscribe to change of value (COV) event notification for many properties. When COV notifications are used, the BACnet/IP Driver does not have to continuously poll the device for the current value of these properties. This reduces network traffic and the communications processing load. This driver can be configured to utilize this capability on a per device level. For more information, refer to **COV Notifications**.




### COV Mode

This setting tells the driver to subscribe to COV notifications for all properties that have implicit and explicit COV support. Descriptions of the parameters are as follows:

- **Use Unconfirmed COV:** When selected, this option specifies that the driver will receive COV notifications from the device. The device will not expect acknowledgement of those notifications. The default setting is checked.
- **Use SPID of 0 (select devices only):** When checked, this parameter sets the Subscriber Process Identifier (SPID) for all COV items to 0. When unchecked, a unique SPID will be used for each subscription. The default setting is unchecked. It is only available for **Use Unconfirmed COV** mode.

  **Note:** ALC devices consider all subscriptions with an SPID of 0 to be one subscription.

  **Important:** This option does not follow the ASHRAE Standard, and should only be used by select devices. For information on whether a specific device supports this option, refer to the device manufacturer.
- **Use Confirmed COV:** When selected, the driver will receive COV notifications and will acknowledge each one.
- **Do Not Use COV:** When selected, all of the device properties will be polled, even if the COV address modifier is present.

**Note:** For more information, refer to **Address Descriptions**.

### Driver Shutdown

Descriptions of the parameters are as follows:

- **Cancel COV Subscriptions:** When selected, the driver will send messages to the device on driver shutdown to cancel each of its COV subscriptions. Although this may slightly delay the driver's shutdown, it can be important if the device has limited resources for subscriptions (and if the subscription lifetime is long or permanent). The default setting is enabled.
- **Await COV Cancellation ACKs:** When selected, this option will cancel subscriptions one at a time. It will wait for the device to reply with an acknowledgement before cancelling the next subscription. It also prevents the Runtime from shutting down until all COV subscriptions have been cancelled. This option is only available when **Cancel COV subscriptions** is enabled. It may be helpful for a device that cannot process multiple subscription cancellations at once.

  **Note:** In large projects, this option may appear to hang the Runtime. The Runtime will recover once all subscriptions have been cancelled.

### COV Resubscription Interval (Sec)

An application can subscribe to COV notifications on a temporary or permanent basis. If 0 is specified, the driver will request permanent subscriptions. In this case, users should check **Cancel COV Subscriptions** to make sure the device can immediately reclaim resources that are no longer needed. Users may elect to subscribe to temporary subscriptions with a lifetime specified ranging from 1 to 3600 seconds. One second prior to the end of the subscription, the driver will automatically renew the subscription for active tags.

**Note:** If a COV subscription request fails for any reason, the driver will poll the device for the associated properties. A message will be placed in the server's event log indicating when this occurs.

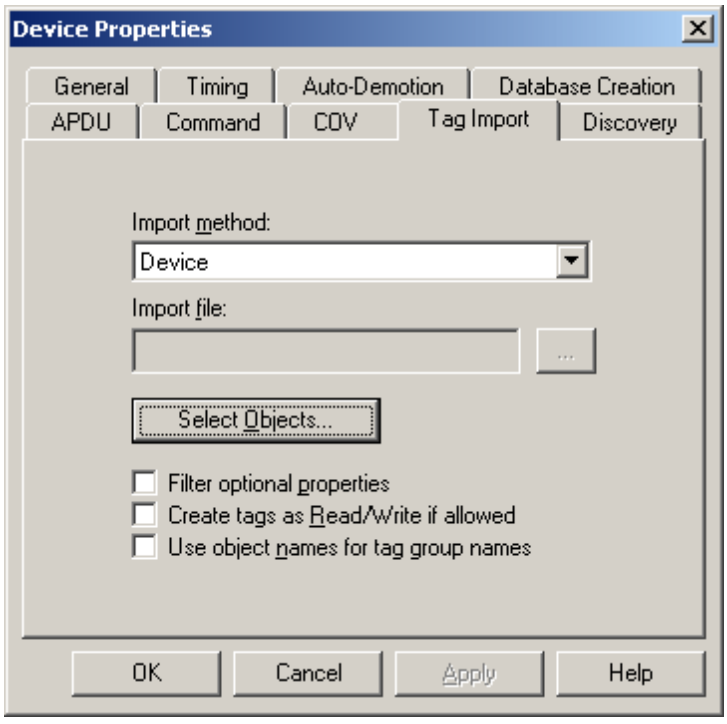
## Tag Import Settings

This driver has the ability to automatically create tags for almost all of the supported device properties. The import (tag generation) can be from a device or from a Cimetrics OPC Server export file. The Tag Import tab can be used to specify the object types that tags will be generated in addition to other tag generation options. For information on creating a tag database from the device, refer to the instructions below.

**Note:** The Database Creation settings control when the automatic tag database generation occurs. For more information, refer to the server's help documentation.

1. Leave the default settings on the **Database Creation** tab.
2. Click **Tag Import**.
3. Select the desired **Tag Import Options**, and then click **Apply**.
4. Return to the **Database Creation** tab and click **Auto Create** in order to generate tags for the selected device immediately. For this to succeed, the device must be online and network visible to the driver.

**Note:** Tags must be generated for each device individually using this method. If there is large number of devices or if the configuration of the devices changes, users should specify one of the **Generate on startup** options on the Database Creation tab.




Descriptions of the parameters are as follows:

- **Import Method:** Tags can be imported in one of the following ways.

    1. Select **Device** to import tags from the device.The device must be online (and network visible to the driver) at the time of import.

2. Select **Cimetrics OPC Server csv export file** to import tags exported to a csv file from the Cimetrics OPC Server. Although export files may contain data for multiple devices, the driver will only import data for this device.

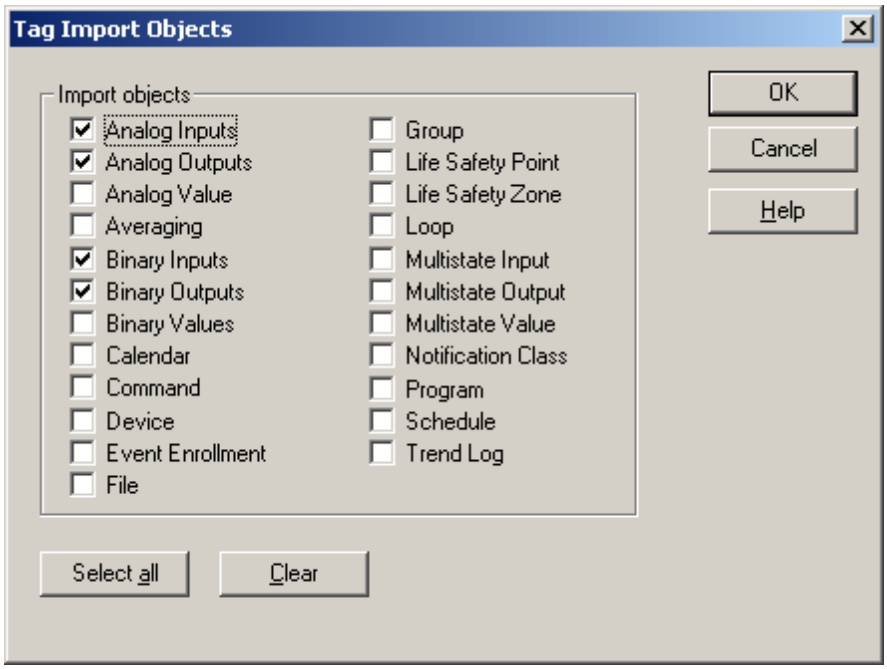
- **Import file:** This parameter is used to specify the import file's path and file name. To browse for the import file, click on the Browse button (**...**).

  **Note:** This parameter will be disabled if Device is the chosen the import method.
- **Select Objects:** When clicked, this button will invoke the **Tag Import Objects** dialog.
- **Filter Optional Properties:** When left at the default unchecked setting, the driver will generate tags for all the supported properties imported. This could result in a very large number of tags, many of which may not be needed. To reduce the number of tags that are generated, check **Filter Optional Properties** in order to generate tags for properties that are required by the BACnet specification only. Required properties are those with conformance code **R** (readable) or **W** (writable). Tags for non-standard properties and properties with conformance code **O** (optional) will not be generated.
- **Create Tags as Read/Write If Allowed:** Tags will be generated with Read Only access for properties with a conformance code of **R**, or with Read/Write access if the conformance code is **W**. The default access for properties with a conformance code of **O** depends on the nature of the data. Some BACnet/IP devices allow writes to properties that are described as Read Only in the BACnet specification. The BACnet specification does not specifically forbid this for most properties. All tags may be generated with full Read/Write access to accommodate these non-standard implementations. For more information, refer to **Address Descriptions**.

  **Note:** This parameter will be disabled if **Cimetrics OPC Server csv Export File** has been selected as the import method, because the access level will be given in the file.
- **Use Object Names for Tag Group Names:** New tag groups will be given the name of the corresponding BACnet object. If the object name is not defined or is not unique, the driver will assign a default name to the group.

  **Note:** This parameter will be disabled if **Cimetrics OPC Server csv Export File** has been selected as the import method, because the name was not given in the file.

### Tag Import Objects



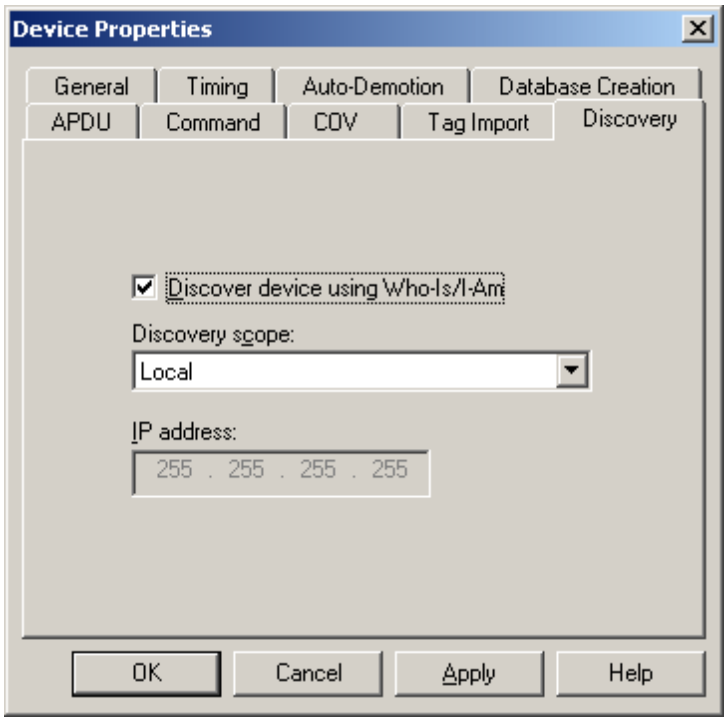

Descriptions of the parameters are as follows:

- **Import Objects:** The driver supports most properties of all of the standard BACnet object types. Select the object types for the driver to generate tags. The basic I/O object types are selected by default.

- **Select All:** When clicked, this button will automatically check all of the boxes. It does no harm to check an object type that does not exist in the device at the time of tag import.
- **Clear:** When clicked, this button will uncheck all of the selected boxes.

## Discovery

The Device ID, which is set on the General tab of the Device Properties dialog, is sufficient to uniquely identify a device on a BACnet network. However, the driver requires additional information to establish communication with a device. This information includes the IP address of the device or router that messages should be sent to, framing constraints, and the BACnet MAC (medium access control) address of the device. In most cases, the driver will obtain this information from an I-Am message sent from the device. The driver prompts each device to send an I-Am by broadcasting a Who-Is message targeted at the particular device instance. The Discovery tab's parameters can be used to control how the driver will obtain the necessary communication parameters.




### Discover Device Using Who-Is/I-Am

When checked, this setting allows the driver to automatically discover the required device properties. This option should be unchecked if the device does not support the I-Am and Who-Is services or if broadcasting messages on the network are undesired. The default setting is checked. Be sure to enter the IP of the device below.

**Note 1:** If Discovery through Who-Is/I-Am is not used, the destination device must reside on a BACnet/IP network.

**Note 2:** If the Network Interface selected for the channel is not connected to a BACnet/IP network, users must configure the channel to operate as a foreign device in order to discover devices. For more information, refer to **Foreign Device**.

**See Also:** **Configuring Multiple Channels**

### Discovery Scope

This option controls how the driver will broadcast Who-Is messages. These options can be used to limit the amount of broadcast traffic on the network. Three options are available:

1. **Local:** Who-Is messages are broadcast over the local Ethernet subnet. Devices on remote Ethernet subnets will not see these messages. BACnet gateways that are visible from a local subnet can forward these messages to non-BACnet/IP subnets.

2. **Global:** Who-Is messages are broadcast over the entire Ethernet network. Devices on remote Ethernet subnets will see these messages unless network routers have been configured to block broadcasts between subnets. This is common. In these cases, a BBMD must be placed on each Ethernet subnet to forward broadcast BACnet messages.

3. **Remote:** Who-Is messages are sent with the global broadcast IP (255.255.255.255), but contain information such that BACnet routers and BBMDs will forward them to a single destination network. The destination BACnet network will be that given in the Device ID.

### IP Address

The IP address defines to whom the driver will send messages and expect responses from. This will be the IP address of the device if that device is on the local Ethernet network. If the destination device is on a remote network, then this setting must be the IP address of the local router through which communications will be conducted.

## Optimizing Your BACnet/IP Communications

### Use Multiple Channels

This driver has been designed to service read and write requests to multiple devices simultaneously. However, all pending requests on a channel must be completed before the next set of requests for that channel's devices can be issued. If one device is slow or not responding, it will degrade the performance of all devices on that channel. Each channel, however, operates independently. Therefore, it is recommended that users separate devices into several channels for optimum performance. For more information on the special system requirements for a multi-channel configuration, refer to **Configuring Multiple Channels**.

### Maximize Frame Size

Messages will be sent in multiple segments if necessary. To reduce the overhead incurred by message segmentation, use the largest frame size possible. Before the driver begins reading and writing data to a device, it will read the APDU length limit of that device. From that point on, the driver will use its maximum APDU length or the devices maximum APDU length, whichever is smallest. Thus, selecting the largest APDU length option on the APDU Settings Device Properties will result in the optimum frame size. However, the driver does not try to see if any network hardware between it and the device (such as BACnet routers and gateways) has more restrictive limits. Thus, users may need to reduce the driver's limit to accommodate.

### Maximize Window Size

The window size is the number of message segments that can be sent before the receiving party must return a segment acknowledgment. The sender must wait for this acknowledgment before sending the next series of message segments. Maximizing the window size will reduce the amount of time consumed waiting for acknowledgements. However, this must be done with the knowledge that communication errors will not be detected as quickly, and that more data will have to be resent to correct the problem. For more information on configuring the driver's window size, refer to **APDU Settings**.

### Utilize ReadPropertyMultiple and WritePropertyMultiple Services

Packing multiple Read/Write operations into a single request can greatly improve performance by reducing the number of transactions required for a given number of tag reads or writes. Check the hardware's PICS document to see if these services are supported. As more items are added to a request, the larger the request and/or response messages will become. Large messages may need to be segmented. While it is unlikely that the increased overhead required to send segmented messages would completely negate the performance advantage of using multiple property requests, it is a consideration. Also, when using multiple property requests, the frame and window size issues described above become more relevant. For more information on how to enable multiple property requests, refer to **APDU Settings**.

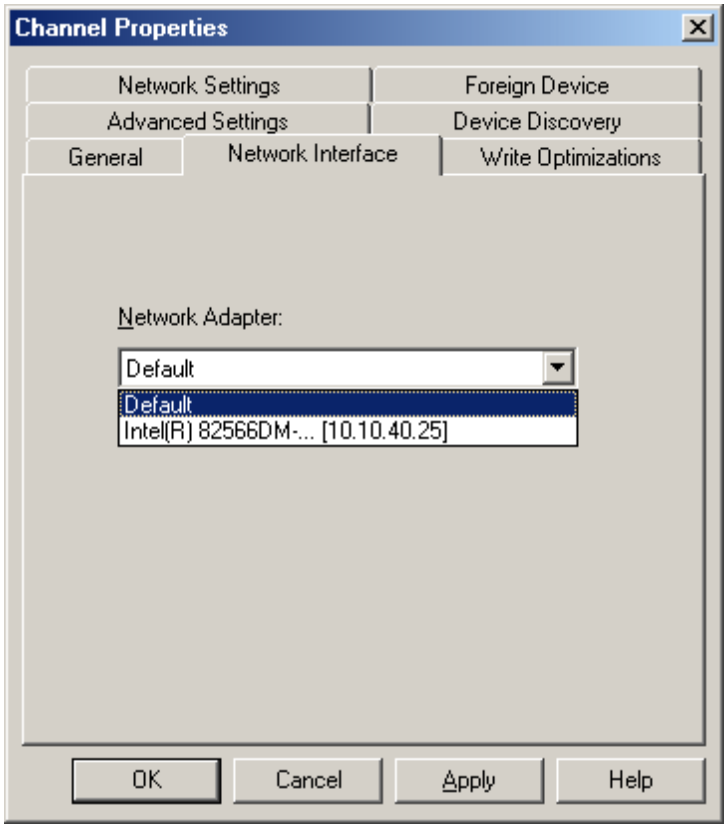
### Use COV Reporting

The amount of network traffic and request processing load can be reduced by using Change Of Value (COV) reporting wherever possible. For more information, refer to **COV Notifications**.

## Configuring Multiple Channels

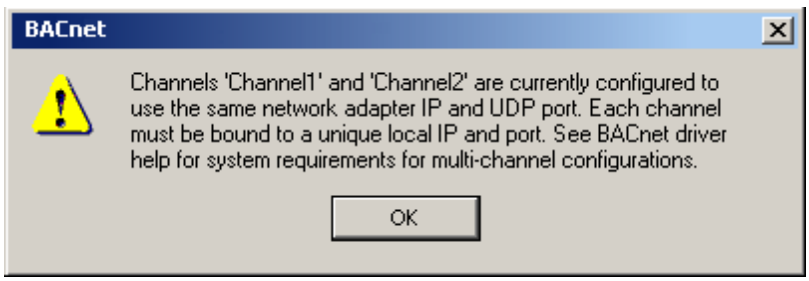
Multiple channels can increase the driver's performance; however, unlike most Ethernet drivers for the OPC server, the BACnet/IP driver requires that each channel bind to a unique local address. The address is the combination of IP and port. The local port is set with the **Network Settings** Channel Properties. The local IP used is associated with a Network Adapter Card (NIC) installed on the system, and can be selected with the Network Interface Channel Properties shown below.

**See Also:** **Optimizing Your BACnet Communications**




Selecting **Default** will allow the driver to use the operating system's normal bind order to set the NIC that will be used. This is the recommended selection for a single channel BACnet/IP driver project, since such a project may be run on another computer without modification. For clarity of the actual local IP usage, it is not recommended that Default be selected in any of the channels in a multi-channel project.

**Note:** If a channel is configured with an IP and port combination that is already in use by another BACnet/IP driver channel, a message box similar to the one shown below will be displayed.




This box will not be displayed if an invalid adapter selection is written to the "_NetworkAdapter" channel system tag. However, such configuration changes will cause communications with all devices on that channel to fail.

### Using Multiple Local IP Addresses

To create a BACnet/IP driver project with multiple channels while using the same UDP port on each, the project will need to be run on a multihomed computer, which is a computer that has multiple IP addresses associated with it. Each BACnet/IP driver channel will then bind to a different local IP. A computer can be multihomed by installing multiple NICs, or by associating multiple IP addresses with a single NIC. The process of adding IP addresses to a single NIC system is easy, but differs slightly depending on the version of Windows being used.

### Adding IP Addresses to a Single NIC on Windows NT

1. Click the **My Computer** icon and then select **Control Panel**.
2. Click the **Network** icon and then click on the **Protocols** tab.
3. Select **TCP/IP Protocol**.

4. Click **Properties** and then select the **IP Address** tab.
5. Click **Advanced** | **Add**.
6. Enter the additional IP address and subnet mask.
7. Click **OK**.
8. Restart the computer.

### Adding IP Addresses to a Single NIC on Windows 2000, XP, and 2003

1. Click the **My Computer** icon and then select **Control Panel**.
2. Click the **Network andDial-Up Connections** icon.
3. Click the **Local Area Connection** icon (or other icon associated with NIC of interest).
4. Click the **Properties** button and then select **Internet Protocol (TCP/IP)**.
5. Click **Properties** | **Advanced**.
6. Select the **IP Settings** tab and then click **Add**.
7. Enter the additional IP address and subnet mask.
8. Click **OK**.

### Adding IP Addresses to Windows Vista, 2008, and 7

1. Click **Start** and then open **Network Connections**.
2. Next, click **Control Panel** | **Network and Internet**. Then, select the connection that will be changed.
3. Click **Properties**, and then provide the administrator password or confirmation (if prompted).
4. Click the **Networking** tab. Beneath **This connection uses the following items**, click **Internet Protocol Version 4 (TCP/IPv4)**.
5. Next, click **Properties**. Ensure that the connection is set to use an IP address by clicking **Use the following IP address**.
6. Next, specify the IP address settings in the **IP Address**, **Subnet Mask**, and **Default Gateway** fields.
7. To add a second IP address click **Advanced** | **IP Settings**. Beneath **IP Address**, click **Add**. Then enter a new IP address and subnet mask.

### Multihoming Notes

1. Multihoming is not supported under Windows 95 or 98.

2. Users can only multi-home a network card that is configured to use static IP addresses.

3. Windows NT can be used to add up to five IP addresses for each NIC via the control panel. If more IP addresses are required, add them to the registry manually. To browse, look under **HEKY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services**. Select the service associated with the adapter card in question. Under the service, go to the **Parameters\Tcpip** subkey. Add the IP addresses to **IPAddress**. Edit **SubnetMask** and add an entry for each new IP address.

4. Windows 2000 and XP do not impose a limit on the number of IP addresses that may be added via the control panel.

5. There will be additional operating system overhead when running on a multihomed system. However, this will typically be negligible when compared to the performance gained from the use of multiple channels.

## COV Notifications

BACnet provides for unsolicited Change Of Value (COV) reporting of critical properties. The advantage of COV is that the driver does not have to continuously poll the device for these values. Depending on the application, this can result in a significant reduction in network traffic as well as reducing the request processing load on the device and driver.

### Implicit vs. Explicit COV

The BACnet specification requires that devices support COV reporting for certain properties. The device may also support COV reporting for other properties, depending on the implementation. Tag addresses to those properties that are required to support COV, are said to have implicit COV support. To take advantage of COV capability of other properties, if implemented, users must add the COV modifier to the tag's address. These tags are said to have explicit COV support. For more information on address syntax, refer to **Address Descriptions**.

### COV Subscription

Before the driver can receive COV notifications for a particular property, it must first issue a COV subscription request. If the subscription request succeeds, the driver will receive the initial value of the property and COV notifications whenever necessary for as long as the subscription is in effect. The driver will re-subscribe as needed. If a subscription attempt fails, the driver will issue a message to that effect in the server's event log. Users may choose to configure the driver to begin polling that property if the subscription request fails. For configuration details, refer to **COV Settings**.

### COV Mode

COV notifications can be confirmed or unconfirmed. Confirmed notifications require the driver to acknowledge the notification. Unconfirmed notifications are not acknowledged. The driver supports both modes of operations, along with a COV disabled mode where all tags are polled.

### COV Watchdog Tags

The OPC quality of tags updated by COV notifications may be GOOD, even if the device has momentarily gone offline. Polled tags, by comparison, can quickly reveal a communications problem when an expected read response fails to arrive. When a poll fails, the driver will flag the device as being in an error state. The driver uses the device error state to set the quality of COV data. Therefore, users interested in monitoring just COV properties should consider polling for one additional property in the device. That polled tag will act as a watchdog for all COV data from that device.

**Note 1:** The BACnet SubscribeCOV service will be used for properties with implicit COV support. The BACnet SubscribeCOVProperty service will be used for all other properties addressed with the COV address modifier.

**Note 2:** If a COV subscription request fails for any reason, the driver will poll the device for the associated properties. A message will be placed in the server's event log indicating when this occurs.

## Data Types Description

| Data Type | Description |
|---|---|
| Boolean | Single bit |
| Word | Unsigned 16 bit value<br><br>bit 0 is the least significant bit<br>bit 15 the most significant bit |
| Short | Signed 16 bit value<br><br>bit 0 is the least significant bit<br>bit 14 the most significant bit<br>bit 15 is the sign bit |
| DWord | Unsigned 32 bit value<br><br>bit 0 is the least significant bit<br>bit 31 the most significant bit |
| Long | Signed 32 bit value<br><br>bit 0 is the least significant bit<br>bit 30 the most significant bit<br>bit 31 is the sign bit |
| Float | 32 bit Floating point value<br><br>bit 0 is the low bit<br>bit 31 is the high bit |
| String | Null terminated ASCII string |

## Enumerated Data Types

This driver expresses property values with enumerated BACnet data types as integers. The standard enumeration definitions given in the BACnet protocol specification are displayed below.

### Enumerated BACnet Data Types

**BACnetAction**
**BACnetBackupState**
**BACnetBinaryPV**
**BACnetDeviceStatus**
**BACnetEngineeringUnits**
**BACnetEventState**
**BACnetEventType**
**BACnetFileAccessMethod**
**BACnetLifeSafetyMode**
**BACnetLifeSafetyOperation**
**BACnetLifeSafetyState**
**BACnetMaintenance**
**BACnetNotifyType**
**BACnetObjectType**
**BACnetPolarity**
**BACnetProgramError**
**BACnetProgramRequest**
**BACnetProgramState**
**BACnetReliability**
**BACnetSegmentation**
**BACnetSilencedState**
**BACnetVTClass**

### BACnetAction

| Value | Action |
|---|---|
| 0 | Direct |
| 1 | Reverse |

**BACnetBackupState**

| Value | Event Type |
|---|---|
| 0 | Idle |
| 1 | Preparing for backup |
| 2 | Preparing for restore |
| 3 | Performing a backup |
| 4 | Performing a restore |
| 5 | Backup failure |
| 6 | Restore failure |

**BACnetBinaryPV**

| Value | Binary Present Value |
|---|---|
| 0 | Inactive |
| 1 | Active |

**BACnetDeviceStatus**

| Value | Device Status |
|---|---|
| 0 | Operational |
| 1 | Operational-Read Only |
| 2 | Download-required |
| 3 | Download-in-progress |
| 4 | Non-operational |
| 5 | Backup-in-progress |

**BACnetEngineeringUnits**

**Acceleration**

| Value | Unit |
|---|---|
| 166 | Meters-per-second-per-second |

**Area**

| Value | Unit |
|---|---|
| 0 | Square-meters |
| 1 | Square-feet |
| 115 | Square-inches |
| 116 | Square-centimeters |

**Currency**

| Value | Unit |
|---|---|
| 105 | Currency1 |
| 106 | Currency2 |
| 107 | Currency3 |
| 108 | Currency4 |
| 109 | Currency5 |
| 110 | Currency6 |
| 111 | Currency7 |
| 112 | Currency8 |
| 113 | Currency9 |
| 114 | Currency10 |

**Electrical**

| Value | Unit |
|---|---|
| 2 | Milliamperes |
| 3 | Amperes |
| 4 | Ohms |
| 5 | Volts |

| | |
|---|---|
| 6 | Kilovolts |
| 7 | Megavolts |
| 8 | Volt-amperes |
| 9 | Kilovolt-amperes |
| 10 | Megavolt-amperes |
| 11 | Volt-amperes-reactive |
| 12 | Kilovolt-amperes-reactive |
| 13 | Megavolt-amperes-reactive |
| 14 | Degrees-phase |
| 15 | Power-factor |
| 122 | Kilohms |
| 123 | Megohms |
| 124 | Millivolts |
| 145 | Milliohms |
| 167 | Amperes-per-meter |
| 168 | Amperes-per-square-meter |
| 169 | Ampere-square-meters |
| 170 | Farads |
| 171 | Henrys |
| 172 | Ohm-meters |
| 173 | Siemens |
| 174 | Siemens-per-meter |
| 175 | Teslas |
| 176 | Volts-per-degree-Kelvin |
| 177 | Volts-per-meter |
| 178 | Webers |

**Energy**

| Value | Unit |
|---|---|
| 16 | Joules |
| 17 | Kilojoules |
| 18 | Watt-hours |
| 19 | Kilowatt-hours |
| 20 | Btus |
| 21 | Therms |
| 22 | Ton-hours |
| 125 | Kilojoules-per-kilogram |
| 126 | Megajoules |
| 146 | Megawatt-hours |
| 147 | Kilo-btus |
| 148 | Mega-btus |

**Enthalpy**

| Value | Unit |
|---|---|
| 23 | Joules-per-kilogram-dry-air |
| 24 | Btus-per-pound-dry-air |
| 117 | Btus-per-pound |
| 149 | Kilojoules-per-kilogram-dry-air |
| 150 | Megajoules-per-kilogram-dry-air |

**Entrophy**

| Value | Unit |
|---|---|
| 127 | Joules-per-degree-Kelvin |
| 128 | Joules-per-kilogram-degree-Kelvin |
| 151 | Kilojoules-per-degree-Kelvin |
| 152 | Megajoules-per-degree-Kelvin |

**Force**

| Value | Unit |
|---|---|
| 153 | Newton |

**Frequency**

| Value | Unit |
|---|---|
| 25 | Cycles-per-hour |
| 26 | Cycles-per-minute |
| 27 | Hertz |
| 129 | Kilohertz |
| 130 | Megahertz |
| 131 | Per-hour |

**Humidity**

| Value | Unit |
|---|---|
| 28 | Grams-of-water-per-kilogram-dry-air |
| 29 | Gercent-relative-humidity |

**Length**

| Value | Unit |
|---|---|
| 30 | Millimeters |
| 31 | Meters |
| 32 | Inches |
| 33 | Feet |
| 118 | Centimeters |

**Light**

| Value | Unit |
|---|---|
| 34 | Watts-per-square-foot |
| 35 | Watts-per-square-meter |
| 36 | Lumens |
| 37 | Luxes |
| 38 | Foot-candles |
| 179 | Candelas |
| 180 | Candelas-per-square-meter |

**Mass**

| Value | Unit |
|---|---|
| 39 | Kilograms |
| 40 | Pounds-mass |
| 41 | Tons |

**Mass Flow**

| Value | Unit |
|---|---|
| 42 | Kilograms-per-second |
| 43 | Kilograms-per-mimute |
| 44 | Kilograms-per-hour |
| 45 | Pounds-mass-per-minute |
| 46 | Pounds-mass-per-hour |
| 119 | Pounds-mass-per-second |
| 154 | Grams-per-second |
| 155 | Grams-per-minute |
| 156 | Tons-per-hour |

**Power**

| Value | Unit |
|---|---|

| | |
|---|---|
| 47 | Watts |
| 48 | Kilowatts |
| 49 | Megawatts |
| 50 | Btus-per-hour |
| 51 | Horsepower |
| 52 | Tons-refrigeration |
| 132 | Milliwatts |
| 157 | Kilo-btus-per-hour |

**Pressure**

| Value | Unit |
|---|---|
| 53 | Pascals |
| 54 | Kilopascals |
| 55 | Bars |
| 56 | Pounds-force-per-square-inch |
| 57 | Centimeters-of-water |
| 58 | Inches-of-water |
| 59 | Millimeters-of-mercury |
| 60 | Centimeters-of-mercury |
| 61 | Inches-of-mercury |
| 133 | Hectopascals |
| 134 | Millibars |

**Temperature**

| Value | Unit |
|---|---|
| 62 | Degrees-Celsius |
| 63 | Degrees-Kelvin |
| 64 | Degrees-Fahrenheit |
| 65 | Degree-days-Celsius |
| 66 | Degree-days-Fahrenheit |
| 120 | Delta-Degrees-Fahrenheit |
| 121 | Delta-Degrees-Kelvin |
| 181 | Kelvins-per-hour |
| 182 | Kelvins-per-minute |

**Time**

| Value | Unit |
|---|---|
| 67 | Years |
| 68 | Months |
| 69 | Weeks |
| 70 | Days |
| 71 | Hours |
| 72 | Minutes |
| 73 | Seconds |
| 158 | Hundredths-seconds |
| 159 | Milliseconds |

**Torque**

| Value | Unit |
|---|---|
| 160 | Newton-meters |

**Velocity**

| Value | Unit |
|---|---|
| 74 | Meters-per-second |
| 75 | Kilometers-per-hour |
| 76 | Feet-per-second |

| | |
|---|---|
| 77 | Feet-per-minute |
| 78 | Miles-per-hour |
| 161 | Millimeters-per-second |
| 162 | Millimeters-per-minute |
| 163 | Meters-per-minute |
| 164 | Meters-per-hour |

**Volume**

| Value | Unit |
|---|---|
| 79 | Cubic-feet |
| 80 | Cubic-meters |
| 81 | Imperial-gallons |
| 82 | Liters |
| 83 | Us-gallons |

**Volumetric Flow**

| Value | Unit |
|---|---|
| 84 | Cubic-feet-per-minute |
| 85 | Cubic-meters-per-second |
| 86 | Imperial-gallons-per-minute |
| 87 | Liters-per-second |
| 88 | Liters-per-minute |
| 89 | Us-gallons-per-minute |
| 135 | Cubic-meters-per-hour |
| 136 | Liters-per-hour |
| 142 | Cubic-feet-per-second |
| 165 | Cubic-meters-per-minute |

**Other**

| Value | Unit |
|---|---|
| 90 | Degrees-angular |
| 91 | Degrees-Celsius-per-hour |
| 92 | Degrees-Celsius-per-minute |
| 93 | Degrees-Fahrenheit-per-hour |
| 94 | Degrees-Fahrenheit-per-minute |
| 95 | No-units |
| 96 | Parts-per-million |
| 97 | Parts-per-billion |
| 98 | Percent |
| 99 | Percent-per-second |
| 100 | Per-minute |
| 101 | Per-second |
| 102 | Psi-per-Degree-Fahrenheit |
| 103 | Radians |
| 104 | Revolutions-per-minute |
| 137 | Kilowatt-hours-per-square-meter |
| 138 | Kilowatt-hours-per-square-foot |
| 139 | Megajoules-per-square-meter |
| 140 | Megajoules-per-square-foot |
| 141 | Watts-per-square-meter-Degree-Kelvin |
| 143 | Percent-obscuration-per-foot |
| 144 | Percent-obscuration-per-meter |
| 183 | Joule-seconds |
| 185 | Square-meters-per-Newton |
| 186 | Kilogram-per-cubic-meter |
| 187 | Newton-seconds |

| | |
|---|---|
| 188 | Newtons-per-meter |
| 189 | Watts-per-meter-per-degree-Kelvin |

**BACnetEventState**

| Value | Event State |
|---|---|
| 0 | Normal |
| 1 | Fault |
| 2 | Off-normal |
| 3 | High-limit |
| 4 | Low-limit |
| 5 | Life-safety-alarm |

**BACnetEventType**

| Value | Event Type |
|---|---|
| 0 | Change-of-bitstring |
| 1 | Change-of-state |
| 2 | Change-of-value |
| 3 | Command-failure |
| 4 | Floating-limit |
| 5 | Out-of-range |
| 6 | Complex-event-type |
| 7 | Buffer-ready |
| 8 | Change-of-life-safety |

**BACnetFileAccessMethod**

| Value | Access Method |
|---|---|
| 0 | Record-access |
| 1 | Stream-access |

**BACnetLifeSafetyMode**

| Value | Life Safety Mode |
|---|---|
| 0 | Off |
| 1 | On |
| 2 | Test |
| 3 | Manned |
| 4 | Unmanned |
| 5 | Armed |
| 6 | Disarmed |
| 7 | Prearmed |
| 8 | Slow |
| 9 | Fast |
| 10 | Disconnected |
| 11 | Enabled |
| 12 | Disabled |
| 13 | Automatic-release-disabled |
| 14 | Default |

**BACnetLifeSafetyOperation**

| Value | Life Safety Operation |
|---|---|
| 0 | None |
| 1 | Silence |
| 2 | Silence-audible |
| 3 | Silence-visible |
| 4 | Reset |
| 5 | Reset-alarm |

| | |
|---|---|
| 6 | Reset-fault |
| 7 | Unsilence |
| 8 | Unsilence-audible |
| 9 | Unsilence-visual |

**BACnetLifeSafetyState**

| Value | Life Safety State |
|---|---|
| 0 | Quiet |
| 1 | Pre-alarm |
| 2 | Alarm |
| 3 | Fault |
| 4 | Fault-pre-alarm |
| 5 | Fault-alarm |
| 6 | Not-ready |
| 7 | Active |
| 8 | Tamper |
| 9 | Test-alarm |
| 10 | Test-active |
| 11 | Test-fault |
| 12 | Test-fault-alarm |
| 13 | Holdup |
| 14 | Duress |
| 15 | Tamper-alarm |
| 16 | Abnormal |
| 17 | Emergency-power |
| 18 | Delayed |
| 19 | Blocked |
| 20 | Local-alarm |
| 21 | General-alarm |
| 22 | Supervisory |
| 23 | Test-supervisory |

**BACnetMaintenance**

| Value | Maintenance |
|---|---|
| 0 | None |
| 1 | Periodic-test |
| 2 | Need-service-operational |
| 3 | Need-service-inoperative |

**BACnetNotifyType**

| Value | Notify Type |
|---|---|
| 0 | Alarm |
| 1 | Event |
| 2 | Lack-notification |

**BACnetObjectType**

| Value | Object Type |
|---|---|
| 0 | Analog-input |
| 1 | Analog-output |
| 2 | Analog-value |
| 3 | Binary-input |
| 4 | Binary-output |
| 5 | Binary-value |
| 6 | Calendar |
| 7 | Command |

| | |
|---|---|
| 8 | Device |
| 9 | Event-enrollment |
| 10 | File |
| 11 | Group |
| 12 | Loop |
| 13 | Multi-state-input |
| 14 | Multi-state-output |
| 15 | Notification-class |
| 16 | Program |
| 17 | Schedule |
| 18 | Averaging |
| 19 | Multi-state-value |
| 20 | Trend-log |
| 21 | Life-safety-point |
| 22 | Life-safety-zone |

**BACnetPolarity**

| Value | Polarity |
|---|---|
| 0 | Normal |
| 1 | Reverse |

**BACnetProgramError**

| Value | Program Error |
|---|---|
| 0 | Normal |
| 1 | Load-failed |
| 2 | Internal |
| 3 | Program |
| 4 | Other |

**BACnetProgramRequest**

| Value | Program Request |
|---|---|
| 0 | Ready |
| 1 | Load |
| 2 | Run |
| 3 | Halt |
| 4 | Restart |
| 5 | Unload |

**BACnetProgramState**

| Value | Program State |
|---|---|
| 0 | Idle |
| 1 | Loading |
| 2 | Running |
| 3 | Waiting |
| 4 | Halted |
| 5 | Unloading |

**BACnetReliability**

| Value | Reliability |
|---|---|
| 0 | No-fault-detected |
| 1 | No-sensor |
| 2 | Over-range |
| 3 | Under-range |
| 4 | Open-loop |
| 5 | Shorted-loop |

| | |
|---|---|
| 6 | No-output |
| 7 | Unreliable-other |
| 8 | Process-error |
| 9 | Multi-state-fault |
| 10 | Configuration-error |
| 12 | Communication-failure |

**BACnetSegmentation**

| Value | Segmentation |
|---|---|
| 0 | Segmented-both |
| 1 | Segmented-transmit |
| 2 | Segmented-receive |
| 3 | No-segmentation |

**BACnetSilencedState**

| Value | Silenced State |
|---|---|
| 0 | Unsilenced |
| 1 | Audible-silenced |
| 2 | Visible-silenced |
| 3 | All-silenced |

**BACnetVTClass**

| Value | VT Class |
|---|---|
| 0 | Default-terminal |
| 1 | Ansi-x3-64 |
| 2 | Dec-vt52 |
| 3 | Dec-vt100 |
| 4 | Dec-vt220 |
| 5 | Hp-700-94 |
| 6 | Ibm-3130 |

## Address Descriptions

All addresses have three required files consisting of object type, object instance, and property identifier. Additional fields may be required for some properties. Many addresses may take optional fields.

### Basic Addresses (Primitive Data Types)

Properties with primitive data types are addressed using the following format:

*<object type>.<object instance>.<property identifier>*

- The **object type** field contains a mnemonic from the list of supported BACnet objects.
- The **object instance** field is the numerical object instance. Object instances may range from 0 to 4194303.
- The **property identifier** field contains the mnemonic for a property that is a member of the selected object type.

**See Also:** **BACnet/IP Objects**

### BACnet Array and List Addresses

Elements of arrays and lists are addressed using the following format:

*<object type>.<object instance>.<property identifier>[index]*

This data exists in array or list form in the BACnet/IP device, and not the OPC server. One tag must be configured for each array or list element. The data will not be presented to the OPC clients in array form because BACnet array and list elements may not have primitive data types. The driver will attempt to optimize reads of array data, meaning that it will generally read data for all referenced elements in a single transaction. BACnet lists must be read in their entirety, regardless of the number of elements needed. Element indices start at 1. The upper limit depends on both the property for arrays and the device configuration for lists.

### Complex Addresses (Structured Data Types)

Elements of structured data types are addressed using the following format:

*<object type>.<object instance>.<property identifier>.<sub property 1>.<sub property 2> ...*

- The **sub property n** fields are one of the mnemonics given in the links from the supported object types. For more information, refer to **BACnet/IP Objects**.

### Address Modifiers

Optional address modifiers may be added to alter the behavior of the driver. The property address and modifier must be separated by a single space character. The available address modifiers are as follows:

- **COV:** If this modifier is present, the driver will attempt to subscribe to change of value (COV) notifications for the addressed property. Use of COV notifications over polling can greatly reduce network traffic. The BACnet specification requires that certain properties must support COV, but does not demand that others cannot. A particular device may offer COV support for any property. This modifier is primarily for these non-standard COV properties. The driver can be configured to assume COV is to be used for certain standard COV properties, regardless of the presence of this modifier. This behavior can be turned on or off with the COV mode device setting. For more information, refer to the device's PICS statement.
- **RELINQUISH:** BACnet requires that a device execute writes according to a command priority. Once a write has been executed, the issuing application retains "command" over that property. That is, no other application can write to that property unless it uses a higher priority, or the "commanding" application "relinquishes" its command over the property. A tag with this address modifier can be used to relinquish command over the addressed property. Such tags are Write Only, and have a default data type of Boolean. The driver will issue a relinquish command request when any value is written to this tag. Command over other properties will not be affected.

**See Also:** **BACnet/IP Objects**

## Addressing Examples

The following examples assume an analog value object with instance number 100. For more information on the object, refer to **Analog Value**.

1. **AnalogValue.100.PresentValue** addresses the **Present Value** property. Since this property has "implicit COV," the driver may subscribe to COV notifications of this property or continuously poll for its current value.
2. **AnalogValue.100.OutOfService COV** addresses the **Out Of Service** property. The COV address modifier is used to tell the driver that COV reporting can be used for this property, even though this property does not normally support COV.
3. **AnalogValue.100.PresentValue RELINQUISH** address is used to create a Write Only tag for relinquishing the driver's command over the **Present Value** property. For more information, refer to **Command Settings**.
4. **AnalogValue.100.PriorityArray [4]** addresses element 4 of the **Priority Array**.
5. **AnalogValue.100.EventEnable.ToFault** addresses the **To Fault** element of the **Event Enable** bit string property.
6. **AnalogValue.100.EventEnable** addresses all bits of the Event Enable bit string property, packed as Word value. Only the lowest 3 bits of the Word will be meaningful in this case. These will be **ToOffNormal**, **ToFault**, and **ToNormal** respectively.
7. **AnalogValue.100.ObjectIdentifier.ObjectInstance** addresses the **Object Instance** member of the **Object Identifier** property structure.
8. **AnalogValue.100.ObjectIdentifier**addresses the Object Identifier property structure, and packs its member values into a single DWord value. The high 10 bits will be the **Object Type** member, and the low 22 bits will be the Object Instance member.

**Note:** For more information on enabling COV in the BACnet/IP Driver, refer to **COV Notifications** and **COV Settings**.

## BACnet/IP Objects

For more information on a specific BACnet/IP object, select a link from the list below.

**Analog Input**
**Analog Output**
**Analog Value**
**Averaging**
**Binary Input**
**Binary Output**
**Binary Value**
**Calendar**
**Command**
**Device**
**Event Enrollment**
**File**
**Group**
**Life Safety Point**
**Life Safety Zone**
**Loop**
**Multi-State Input**
**Multi-State Output**
**Multi-State Value**
**Notification Class**
**Program**
**Schedule**
**Trend Log**

## Analog Input

The following table describes the object's supported properties. The Access column specifies the default access permission for tags. To accommodate non-standard implementations of BACnet, tags may be given Read/Write access unless noted otherwise. The COV column specifies whether the driver considers the property to have implicit Change Of Value (COV) notification capability; that is, whether the BACnet specification requires the prop-

erty to support COV. For some properties, COV support depends on implementation. The "COV" modifier must be added to the tag's address for use. For more information, refer to **COV Settings**.

**Note:** The length of the array property will be specified by *[m]*, where m is the number of supported elements (according to the BACnet specification). The BACnet array properties that do not have a length specified by the BACnet standard will be designated by *[N]*. This means the length of the property array depends on the BACnet device. For more information, refer to **Addressing Examples**.

**See Also:** **Address Descriptions**

| Property Mnemonic | BACnet Data Type | OPC Data Type | Access | COV |
|---|---|---|---|---|
| AckedTransitions**<br><br>.ToFault<br>.ToNormal<br>.ToOffNormal | BACnetEventTransitionBits | **Word**, Short<br><br>Boolean<br>Boolean<br>Boolean | Read Only | No |
| COVIncrement | REAL | **Float** | Read/Write | No |
| Deadband | REAL | **Float** | Read/Write | No |
| Description | CharacterString | **String** | Read Only | No |
| DeviceType | CharacterString | **String** | Read Only | No |
| EventEnable**<br><br>.ToFault<br>.ToNormal<br>.ToOffNormal | BACnetEventTransitionBits | **Word**, Short<br><br>Boolean<br>Boolean<br>Boolean | Read/Write | No |
| EventState | **BACnetEventState**** | **DWord**, Long | Read Only | No |
| EventTimeStamps[3] | Array of TimeStamp | **String** | Read Only* | No |
| HighLimit | REAL | **Float** | Read Only | No |
| LimitEnable**<br><br>.HighLimitEnable<br>.LowLimitEnable | BACnetLimitEnable | **Word**, Short<br><br>Boolean<br>Boolean | Read/Write | No |
| LowLimit | REAL | **Float** | Read Only | No |
| MaxPresValue | REAL | **Float** | Read Only | No |
| MinPresValue | REAL | **Float** | Read Only | No |
| NotificationClass | Unsigned | **DWord**, Long | Read Only | No |
| NotifyType | **BACnetNotifyType**** | **DWord**, Long | Read Only | No |
| ObjectIdentifier<br><br>.ObjectInstance<br>.ObjectType | BACnetObjectIdentifier | **DWord**, Long<br><br>**DWord**, Long<br>**DWord**, Long | Read Only* | No |
| ObjectName | CharacterString | **String** | Read Only | No |
| ObjectType | **BACnetObjectType**** | **DWord**, Long | Read Only* | No |
| OutOfService | Boolean | **Boolean** | Read/Write | No |
| PresentValue | REAL | **Float** | Read/Write | Yes |
| ProfileName | CharacterString | **String** | Read Only | No |
| Reliability | **BACnetReliability**** | **DWord**, Long | Read Only | No |
| Resolution | REAL | **Float** | Read Only | No |
| StatusFlags**<br><br>.Fault<br>.InAlarm<br>.OutOfService<br>.Overridden | BACnetStatusFlags | **Word**, Short<br><br>Boolean<br>Boolean<br>Boolean<br>Boolean | Read Only | Yes |
| TimeDelay | Unsigned | **DWord**, Long | Read/Write | No |
| Units | **BACnetEngineeringUnits**** | **DWord**, Long | Read Only | No |
| UpdateInterval | Unsigned | **DWord**, Long | Read/Write | No |

*May not be made writable.

**Bit string types may be viewed as a packed Word value, whose actual number of meaningful bits will depend on specific property. They may also be viewed as individual bits using optional sub-property fields.

***Properties with enumerated BACnet Data Types are expressed as integer values. For standard interpretations, refer to **Enumerated Data Types**.

**Note:** Object Identifier may be viewed as a packed DWord value (high 10 bits are the object type and low 22 bits are the object instance) or as individual tags for object type and instance using optional sub-property fields.

### Priority Array Elements

Priority Array elements may be "NULL" or the numerical command value currently in effect. The array element index may range from 1 to 16, inclusive.

## Analog Output

The following table describes the object's supported properties. The Access column specifies the default access permission for tags. To accommodate non-standard implementations of BACnet, tags may be given Read/Write access unless noted otherwise. The COV column specifies whether the driver considers the property to have implicit Change Of Value (COV) notification capability; that is, whether the BACnet specification requires the property to support COV. For some properties, COV support depends on implementation. The "COV" modifier must be added to the tag's address for use. For more information, refer to **COV Settings**.

**Note:** The length of the array property will be specified by *[m]*, where m is the number of supported elements (according to the BACnet specification). The BACnet array properties that do not have a length specified by the BACnet standard will be designated by *[N]*. This means the length of the property array depends on the BACnet device. For more information, refer to **Addressing Examples**.

**See Also:** **Address Descriptions**

| Property Mnemonic | BACnet Data Type | OPC Data Type | Access | COV |
|---|---|---|---|---|
| AckedTransitions***<br><br>.ToFault<br>.ToNormal<br>.ToOffNormal | BACnetEventTransitionBits | **Word**, Short<br><br>Boolean<br>Boolean<br>Boolean | Read Only | No |
| COVIncrement | REAL | **Float** | Read/Write | No |
| Deadband | REAL | **Float** | Read/Write | No |
| Description | CharacterString | **String** | Read Only | No |
| DeviceType | CharacterString | **String** | Read Only | No |
| EventEnable***<br><br>.ToFault<br>.ToNormal<br>.ToOffNormal | BACnetEventTransitionBits | **Word**, Short<br><br>Boolean<br>Boolean<br>Boolean | Read/Write | No |
| EventState | **BACnetEventState**** | **DWord**, Long | Read Only | No |
| EventTimeStamps[3] | Array of TimeStamp | **String** | Read Only* | No |
| HighLimit | REAL | **Float** | Read Only | No |
| LimitEnable***<br><br>.HighLimitEnable<br>.LowLimitEnable | BACnetLimitEnable | **Word**, Short<br><br>Boolean<br>Boolean | Read/Write | No |
| LowLimit | REAL | **Float** | Read Only | No |
| MaxPresValue | REAL | **Float** | Read Only | No |
| MinPresValue | REAL | **Float** | Read Only | No |
| NotificationClass | Unsigned | **DWord**, Long | Read Only | No |
| NotifyType | **BACnetNotifyType**** | **DWord**, Long | Read Only | No |
| ObjectIdentifier**<br><br>.ObjectInstance<br>.ObjectType | BACnetObjectIdentifier | **DWord**, Long<br><br>**DWord**, Long<br>**DWord**, Long | Read Only* | No |
| ObjectName | CharacterString | **String** | Read Only | No |
| ObjectType | **BACnetObjectType**** | **DWord**, Long | Read Only* | No |

| | | | | |
|---|---|---|---|---|
| OutOfService | Boolean | **Boolean** | Read/Write | No |
| PresentValue | REAL | **Float** | Read/Write | Yes |
| PriorityArray[16] | Array of BACnetPriorityArray | **String** | Read Only* | No |
| ProfileName | CharacterString | **String** | Read Only | No |
| Reliability | **BACnetReliability**** | **DWord**, Long | Read Only | No |
| RelinquishDefault | REAL | **Float** | Read/Write | No |
| Resolution | REAL | **Float** | Read Only | No |
| StatusFlags***<br><br>.Fault<br>.InAlarm<br>.OutOfService<br>.Overridden | BACnetStatusFlags | **Word**, Short<br><br>Boolean<br>Boolean<br>Boolean<br>Boolean | Read Only | Yes |
| TimeDelay | Unsigned | **DWord**, Long | Read/Write | No |
| Units | **BACnetEngineeringUnits**** | **DWord**, Long | Read Only | No |

*May not be made writable.

**Object Identifier may be viewed as a packed DWord value (high 10 bits are the object type and low 22 bits are the object instance) or as individual tags for object type and instance using optional sub-property fields.

***Bit string types may be viewed as a packed Word value, whose actual number of meaningful bits will depend on specific property. They may also be viewed as individual bits using optional sub-property fields.

****Properties with enumerated BACnet Data Types are expressed as integer values. For standard interpretations, refer to **Enumerated Data Types**.

### Priority Array Elements

Priority Array elements may be "NULL" or the numerical command value currently in effect. The array element index may range from 1 to 16, inclusive.

## Analog Value

The following table describes the object's supported properties. The Access column specifies the default access permission for tags. To accommodate non-standard implementations of BACnet, tags may be given Read/Write access unless noted otherwise. The COV column specifies whether the driver considers the property to have implicit Change Of Value (COV) notification capability; that is, whether the BACnet specification requires the property to support COV. For some properties, COV support depends on implementation. The "COV" modifier must be added to the tag's address for use. For more information, refer to **COV Settings**.

**Note:** The length of the array property will be specified by *[m]*, where m is the number of supported elements (according to the BACnet specification). The BACnet array properties that do not have a length specified by the BACnet standard will be designated by *[N]*. This means the length of the property array depends on the BACnet device. For more information, refer to **Addressing Examples**.

**See Also:** **Address Descriptions**

| Property Mnemonic | BACnet Data Type | OPC Data Type | Access | COV |
|---|---|---|---|---|
| AckedTransitions***<br><br>.ToFault<br>.ToNormal<br>.ToOffNormal | BACnetEventTransitionBits | **Word**, Short<br><br>Boolean<br>Boolean<br>Boolean | Read Only | No |
| COVIncrement | REAL | **Float** | Read/Write | No |
| Deadband | REAL | **Float** | Read/Write | No |
| Description | CharacterString | **String** | Read Only | No |
| EventEnable***<br><br>.ToFault<br>.ToNormal<br>.ToOffNormal | BACnetEventTransitionBits | **Word**, Short<br><br>Boolean<br>Boolean<br>Boolean | Read/Write | No |
| EventState | **BACnetEventState**** | **DWord**, Long | Read Only | No |

| | | | | |
|---|---|---|---|---|
| EventTimeStamps[3] | Array of TimeStamp | **String** | Read Only* | No |
| HighLimit | REAL | **Float** | Read Only | No |
| LimitEnable***<br><br>.HighLimitEnable<br>.LowLimitEnable | BACnetLimitEnable | **Word**, Short<br><br>Boolean<br>Boolean | Read/Write | No |
| LowLimit | REAL | **Float** | Read Only | No |
| NotificationClass | Unsigned | **DWord**, Long | Read Only | No |
| NotifyType | **BACnetNotifyType**** | **DWord**, Long | Read Only | No |
| ObjectIdentifier**<br><br>.ObjectInstance<br>.ObjectType | BACnetObjectIdentifier | **DWord**, Long<br><br>**DWord**, Long<br>**DWord**, Long | Read Only* | No |
| ObjectName | CharacterString | **String** | Read Only | No |
| ObjectType | **BACnetObjectType**** | **DWord**, Long | Read Only* | No |
| OutOfService | Boolean | **Boolean** | Read/Write | No |
| PresentValue | REAL | **Float** | Read/Write | Yes |
| PriorityArray[16] | Array of BACnetPriorityArray | **String** | Read Only* | No |
| ProfileName | CharacterString | **String** | Read Only | No |
| Reliability | **BACnetReliability**** | **DWord**, Long | Read Only | No |
| RelinquishDefault | REAL | **Float** | Read/Write | No |
| StatusFlags***<br><br>.Fault<br>.InAlarm<br>.OutOfService<br>.Overridden | BACnetStatusFlags | **Word**, Short<br><br>Boolean<br>Boolean<br>Boolean<br>Boolean | Read Only | Yes |
| TimeDelay | Unsigned | **DWord**, Long | Read/Write | No |
| Units | **BACnetEngineeringUnits**** | **DWord**, Long | Read Only | No |

*May not be made writable.

**Object Identifier may be viewed as a packed DWord value (high 10 bits are the object type and low 22 bits are the object instance) or as individual tags for object type and instance using optional sub-property fields.

***Bit string types may be viewed as a packed Word value, whose actual number of meaningful bits will depend on specific property. They may also be viewed as individual bits using optional sub-property fields.

****Properties with enumerated BACnet Data Types are expressed as integer values. For standard interpretations, refer to **Enumerated Data Types**.

### Priority Array Elements

Priority Array elements may be "NULL" or the numerical command value currently in effect. The array element index may range from 1 to 16, inclusive.

## Averaging

The following table describes the object's supported properties. The Access column specifies the default access permission for tags. To accommodate non-standard implementations of BACnet, tags may be given Read/Write access unless noted otherwise.

**Note 1:** The length of the array property will be specified by *[m]*, where m is the number of supported elements (according to the BACnet specification). The BACnet array properties that do not have a length specified by the BACnet standard will be designated by *[N]*. This means the length of the property array depends on the BACnet device. For more information, refer to **Addressing Examples**.

**Note 2:** Implicit Change of Value (COV) notifications are not supported for this object. For more information on devices with explicit COV support, refer to **COV Notifications**.

**See Also:** **Address Descriptions**

| Property Mnemonic | BACnet Data Type | OPC Data Type | Access |
|---|---|---|---|
| AttemptedSamples | Unsigned | **DWord**, Long | Read Only |

| | | | |
|---|---|---|---|
| AverageValue | REAL | **Float** | Read Only |
| Description | CharacterString | **String** | Read Only |
| MaximumValue | REAL | **Float** | Read Only |
| MaximumValueTimestamp | BACnetDateTime | **String** | Read Only* |
| MinimumValue | REAL | **Float** | Read Only |
| MinimumValueTimestamp | BACnetDateTime | **String** | Read Only* |
| ObjectIdentifier**<br><br>.ObjectInstance<br>.ObjectType | BACnetObjectIdentifier | **DWord**, Long<br><br>**DWord**, Long<br>**DWord**, Long | Read Only* |
| ObjectName | CharacterString | **String** | Read Only |
| ObjectType | **BACnetObjectType**** | **DWord**, Long | Read Only* |
| ProfileName | CharacterString | **String** | Read Only |
| ValidSamples | Unsigned | **DWord**, Long | Read Only |
| VarianceValue | REAL | **Float** | Read Only |
| WindowInterval | Unsigned | **DWord**, Long | Read/Write |
| WindowSamples | Unsigned | **DWord**, Long | Read/Write |

*May not be made writable.

**Object Identifier may be viewed as a packed DWord value (high 10 bits are the object type and low 22 bits are the object instance) or as individual tags for object type and instance using optional sub-property fields.

***Properties with enumerated BACnet Data Types are expressed as integer values. For standard interpretations, refer to **Enumerated Data Types**.

**Note:** Bit string types may be viewed as a packed Word value, whose actual number of meaningful bits will depend on specific property. They may also be viewed as individual bits using optional sub-property fields.

### Priority Array Elements

Priority Array elements may be "NULL" or the numerical command value currently in effect. The array element index may range from 1 to 16, inclusive.

## Binary Input

The following table describes the object's supported properties. The Access column specifies the default access permission for tags. To accommodate non-standard implementations of BACnet, tags may be given Read/Write access unless noted otherwise. The COV column specifies whether the driver considers the property to have implicit Change Of Value (COV) notification capability; that is, whether the BACnet specification requires the property to support COV. For some properties, COV support depends on implementation. The "COV" modifier must be added to the tag's address for use. For more information, refer to **COV Settings**.

**Note:** The length of the array property will be specified by *[m]*, where m is the number of supported elements (according to the BACnet specification). The BACnet array properties that do not have a length specified by the BACnet standard will be designated by *[N]*. This means the length of the property array depends on the BACnet device. For more information, refer to **Addressing Examples**.

**See Also:** **Address Descriptions**

| Property Mnemonic | BACnet Data Type | OPC Data Type | Access | COV |
|---|---|---|---|---|
| AckedTransitions***<br><br>.ToFault<br>.ToNormal<br>.ToOffNormal | BACnetEventTransitionBits | **Word**, Short<br><br>Boolean<br>Boolean<br>Boolean | Read Only | No |
| ActiveText | CharacterString | **String** | Read Only | No |
| AlarmValue | **BACnetBinaryPV**** | **Boolean** | Read/Write | No |
| ChangeOfStateCount | Unsigned | **DWord**, Long | Read/Write | No |
| ChangeOfStateTime | BACnetDataTime | **String** | Read Only* | No |
| Description | CharacterString | **String** | Read Only | No |
| DeviceType | CharacterString | **String** | Read Only | No |
| ElapsedActiveTime | Unsigned | **DWord**, Long | Read Only | No |

| | | | | |
|---|---|---|---|---|
| EventEnable***<br><br>.ToFault<br>.ToNormal<br>.ToOffNormal | BACnetEventTransitionBits | **Word**, Short<br><br>Boolean<br>Boolean<br>Boolean | Read/Write | No |
| EventState | **BACnetEventState**** | **DWord**, Long | Read Only | No |
| EventTimeStamps[3] | Array of TimeStamp | **String** | Read Only* | No |
| InactiveText | CharacterString | **String** | Read Only | No |
| NotificationClass | Unsigned | **DWord**, Long | Read Only | No |
| NotifyType | **BACnetNotifyType**** | **DWord**, Long | Read Only | No |
| ObjectIdentifier**<br><br>.ObjectInstance<br>.ObjectType | BACnetObjectIdentifier | **DWord**, Long<br><br>**DWord**, Long<br>**DWord**, Long | Read Only* | No |
| ObjectName | CharacterString | **String** | Read Only | No |
| ObjectType | **BACnetObjectType**** | **DWord**, Long | Read Only* | No |
| OutOfService | Boolean | **Boolean** | Read/Write | No |
| Polarity | **BACnetPolarity**** | **Boolean** | Read/Write | No |
| PresentValue | **BACnetBinaryPV**** | **Boolean** | Read/Write | Yes |
| ProfileName | CharacterString | **String** | Read Only | No |
| Reliability | **BACnetReliability**** | **DWord**, Long | Read Only | No |
| StatusFlags***<br><br>.Fault<br>.InAlarm<br>.OutOfService<br>.Overridden | BACnetStatusFlags | **Word**, Short<br><br>Boolean<br>Boolean<br>Boolean<br>Boolean | Read Only | Yes |
| TimeDelay | Unsigned | **DWord**, Long | Read/Write | No |
| TimeOfActiveTimeReset | BACnetDateTime | **String** | Read Only* | No |
| TimeOfStateCountReset | BACnetDateTime | **String** | Read Only* | No |

*May not be made writable.

**Object Identifier may be viewed as a packed DWord value (high 10 bits are the object type and low 22 bits are the object instance) or as individual tags for object type and instance using optional sub-property fields.

***Bit string types may be viewed as a packed Word value, whose actual number of meaningful bits will depend on specific property. They may also be viewed as individual bits using optional sub-property fields.

****Properties with enumerated BACnet Data Types are expressed as integer values. For standard interpretations, refer to **Enumerated Data Types**.

### Priority Array Elements

Priority Array elements may be "NULL" or the numerical command value currently in effect. The array element index may range from 1 to 16, inclusive.

## Binary Output

The following table describes the object's supported properties. The Access column specifies the default access permission for tags. To accommodate non-standard implementations of BACnet, tags may be given Read/Write access unless noted otherwise. The COV column specifies whether the driver considers the property to have implicit Change Of Value (COV) notification capability; that is, whether the BACnet specification requires the property to support COV. For some properties, COV support depends on implementation. The "COV" modifier must be added to the tag's address for use. For more information, refer to **COV Settings**.

**Note:** The length of the array property will be specified by *[m]*, where m is the number of supported elements (according to the BACnet specification). The BACnet array properties that do not have a length specified by the BACnet standard will be designated by *[N]*. This means the length of the property array depends on the BACnet device. For more information, refer to **Addressing Examples**.

**See Also:** **Address Descriptions**

| Property Mnemonic | BACnet Data Type | OPC Data Type | Access | COV |
|---|---|---|---|---|
| AckedTransitions*** | BACnetEventTransitionBits | **Word**, Short | Read Only | No |
| .ToFault | | Boolean | | |
| .ToNormal | | Boolean | | |
| .ToOffNormal | | Boolean | | |
| ActiveText | CharacterString | **String** | Read Only | No |
| ChangeOfStateCount | Unsigned | **DWord**, Long | Read/Write | No |
| ChangeOfStateTime | BACnetDateTime | **String** | Read Only* | No |
| Description | CharacterString | **String** | Read Only | No |
| DeviceType | CharacterString | **String** | Read Only | No |
| ElapsedActiveTime | Unsigned | **DWord**, Long | Read Only | No |
| EventEnable*** | BACnetEventTransitionBits | **Word**, Short | Read/Write | No |
| .ToFault | | Boolean | | |
| .ToNormal | | Boolean | | |
| .ToOffNormal | | Boolean | | |
| EventState | **BACnetEventState**** | **DWord**, Long | Read Only | No |
| EventTimeStamps[3] | Array of TimeStamp | **String** | Read Only* | No |
| FeedbackValue | **BACnetBinaryPV**** | **Boolean** | Read/Write | No |
| InactiveText | CharacterString | **String** | Read Only | No |
| MinimumOffTime | Unsigned | **DWord**, Long | Read Only | No |
| MinimumOnTime | Unsigned | **DWord**, Long | Read Only | No |
| NotificationClass | Unsigned | **DWord**, Long | Read Only | No |
| NotifyType | **BACnetNotifyType**** | **DWord**, Long | Read Only | No |
| ObjectIdentifier** | BACnetObjectIdentifier | **DWord**, Long | Read Only* | No |
| .ObjectInstance | | **DWord**, Long | | |
| .ObjectType | | **DWord**, Long | | |
| ObjectName | CharacterString | **String** | Read Only | No |
| ObjectType | **BACnetObjecType**** | **DWord**, Long | Read Only* | No |
| OutOfService | Boolean | **Boolean** | Read/Write | No |
| Polarity | **BACnetPolarity**** | **Boolean** | Read/Write | No |
| PresentValue | **BACnetBinaryPV**** | **Boolean** | Read/Write | Yes |
| PriorityArray[16] | Array of BACnetPriorityArray | **String** | Read Only* | No |
| ProfileName | CharacterString | **String** | Read Only | No |
| Reliability | **BACnetReliability**** | **DWord**, Long | Read Only | No |
| RelinquishDefault | **BACnetBinaryPV**** | **Boolean** | Read/Write | No |
| StatusFlags*** | BACnetStatusFlags | **Word**, Short | Read Only | Yes |
| .Fault | | Boolean | | |
| .InAlarm | | Boolean | | |
| .OutOfService | | Boolean | | |
| .Overridden | | Boolean | | |
| TimeDelay | Unsigned | **DWord**, Long | Read/Write | No |
| TimeOfActiveTimeReset | BACnetDateTime | **String** | Read Only* | No |
| TimeOfStateCountReset | BACnetDateTime | **String** | Read Only* | No |

*May not be made writable.

**Object Identifier may be viewed as a packed DWord value (high 10 bits are the object type and low 22 bits are the object instance) or as individual tags for object type and instance using optional sub-property fields.

***Bit string types may be viewed as a packed Word value, whose actual number of meaningful bits will depend on specific property. They may also be viewed as individual bits using optional sub-property fields.

****Properties with enumerated BACnet Data Types are expressed as integer values. For standard interpretations, refer to **Enumerated Data Types**.

### Priority Array Elements

Priority Array elements may be "NULL" or the numerical command value currently in effect. The array element index may range from 1 to 16, inclusive.

## Binary Value

The following table describes the object's supported properties. The Access column specifies the default access permission for tags. To accommodate non-standard implementations of BACnet, tags may be given Read/Write access unless noted otherwise. The COV column specifies whether the driver considers the property to have implicit Change Of Value (COV) notification capability; that is, whether the BACnet specification requires the property to support COV. For some properties, COV support depends on implementation. The "COV" modifier must be added to the tag's address for use. For more information, refer to **COV Settings**.

**Note:** The length of the array property will be specified by *[m]*, where m is the number of supported elements (according to the BACnet specification). The BACnet array properties that do not have a length specified by the BACnet standard will be designated by *[N]*. This means the length of the property array depends on the BACnet device. For more information, refer to **Addressing Examples**.

**See Also:** **Address Descriptions**

| Property Mnemonic | BACnet Data Type | OPC Data Type | Access | COV |
|---|---|---|---|---|
| AckedTransitions***<br><br>.ToFault<br>.ToNormal<br>.ToOffNormal | BACnetEventTransitionBits | **Word**, Short<br><br>Boolean<br>Boolean<br>Boolean | Read Only | No |
| ActiveText | CharacterString | **String** | Read Only | No |
| AlarmValue | **BACnetBinaryPV**** | **Boolean** | Read/Write | No |
| ChangeOfStateCount | Unsigned | **DWord**, Long | Read/Write | No |
| ChangeOfStateTime | BACnetDateTime | **String** | Read Only* | No |
| Description | CharacterString | **String** | Read Only | No |
| ElapsedActiveTime | Unsigned | **DWord**, Long | Read Only | No |
| EventEnable***<br><br>.ToFault<br>.ToNormal<br>.ToOffNormal | BACnetEventTransitionBits | **Word**, Short<br><br>Boolean<br>Boolean<br>Boolean | Read/Write | No |
| EventState | **BACnetEventState**** | **DWord**, Long | Read Only | No |
| EventTimeStamps[3] | Array of TimeStamp | **String** | Read Only* | No |
| InactiveText | CharacterString | **String** | Read Only | No |
| MinimumOffTime | Unsigned | **DWord**, Long | Read Only | No |
| MinimumOnTime | Unsigned | **DWord**, Long | Read Only | No |
| NotificationClass | Unsigned | **DWord**, Long | Read Only | No |
| NotifyType | **BACnetNotifyType**** | **DWord**, Long | Read Only | No |
| ObjectIdentifier**<br><br>.ObjectInstance<br>.ObjectType | BACnetObjectIdentifier | **DWord**, Long<br><br>**DWord**, Long<br>**DWord**, Long | Read Only* | No |
| ObjectName | CharacterString | **String** | Read Only | No |
| ObjectType | **BACnetObjectType**** | **DWord**, Long | Read Only* | No |
| OutOfService | Boolean | **Boolean** | Read/Write | No |
| PresentValue | **BACnetBinaryPV**** | **Boolean** | Read/Write | Yes |
| PriorityArray[16] | Array of BACnetPriorityArray | **String** | Read Only* | No |
| ProfileName | CharacterString | **String** | Read Only | No |
| Reliability | **BACnetReliability**** | **DWord**, Long | Read Only | No |
| RelinquishDefault | **BACnetBinaryPV**** | **Boolean** | Read/Write | No |
| StatusFlags***<br><br>.Fault<br>.InAlarm<br>.OutOfService<br>.Overridden | BACnetStatusFlags | **Word**, Short<br><br>Boolean<br>Boolean<br>Boolean<br>Boolean | Read Only | Yes |
| TimeDelay | Unsigned | **DWord**, Long | Read/Write | No |

| | | | | |
|---|---|---|---|---|
| TimeOfActiveTimeReset | BACnetDateTime | **String** | Read Only* | No |
| TimeOfStateCountReset | BACnetDateTime | **String** | Read Only* | No |

*May not be made writable.

**Object Identifier may be viewed as a packed DWord value (high 10 bits are the object type and low 22 bits are the object instance) or as individual tags for object type and instance using optional sub-property fields.

***Bit string types may be viewed as a packed Word value, whose actual number of meaningful bits will depend on specific property. They may also be viewed as individual bits using optional sub-property fields.

****Properties with enumerated BACnet Data Types are expressed as integer values. For standard interpretations, refer to **Enumerated Data Types**.

### Priority Array Elements

Priority Array elements may be "NULL" or the numerical command value currently in effect. The array element index may range from 1 to 16, inclusive.

## Calendar

The following table describes the object's supported properties. The Access column specifies the default access permission for tags. To accommodate non-standard implementations of BACnet, tags may be given Read/Write access unless noted otherwise.

**Note 1:** The length of the array property will be specified by *[m]*, where m is the number of supported elements (according to the BACnet specification). The BACnet array properties that do not have a length specified by the BACnet standard will be designated by *[N]*. This means the length of the property array depends on the BACnet device. For more information, refer to **Addressing Examples**.

**Note 2:** Implicit Change of Value (COV) notifications are not supported for this object. For more information on devices with explicit COV support, refer to **COV Notifications**.

**See Also:** **Address Descriptions**

| Property Mnemonic | BACnet Data Type | OPC Data Type | Access |
|---|---|---|---|
| DateList | List of CalendarEntry | **String** | Read Only* |
| Description | CharacterString | **String** | Read Only |
| ObjectIdentifier**<br><br>.ObjectInstance<br>.ObjectType | BACnetObjectIdentifier | **DWord**, Long<br><br>**DWord**, Long<br>**DWord**, Long | Read Only* |
| ObjectName | CharacterString | **String** | Read Only |
| ObjectType | **BACnetObjectType**** | **DWord**, Long | Read Only* |
| PresentValue | Boolean | **Boolean** | Read/Write |
| ProfileName | CharacterString | **String** | Read Only |

*May not be made writable.

**Object Identifier may be viewed as a packed DWord value (high 10 bits are the object type and low 22 bits are the object instance) or as individual tags for object type and instance using optional sub-property fields.

***Properties with enumerated BACnet Data Types are expressed as integer values. For standard interpretations, refer to **Enumerated Data Types**.

**Note:** Bit string types may be viewed as a packed Word value, whose actual number of meaningful bits will depend on specific property. They may also be viewed as individual bits using optional sub-property fields.

### Priority Array Elements

Priority Array elements may be "NULL" or the numerical command value currently in effect. The array element index may range from 1 to 16, inclusive.

## Command

The following table describes the object's supported properties. The Access column specifies the default access permission for tags. To accommodate non-standard implementations of BACnet, tags may be given Read/Write

access unless noted otherwise.

**Note 1:** The length of the array property will be specified by *[m]*, where m is the number of supported elements (according to the BACnet specification). The BACnet array properties that do not have a length specified by the BACnet standard will be designated by *[N]*. This means the length of the property array depends on the BACnet device. For more information, refer to **Addressing Examples**.

**Note 2:** Implicit Change of Value (COV) notifications are not supported for this object. For more information on devices with explicit COV support, refer to **COV Notifications**.

**See Also:** **Address Descriptions**

| Property Mnemonic | BACnet Data Type | OPC Data Type | Access |
|---|---|---|---|
| ActionText[N] | Array of CharacterString | **String** | Read/Write |
| AllWritesSuccessful | Boolean | **Boolean** | Read Only |
| Description | CharacterString | **String** | Read Only |
| InProcess | Boolean | **Boolean** | Read Only |
| ObjectIdentifier**<br><br>.ObjectInstance<br>.ObjectType | BACnetObjectIdentifier | **DWord**, Long<br><br>**DWord**, Long<br>**DWord**, Long | Read Only* |
| ObjectName | CharacterString | **String** | Read Only |
| ObjectType | **BACnetObjectType**** | **DWord**, Long | Read Only* |
| PresentValue | Unsigned | **DWord**, Long | Read/Write |
| ProfileName | CharacterString | **String** | Read Only |

*May not be made writable.

**Object Identifier may be viewed as a packed DWord value (high 10 bits are the object type and low 22 bits are the object instance) or as individual tags for object type and instance using optional sub-property fields.

***Properties with enumerated BACnet Data Types are expressed as integer values. For standard interpretations, refer to **Enumerated Data Types**.

**Note:** Bit string types may be viewed as a packed Word value, whose actual number of meaningful bits will depend on specific property. They may also be viewed as individual bits using optional sub-property fields.

### Priority Array Elements

Priority Array elements may be "NULL" or the numerical command value currently in effect. The array element index may range from 1 to 16, inclusive.

## Device

The following table describes the object's supported properties. The Access column specifies the default access permission for tags. To accommodate non-standard implementations of BACnet, tags may be given Read/Write access unless noted otherwise.

**Note 1:** The length of the array property will be specified by *[m]*, where m is the number of supported elements (according to the BACnet specification). The BACnet array properties that do not have a length specified by the BACnet standard will be designated by *[N]*. This means the length of the property array depends on the BACnet device. For more information, refer to **Addressing Examples**.

**Note 2:** Implicit Change of Value (COV) notifications are not supported for this object. For more information on devices with explicit COV support, refer to **COV Notifications**.

**See Also:** **Address Descriptions**

| Property Mnemonic | BACnet Data Type | OPC Data Type | Access |
|---|---|---|---|
| APDUSegmentTimeout | Unsigned | **DWord**, Long | Read/Write |
| APDUTimeout | Unsigned | **DWord**, Long | Read/Write |
| ApplicationSoftwareVersion | CharacterString | **String** | Read Only |
| BackupFailureTimeout | Unsigned | **Word**, Short | Read Only |
| ConfigurationFiles[N] | Array of ObjectIdentifier | **DWord**, Long | Read Only |
| DatabaseRevision | Unsigned | **DWord**, Long | Read Only |

| | | | |
|---|---|---|---|
| DaylightSavingsStatus | Boolean | **Boolean** | Read Only |
| Description | CharacterString | **String** | Read Only |
| FirmwareRevision | CharacterString | **String** | Read Only |
| LastRestoreTime | BACnetDateTime | **String** | Read Only* |
| LocalDate | Date | **String** | Read Only* |
| LocalTime | Time | **String** | Read Only* |
| Location | CharacterString | **String** | Read Only |
| MaxAPDULengthAccepted | Unsigned | **DWord**, Long | Read/Write |
| MaxInfoFrames | Unsigned | **DWord**, Long | Read Only |
| MaxMaster | Unsigned | **DWord**, Long | Read Only |
| MaxSegmentsAccepted | Unsigned | **DWord**, Long | Read Only |
| ModelName | CharacterString | **String** | Read Only |
| NumberOfAPDURetries | Unsigned | **DWord**, Long | Read/Write |
| ObjectIdentifier**<br><br>.ObjectInstance<br>.ObjectType | BACnetObjectIdentifier | **DWord**, Long<br><br>**DWord**, Long<br>**DWord**, Long | Read Only* |
| ObjectList[N] | Array of ObjectIdentifier | **DWord**, Long | Read Only* |
| ObjectName | CharacterString | **String** | Read Only |
| ObjectType | **BACnetObjectType**** | **DWord**, Long | Read Only* |
| ProfileName | CharacterString | **String** | Read Only |
| ProtocolRevision | Unsigned | **DWord**, Long | Read Only |
| ProtocolVersion | Unsigned | **DWord**, Long | Read Only |
| SegmentationSupported | **BACnetSegmentation**** | **DWord**, Long | Read Only |
| SystemStatus | **BACnetDeviceStatus**** | **DWord**, Long | Read/Write |
| UTCOffset | INTEGER | **Long**, Word | Read Only |
| VendorIdentifier | Unsigned | **Word**, Short | Read Only |
| VendorName | CharacterString | **String** | Read Only |

*May not be made writable.

**Object Identifier may be viewed as a packed DWord value (high 10 bits are the object type and low 22 bits are the object instance) or as individual tags for object type and instance using optional sub-property fields.

***Properties with enumerated BACnet Data Types are expressed as integer values. For standard interpretations, refer to **Enumerated Data Types**.

**Note:** Bit string types may be viewed as a packed Word value, whose actual number of meaningful bits will depend on specific property. They may also be viewed as individual bits using optional sub-property fields.

### Priority Array Elements

Priority Array elements may be "NULL" or the numerical command value currently in effect. The array element index may range from 1 to 16, inclusive.

## Event Enrollment

The following table describes the object's supported properties. The Access column specifies the default access permission for tags. To accommodate non-standard implementations of BACnet, tags may be given Read/Write access unless noted otherwise.

**Note 1:** The length of the array property will be specified by *[m]*, where m is the number of supported elements (according to the BACnet specification). The BACnet array properties that do not have a length specified by the BACnet standard will be designated by *[N]*. This means the length of the property array depends on the BACnet device. For more information, refer to **Addressing Examples**.

**Note 2:** Implicit Change of Value (COV) notifications are not supported for this object. For more information on devices with explicit COV support, refer to **COV Notifications**.

**See Also:** **Address Descriptions**

| Property Mnemonic | BACnet Data Type | OPC Data Type | Access |
|---|---|---|---|
| AckedTransitions*** | BACnetEventTransitionBits | **Word**, Short | Read Only |

| | | | |
|---|---|---|---|
| .ToFault<br>.ToNormal<br>.ToOffNormal | | Boolean<br>Boolean<br>Boolean | |
| Description | CharacterString | **String** | Read Only |
| EventEnable*** <br><br>.ToFault<br>.ToNormal<br>.ToOffNormal | BACnetEventTransitionBits | **Word**, Short<br><br>Boolean<br>Boolean<br>Boolean | Read/Write |
| EventState | **BACnetEventState**** | **DWord**, Long | Read Only |
| EventTimeStamps[3] | Array of TimeStamp | **String** | Read Only* |
| EventType | **BACnetEventType**** | **DWord**, Long | Read Only |
| IssueConfirmedNotifications | Boolean | **Boolean** | Read/Write |
| NotificationClass | Unsigned | **Word**, Short | Read Only |
| NotifyType | **BACnetNotifyType**** | **DWord**, Long | Read Only |
| ObjectIdentifier** <br><br>.ObjectInstance<br>.ObjectType | BACnetObjectIdentifier | **DWord**, Long<br><br>**DWord**, Long<br>**DWord**, Long | Read Only* |
| ObjectName | CharacterString | **String** | Read Only |
| ObjectType | **BACnetObjectType**** | **DWord**, Long | Read Only* |
| Priority | Unsigned | **DWord**, Long | Read Only |
| ProcessIdentifier | Unsigned | **DWord**, Long | Read/Write |
| ProfileName | CharacterString | **String** | Read Only |

*May not be made writable.

**Object Identifier may be viewed as a packed DWord value (high 10 bits are the object type and low 22 bits are the object instance) or as individual tags for object type and instance using optional sub-property fields.

***Bit string types may be viewed as a packed Word value, whose actual number of meaningful bits will depend on specific property. They may also be viewed as individual bits using optional sub-property fields.

****Properties with enumerated BACnet Data Types are expressed as integer values. For standard interpretations, refer to **Enumerated Data Types**.

### Priority Array Elements

Priority Array elements may be "NULL" or the numerical command value currently in effect. The array element index may range from 1 to 16, inclusive.

## File

The following table describes the object's supported properties. The Access column specifies the default access permission for tags. To accommodate non-standard implementations of BACnet, tags may be given Read/Write access unless noted otherwise.

**Note 1:** The length of the array property will be specified by *[m]*, where m is the number of supported elements (according to the BACnet specification). The BACnet array properties that do not have a length specified by the BACnet standard will be designated by *[N]*. This means the length of the property array depends on the BACnet device. For more information, refer to **Addressing Examples**.

**Note 2:** Implicit Change of Value (COV) notifications are not supported for this object. For more information on devices with explicit COV support, refer to **COV Notifications**.

**See Also:** **Address Descriptions**

| Property Mnemonic | BACnet Data Type | OPC Data Type | Access |
|---|---|---|---|
| Archive | Boolean | **Boolean** | Read Only |
| Description | CharacterString | **String** | Read Only |
| FileAccessMethod | **BACnetFileAccessMethod**** | **DWord**, Long | Read Only |
| FileSize | Unsigned | **DWord**, Long | Read Only |

| | | | |
|---|---|---|---|
| FileType | CharacterString | **String** | Read Only |
| ModificationDate | BACnetDateTime | **String** | Read Only* |
| ObjectIdentifier**<br><br>.ObjectInstance<br>.ObjectType | BACnetObjectIdentifier | **DWord**, Long<br><br>**DWord**, Long<br>**DWord**, Long | Read Only* |
| ObjectName | CharacterString | **String** | Read Only |
| ObjectType | **BACnetObjectType***** | **DWord**, Long | Read Only* |
| ProfileName | CharacterString | **String** | Read Only |
| ReadOnly | Boolean | **Boolean** | Read Only |
| RecordCount | Unsigned | **DWord**, Long | Read/Write |

*May not be made writable.

**Object Identifier may be viewed as a packed DWord value (high 10 bits are the object type and low 22 bits are the object instance) or as individual tags for object type and instance using optional sub-property fields.

***Properties with enumerated BACnet Data Types are expressed as integer values. For standard interpretations, refer to **Enumerated Data Types**.

**Note:** Bit string types may be viewed as a packed Word value, whose actual number of meaningful bits will depend on specific property. They may also be viewed as individual bits using optional sub-property fields.

### Priority Array Elements

Priority Array elements may be "NULL" or the numerical command value currently in effect. The array element index may range from 1 to 16, inclusive.

## Group

The following table describes the object's supported properties. The Access column specifies the default access permission for tags. To accommodate non-standard implementations of BACnet, tags may be given Read/Write access unless noted otherwise.

**Note 1:** The length of the array property will be specified by *[m]*, where m is the number of supported elements (according to the BACnet specification). The BACnet array properties that do not have a length specified by the BACnet standard will be designated by *[N]*. This means the length of the property array depends on the BACnet device. For more information, refer to **Addressing Examples**.

**Note 2:** Implicit Change of Value (COV) notifications are not supported for this object. For more information on devices with explicit COV support, refer to **COV Notifications**.

**See Also:** **Address Descriptions**

| Property Mnemonic | BACnet Data Type | OPC Data Type | Access |
|---|---|---|---|
| Description | CharacterString | **String** | Read Only |
| ObjectIdentifier**<br><br>.ObjectInstance<br>.ObjectType | BACnetObjectIdentifier | **DWord**, Long<br><br>**DWord**, Long<br>**DWord**, Long | Read Only* |
| ObjectName | CharacterString | **String** | Read Only |
| ObjectType | **BACnetObjectType***** | **DWord**, Long | Read Only* |
| ProfileName | CharacterString | **String** | Read Only |

*May not be made writable.

**Object Identifier may be viewed as a packed DWord value (high 10 bits are the object type and low 22 bits are the object instance) or as individual tags for object type and instance using optional sub-property fields.

***Properties with enumerated BACnet Data Types are expressed as integer values. For standard interpretations, refer to **Enumerated Data Types**.

**Note:** Bit string types may be viewed as a packed Word value (actual number of meaningful bits will depend on specific property), or as individual bits using optional sub-property fields

### Priority Array Elements

Priority Array elements may be "NULL" or the numerical command value currently in effect. The array element index may range from 1 to 16, inclusive.

## Life Safety Point

The following table describes the object's supported properties. The Access column specifies the default access permission for tags. To accommodate non-standard implementations of BACnet, tags may be given Read/Write access unless noted otherwise. The COV column specifies whether the driver considers the property to have implicit Change Of Value (COV) notification capability; that is, whether the BACnet specification requires the property to support COV. For some properties, COV support depends on implementation. The "COV" modifier must be added to the tag's address for use. For more information, refer to **COV Settings**.

**Note:** The length of the array property will be specified by *[m]*, where m is the number of supported elements (according to the BACnet specification). The BACnet array properties that do not have a length specified by the BACnet standard will be designated by *[N]*. This means the length of the property array depends on the BACnet device. For more information, refer to **Addressing Examples**.

**See Also:** **Address Descriptions**

| Property Mnemonic | BACnet Data Type | OPC Data Type | Access | COV |
|---|---|---|---|---|
| AcceptedModes | List of **BACnetLifeSafetyMode** | **DWord**, Long | Read Only* | No |
| AckedTransitions***<br><br>.ToFault<br>.ToNormal<br>.ToOffNormal | BACnetEventTansitionBits | **Word**, Short<br><br>Boolean<br>Boolean<br>Boolean | Read Only | No |
| AlarmValues | List of **BACnetLifeSafetyState** | **DWord**, Long | Read Only* | No |
| Description | CharacterString | **String** | Read Only | No |
| DeviceType | CharacterString | **String** | Read Only | No |
| DirectReading | REAL | **Float** | Read Only | No |
| EventEnable***<br><br>.ToFault<br>.ToNormal<br>.ToOffNormal | BACnetEventTransitionBits | **Word**, Short<br><br>Boolean<br>Boolean<br>Boolean | Read/Write | No |
| EventState | **BACnetEventState**** | **DWord**, Long | Read Only | No |
| EventTimeStamps[3] | Array of TimeStamp | **String** | Read Only* | No |
| FaultValues | List of **BACnetLifeSafetyState** | **DWord**, Long | Read Only* | No |
| LifeSafetyAlarmValues | List of **BACnetLifeSafetyState** | **DWord**, Long | Read Only* | No |
| MaintenanceRequired | **BACnetMaintenance**** | **DWord**, Long | Read/Write | No |
| Mode | **BACnetLifeSafetyMode**** | **DWord**, Long | Read/Write | No |
| NotificationClass | Unsigned | **DWord**, Long | Read Only | No |
| NotifyType | **BACnetNotifyType**** | **DWord**, Long | Read Only | No |
| ObjectIdentifier**<br><br>.ObjectInstance<br>.ObjectType | BACnetObjectIdentifier | **DWord**, Long<br><br>**DWord**, Long<br>**DWord**, Long | Read Only* | No |
| ObjectName | CharacterString | **String** | Read Only | No |
| ObjectType | **BACnetObjectType**** | **DWord**, Long | Read Only* | No |
| OperationExpected | **BACnetLifeSafetyOperation**** | **DWord**, Long | Read Only | No |
| OutOfService | Boolean | **Boolean** | Read/Write | No |
| PresentValue | **BACnetLifeSafetyState**** | **DWord**, Long | Read/Write | Yes |
| ProfileName | CharacterString | **String** | Read Only | No |
| Reliability | **BACnetReliability**** | **DWord**, Long | Read Only | No |
| Setting | Unsigned | **Byte**, Char | Read/Write | No |
| Silenced | **BACnetSilencedState**** | **DWord**, Long | Read Only | No |
| StatusFlags***<br><br>.Fault<br>.InAlarm | BACnetStatusFlags | **Word**, Short<br><br>Boolean<br>Boolean | Read Only | Yes |

| | | | | |
|---|---|---|---|---|
| .OutOfService<br>.Overridden | | Boolean<br>Boolean | | |
| TimeDelay | Unsigned | **DWord**, Long | Read/Write | No |
| TrackingValue | **BACnetLifeSafetyState**** | **DWord**, Long | Read Only | No |
| Units | **BACnetEngineeringUnits**** | **DWord**, Long | Read Only | No |

*May not be made writable.

**Object Identifier may be viewed as a packed DWord value (high 10 bits are the object type and low 22 bits are the object instance) or as individual tags for object type and instance using optional sub-property fields.

***Bit string types may be viewed as a packed Word value, whose actual number of meaningful bits will depend on specific property. They may also be viewed as individual bits using optional sub-property fields.

****Properties with enumerated BACnet Data Types are expressed as integer values. For standard interpretations, refer to **Enumerated Data Types**.

### Priority Array Elements

Priority Array elements may be "NULL" or the numerical command value currently in effect. The array element index may range from 1 to 16, inclusive.

## Life Safety Zone

The following table describes the object's supported properties. The Access column specifies the default access permission for tags. To accommodate non-standard implementations of BACnet, tags may be given Read/Write access unless noted otherwise. The COV column specifies whether the driver considers the property to have implicit Change Of Value (COV) notification capability; that is, whether the BACnet specification requires the property to support COV. For some properties, COV support depends on implementation. The "COV" modifier must be added to the tag's address for use. For more information, refer to **COV Settings**.

**Note:** The length of the array property will be specified by *[m]*, where m is the number of supported elements (according to the BACnet specification). The BACnet array properties that do not have a length specified by the BACnet standard will be designated by *[N]*. This means the length of the property array depends on the BACnet device. For more information, refer to **Addressing Examples**.

**See Also:** **Address Descriptions**

| Property Mnemonic | BACnet Data Type | OPC Data Type | Access | COV |
|---|---|---|---|---|
| AcceptedModes | List of **BACnetLifeSafetyMode** | **DWord**, Long | Read Only* | No |
| AckedTransitions***<br><br>.ToFault<br>.ToNormal<br>.ToOffNormal | BACnetEventTransitionBits | **Word**, Short<br><br>Boolean<br>Boolean<br>Boolean | Read Only | No |
| AlarmValues | List of **BACnetLifeSafetyState** | **DWord**, Long | Read Only* | No |
| Description | CharacterString | String | Read Only | No |
| DeviceType | CharacterString | String | Read Only | No |
| EventEnable***<br><br>.ToFault<br>.ToNormal<br>.ToOffNormal | BACnetEventTransitionBits | **Word**, Short<br><br>Boolean<br>Boolean<br>Boolean | Read/Write | No |
| EventState | **BACnetEventState**** | **DWord**, Long | Read Only | No |
| EventTimeStamps[3] | Array of TimeStamp | **String** | Read Only* | No |
| FaultValues | List of **BACnetLifeSafetyState** | **DWord**, Long | Read Only* | No |
| LifeSafetyAlarmValues | List of **BACnetLifeSafetyState** | **DWord**, Long | Read Only* | No |
| MaintenanceRequired | Boolean | Boolean | Read/Write | No |
| Mode | **BACnetLifeSafetyMode**** | **DWord**, Long | Read/Write | No |
| NotificationClass | Unsigned | **DWord**, Long | Read Only | No |
| NotifyType | **BACnetNotifyType**** | **DWord**, Long | Read Only | No |
| ObjectIdentifier**<br><br>.ObjectInstance | BACnetObjectIdentier | **DWord**, Long<br><br>**DWord**, Long | Read Only* | No |

| | | | | |
|---|---|---|---|---|
| .ObjectType | | **DWord**, Long | | |
| ObjectName | CharacterString | String | Read Only | No |
| ObjectType | **BACnetObjectType**** | **DWord**, Long | Read Only* | No |
| OperationExpected | **BACnetLifeSafetyOperation**** | **DWord**, Long | Read Only | No |
| OutOfService | Boolean | Boolean | Read/Write | No |
| PresentValue | **BACnetLifeSafetyState**** | **DWord**, Long | Read/Write | Yes |
| ProfileName | CharacterString | String | Read Only | No |
| Reliability | **BACnetReliability**** | **DWord**, Long | Read Only | No |
| Silenced | **BACnetSilencedState**** | **DWord**, Long | Read Only | No |
| StatusFlags***<br><br>.Fault<br>.InAlarm<br>.OutOfService<br>.Overridden | BACnetStatusFlags | **Word**, Short<br><br>Boolean<br>Boolean<br>Boolean<br>Boolean | Read Only | Yes |
| TimeDelay | Unsigned | **DWord**, Long | Read/Write | No |
| TrackingValue | **BACnetLifeSafetyState**** | **DWord**, Long | Read Only | No |

*May not be made writable.

**Object Identifier may be viewed as a packed DWord value (high 10 bits are the object type and low 22 bits are the object instance) or as individual tags for object type and instance using optional sub-property fields.

***Bit string types may be viewed as a packed Word value, whose actual number of meaningful bits will depend on specific property. They may also be viewed as individual bits using optional sub-property fields.

****Properties with enumerated BACnet Data Types are expressed as integer values. For standard interpretations, refer to **Enumerated Data Types**.

### Priority Array Elements

Priority Array elements may be "NULL" or the numerical command value currently in effect. The array element index may range from 1 to 16, inclusive.

## Loop

The following table describes the object's supported properties. The Access column specifies the default access permission for tags. To accommodate non-standard implementations of BACnet, tags may be given Read/Write access unless noted otherwise. The COV column specifies whether the driver considers the property to have implicit Change Of Value (COV) notification capability; that is, whether the BACnet specification requires the property to support COV. For some properties, COV support depends on implementation. The "COV" modifier must be added to the tag's address for use. For more information, refer to **COV Settings**.

**Note:** The length of the array property will be specified by *[m]*, where m is the number of supported elements (according to the BACnet specification). The BACnet array properties that do not have a length specified by the BACnet standard will be designated by *[N]*. This means the length of the property array depends on the BACnet device. For more information, refer to **Addressing Examples**.

**See Also:** **Address Descriptions**

| Property Mnemonic | BACnet Data Type | OPC Data Type | Access | COV |
|---|---|---|---|---|
| AckedTransitions***<br><br>.ToFault<br>.ToNormal<br>.ToOffNormal | BACnetEventTransitionBits | **Word**, Short<br><br>Boolean<br>Boolean<br>Boolean | Read Only | No |
| Action | **BACnetAction**** | **DWord**, Long | Read Only | No |
| Bias | REAL | **Float** | Read/Write | No |
| ControlledVariableUnits | **BACnetEngineeringUnits**** | **DWord**, Long | Read Only | No |
| ControlledVariableValue | REAL | **Float** | Read Only | Yes |
| COVIncrement | REAL | **Float** | Read/Write | No |
| DerivativeConstant | REAL | **Float** | Read/Write | No |
| DerivativeConstantUnits | **BACnetEngineeringUnits**** | **DWord**, Long | Read Only | No |

| | | | | |
|---|---|---|---|---|
| Description | CharacterString | **String** | Read Only | No |
| ErrorLimit | REAL | **Float** | Read/Write | No |
| EventEnable***<br><br>.ToFault<br>.ToNormal<br>.ToOffNormal | BACnetEventTransitionBits | **Word**, Short<br><br>Boolean<br>Boolean<br>Boolean | Read/Write | No |
| EventState | **BACnetEventState**** | **DWord**, Long | Read Only | No |
| EventTimeStamps[3] | Array of TimeStamp | **String** | Read Only* | No |
| IntegralConstant | REAL | **Float** | Read/Write | No |
| IntegralConstantUnits | **BACnetEngineeringUnits**** | **DWord**, Long | Read Only | No |
| MaximumOutput | REAL | **Float** | Read Only | No |
| MinimumOutput | REAL | **Float** | Read Only | No |
| NotificationClass | Unsigned | **DWord**, Long | Read Only | No |
| NotifyType | BACnetNotifyType | **DWord**, Long | Read Only | No |
| ObjectIdentifier**<br><br>.ObjectInstance<br>.ObjectType | BACnetObjectIdentifier | **DWord**, Long<br><br>**DWord**, Long<br>**DWord**, Long | Read Only* | No |
| ObjectName | CharacterString | **String** | Read Only | No |
| ObjectType | **BACnetObjectType**** | **DWord**, Long | Read Only* | No |
| OutOfService | Boolean | **Boolean** | Read/Write | No |
| OutputUnits | **BACnetEngineeringUnits**** | **DWord**, Long | Read Only | No |
| PresentValue | REAL | **Float** | Read Only | Yes |
| PriorityForWriting | Unsigned | **DWord**, Long | Read/Write | No |
| ProfileName | CharacterString | **String** | Read Only | No |
| ProportionalConstant | REAL | **Float** | Read/Write | No |
| ProportionalConstantUnits | **BACnetEngineeringUnits**** | **DWord**, Long | Read Only | No |
| Reliability | **BACnetReliability**** | **DWord**, Long | Read Only | No |
| Setpoint | REAL | **Float** | Read/Write | Yes |
| StatusFlags***<br><br>.Fault<br>.InAlarm<br>.OutOfService<br>.Overridden | BACnetStatusFlags | **Word**, Short<br><br>Boolean<br>Boolean<br>Boolean<br>Boolean | Read Only | Yes |
| TimeDelay | Unsigned | **DWord**, Long | Read/Write | No |
| UpdateInterval | Unsigned | **DWord**, Long | Read/Write | No |

*May not be made writable.

**Object Identifier may be viewed as a packed DWord value (high 10 bits are the object type and low 22 bits are the object instance) or as individual tags for object type and instance using optional sub-property fields.

***Bit string types may be viewed as a packed Word value, whose actual number of meaningful bits will depend on specific property. They may also be viewed as individual bits using optional sub-property fields.

****Properties with enumerated BACnet Data Types are expressed as integer values. For standard interpretations, refer to **Enumerated Data Types**.

### Priority Array Elements

Priority Array elements may be "NULL" or the numerical command value currently in effect. The array element index may range from 1 to 16, inclusive.

## Multi-State Input

The following table describes the object's supported properties. The Access column specifies the default access permission for tags. To accommodate non-standard implementations of BACnet, tags may be given Read/Write access unless noted otherwise. The COV column specifies whether the driver considers the property to have implicit Change Of Value (COV) notification capability; that is, whether the BACnet specification requires the property to support COV. For some properties, COV support depends on implementation. The "COV" modifier must be added to the tag's address for use. For more information, refer to **COV Settings**.

**Note:** The length of the array property will be specified by *[m]*, where m is the number of supported elements (according to the BACnet specification). The BACnet array properties that do not have a length specified by the BACnet standard will be designated by *[N]*. This means the length of the property array depends on the BACnet device. For more information, refer to **Addressing Examples**.

**See Also:** **Address Descriptions**

| Property Mnemonic | BACnet Data Type | OPC Data Type | Access | COV |
|---|---|---|---|---|
| AckedTransitions***<br><br>.ToFault<br>.ToNormal<br>.ToOffNormal | BACnetEventTransitionBits | **Word**, Short<br><br>Boolean<br>Boolean<br>Boolean | Read Only | No |
| AlarmValues | List of Unsigned | **DWord**, Long | Read Only* | No |
| Description | CharacterString | **String** | Read Only | No |
| DeviceType | CharacterString | **String** | Read Only | No |
| EventEnable***<br><br>.ToFault<br>.ToNormal<br>.ToOffNormal | BACnetEventTransitionBits | **Word**, Short<br><br>Boolean<br>Boolean<br>Boolean | Read/Write | No |
| EventState | **BACnetEventState**** | **DWord**, Long | Read Only | No |
| EventTimeStamps[3] | Array of TimeStamp | **String** | Read Only* | No |
| FaultValues | List of Unsigned | **DWord**, Long | Read Only* | No |
| NotificationClass | Unsigned | **DWord**, Long | Read Only | No |
| NotifyType | **BACnetNotifyType**** | **DWord**, Long | Read Only | No |
| NumberOfStates | Unsigned | **DWord**, Long | Read Only | No |
| ObjectIdentifier**<br><br>.ObjectInstance<br>.ObjectType | BACnetObjectIdentifier | **DWord**, Long<br><br>**DWord**, Long<br>**DWord**, Long | Read Only* | No |
| ObjectName | CharacterString | **String** | Read Only | No |
| ObjectType | **BACnetObjectType**** | **DWord**, Long | Read Only* | No |
| OutOfService | Boolean | **Boolean** | Read/Write | No |
| PresentValue | Unsigned | **DWord**, Long | Read/Write | Yes |
| ProfileName | CharacterString | **String** | Read Only | No |
| Reliability | **BACnetReliability**** | **DWord**, Long | Read Only | No |
| StateText[N] | Array of CharacterString | **String** | Read/Write | No |
| StatusFlags***<br><br>.Fault<br>.InAlarm<br>.OutOfService<br>.Overridden | BACnetStatusFlags | **Word**, Short<br><br>Boolean<br>Boolean<br>Boolean<br>Boolean | Read Only | Yes |
| TimeDelay | Unsigned | **DWord**, Long | Read/Write | No |

*May not be made writable.

**Object Identifier may be viewed as a packed DWord value (high 10 bits are the object type and low 22 bits are the object instance) or as individual tags for object type and instance using optional sub-property fields.

***Bit string types may be viewed as a packed Word value, whose actual number of meaningful bits will depend on specific property. They may also be viewed as individual bits using optional sub-property fields.

****Properties with enumerated BACnet Data Types are expressed as integer values. For standard interpretations, refer to **Enumerated Data Types**.

### Priority Array Elements

Priority Array elements may be "NULL" or the numerical command value currently in effect. The array element index may range from 1 to 16, inclusive.

## Multi-State Output

The following table describes the object's supported properties. The Access column specifies the default access permission for tags. To accommodate non-standard implementations of BACnet, tags may be given Read/Write access unless noted otherwise. The COV column specifies whether the driver considers the property to have implicit Change Of Value (COV) notification capability; that is, whether the BACnet specification requires the property to support COV. For some properties, COV support depends on implementation. The "COV" modifier must be added to the tag's address for use. For more information, refer to **COV Settings**.

**Note:** The length of the array property will be specified by *[m]*, where m is the number of supported elements (according to the BACnet specification). The BACnet array properties that do not have a length specified by the BACnet standard will be designated by *[N]*. This means the length of the property array depends on the BACnet device. For more information, refer to **Addressing Examples**.

**See Also:** **Address Descriptions**

| Property Mnemonic | BACnet Data Type | OPC Data Type | Access | COV |
|---|---|---|---|---|
| AckedTransitions***<br><br>.ToFault<br>.ToNormal<br>.ToOffNormal | BACnetEventTransitionBits | **Word**, Short<br><br>Boolean<br>Boolean<br>Boolean | Read Only | No |
| Description | CharacterString | **String** | Read Only | No |
| DeviceType | CharacterString | **String** | Read Only | No |
| EventEnable***<br><br>.ToFault<br>.ToNormal<br>.ToOffNormal | BACEventTransitionBits | **Word**, Short<br><br>Boolean<br>Boolean<br>Boolean | Read/Write | No |
| EventState | **BACnetEventState**** | **DWord**, Long | Read Only | No |
| EventTimeStamps[3] | Array of TimeStamp | **String** | Read Only* | No |
| FeedbackValue | Unsigned | **DWord**, Long | Read/Write | No |
| NotificationClass | Unsigned | **DWord**, Long | Read Only | No |
| NotifyType | **BACnetNotifyType**** | **DWord**, Long | Read Only | No |
| NumberOfStates | Unsigned | **DWord**, Long | Read Only | No |
| ObjectIdentifier**<br><br>.ObjectInstance<br>.ObjectType | BACnetObjectIdentifier | **DWord**, Long<br><br>**DWord**, Long<br>**DWord**, Long | Read Only* | No |
| ObjectName | CharacterString | **String** | Read Only | No |
| ObjectType | **BACnetObjectType**** | **DWord**, Long | Read Only* | No |
| OutOfService | Boolean | **Boolean** | Read/Write | No |
| PresentValue | Unsigned | **DWord**, Long | Read/Write | Yes |
| PriorityArray[16] | Array of BACnetPriorityArray | **String** | Read Only* | No |
| ProfileName | CharacterString | **String** | Read Only | No |
| Reliability | **BACnetReliability**** | **DWord**, Long | Read Only | No |
| RelinquishDefault | Unsigned | **DWord**, Long | Read/Write | No |
| StateText[N] | Array of CharacterString | **String** | Read/Write | No |
| StatusFlags***<br><br>.Fault<br>.InAlarm<br>.OutOfService<br>.Overridden | BACnetStatusFlags | **Word**, Short<br><br>Boolean<br>Boolean<br>Boolean<br>Boolean | Read Only | Yes |
| TimeDelay | Unsigned | **DWord**, Long | Read/Write | No |

*May not be made writable.

**Object Identifier may be viewed as a packed DWord value (high 10 bits are the object type and low 22 bits are the object instance) or as individual tags for object type and instance using optional sub-property fields.

***Bit string types may be viewed as a packed Word value, whose actual number of meaningful bits will depend on specific property. They may also be viewed as individual bits using optional sub-property fields.

****Properties with enumerated BACnet Data Types are expressed as integer values. For standard interpretations, refer to **Enumerated Data Types**.

### Priority Array Elements

Priority Array elements may be "NULL" or the numerical command value currently in effect. The array element index may range from 1 to 16, inclusive.

## Multi-State Value

The following table describes the object's supported properties. The Access column specifies the default access permission for tags. To accommodate non-standard implementations of BACnet, tags may be given Read/Write access unless noted otherwise. The COV column specifies whether the driver considers the property to have implicit Change Of Value (COV) notification capability; that is, whether the BACnet specification requires the property to support COV. For some properties, COV support depends on implementation. The "COV" modifier must be added to the tag's address for use. For more information, refer to **COV Settings**.

**Note:** The length of the array property will be specified by *[m]*, where m is the number of supported elements (according to the BACnet specification). The BACnet array properties that do not have a length specified by the BACnet standard will be designated by *[N]*. This means the length of the property array depends on the BACnet device. For more information, refer to **Addressing Examples**.

**See Also:** **Address Descriptions**

| Property Mnemonic | BACnet Data Type | OPC Data Type | Access | COV |
|---|---|---|---|---|
| AckedTransitions***<br><br>.ToFault<br>.ToNormal<br>.ToOffNormal | BACnetEventTransitionBits | **Word**, Short<br><br>Boolean<br>Boolean<br>Boolean | Read Only | No |
| AlarmValues | List of Unsigned | **DWord**, Long | Read Only* | No |
| Description | CharacterString | **String** | Read Only | No |
| EventEnable***<br><br>.ToFault<br>.ToNormal<br>.ToOffNormal | BACnetEventTransitionBits | **Word**, Short<br><br>Boolean<br>Boolean<br>Boolean | Read/Write | No |
| EventState | **BACnetEventState**** | **DWord**, Long | Read Only | No |
| EventTimeStamps[3] | Array of TimeStamp | **String** | Read Only* | No |
| FaultValues | List of Unsigned | **DWord**, Long | Read Only* | No |
| NotificationClass | Unsigned | **DWord**, Long | Read Only | No |
| NotifyType | **BACnetNotifyType**** | **DWord**, Long | Read Only | No |
| NumberOfStates | Unsigned | **DWord**, Long | Read Only | No |
| ObjectIdentifier**<br><br>.ObjectInstance<br>.ObjectType | BACnetObjectIdentifier | **DWord**, Long<br><br>**DWord**, Long<br>**DWord**, Long | Read Only* | No |
| ObjectName | CharacterString | **String** | Read Only | No |
| ObjectType | **BACnetObjectType**** | **DWord**, Long | Read Only* | No |
| OutOfService | Boolean | **Boolean** | Read/Write | No |
| PresentValue | Unsigned | **DWord**, Long | Read/Write | Yes |
| PriorityArray[16] | Array of BACnetPriorityArray | **String** | Read Only* | No |
| ProfileName | CharacterString | **String** | Read Only | No |
| Reliability | **BACnetReliability**** | **DWord**, Long | Read Only | No |
| RelinquishDefault | Unsigned | **DWord**, Long | Read/Write | No |
| StateText[N] | Array of CharacterString | **String** | Read/Write | No |
| StatusFlags***<br><br>.Fault<br>.InAlarm | BACnetStatusFlags | **Word**, Short<br><br>Boolean<br>Boolean | Read Only | Yes |

| | | | | |
|---|---|---|---|---|
| .OutOfService<br>.Overridden | | Boolean<br>Boolean | | |
| TimeDelay | Unsigned | **DWord**, Long | Read/Write | No |

*May not be made writable.

**Object Identifier may be viewed as a packed DWord value (high 10 bits are the object type and low 22 bits are the object instance) or as individual tags for object type and instance using optional sub-property fields.

***Bit string types may be viewed as a packed Word value, whose actual number of meaningful bits will depend on specific property. They may also be viewed as individual bits using optional sub-property fields.

****Properties with enumerated BACnet Data Types are expressed as integer values. For standard interpretations, refer to **Enumerated Data Types**.

### Priority Array Elements

Priority Array elements may be "NULL" or the numerical command value currently in effect. The array element index may range from 1 to 16, inclusive.

## Notification Class

The following table describes the object's supported properties. The Access column specifies the default access permission for tags. To accommodate non-standard implementations of BACnet, tags may be given Read/Write access unless noted otherwise.

**Note 1:** The length of the array property will be specified by *[m]*, where m is the number of supported elements (according to the BACnet specification). The BACnet array properties that do not have a length specified by the BACnet standard will be designated by *[N]*. This means the length of the property array depends on the BACnet device. For more information, refer to **Addressing Examples**.

**Note 2:** Implicit Change of Value (COV) notifications are not supported for this object. For more information on devices with explicit COV support, refer to **COV Notifications**.

**See Also:** **Address Descriptions**

| Property Mnemonic | BACnet Data Type | OPC Data Type | Access |
|---|---|---|---|
| AckRequired***<br><br>.ToFault<br>.ToNormal<br>.ToOffNormal | BACnetEvnetTransitionBits | **Word**, Short<br><br>Boolean<br>Boolean<br>Boolean | Read Only |
| Description | CharacterString | String | Read Only |
| NotificationClass | Unsigned | **DWord**, Long | Read Only |
| ObjectIdentifier**<br><br>.ObjectInstance<br>.ObjectType | BACnetObjectIdentifier | **DWord**, Long<br><br>**DWord**, Long<br>**DWord**, Long | Read Only* |
| ObjectName | CharacterString | String | Read Only |
| ObjectType | **BACnetObjectType****** | **DWord**, Long | Read Only* |
| Priority[3] | Array of Unsigned | **DWord**, Long | Read Only* |
| ProfileName | CharacterString | String | Read Only |

*May not be made writable.

**Object Identifier may be viewed as a packed DWord value (high 10 bits are the object type and low 22 bits are the object instance) or as individual tags for object type and instance using optional sub-property fields.

***Bit string types may be viewed as a packed Word value, whose actual number of meaningful bits will depend on specific property. They may also be viewed as individual bits using optional sub-property fields.

****Properties with enumerated BACnet Data Types are expressed as integer values. For standard interpretations, refer to **Enumerated Data Types**.

### Priority Array Elements

Priority Array elements may be "NULL" or the numerical command value currently in effect. The array element index may range from 1 to 16, inclusive.

## Program

The following table describes the object's supported properties. The Access column specifies the default access permission for tags. To accommodate non-standard implementations of BACnet, tags may be given Read/Write access unless noted otherwise.

**Note 1:** The length of the array property will be specified by *[m]*, where m is the number of supported elements (according to the BACnet specification). The BACnet array properties that do not have a length specified by the BACnet standard will be designated by *[N]*. This means the length of the property array depends on the BACnet device. For more information, refer to **Addressing Examples**.

**Note 2:** Implicit Change of Value (COV) notifications are not supported for this object. For more information on devices with explicit COV support, refer to **COV Notifications**.

**See Also:** **Address Descriptions**

| Property Mnemonic | BACnet Data Type | OPC Data Type | Access |
|---|---|---|---|
| Description | CharacterString | **String** | Read Only |
| DescriptionOfHalt | CharacterString | **String** | Read Only |
| InstanceOf | CharacterString | **String** | Read Only |
| ObjectIdentifier**<br><br>.ObjectInstance<br>.ObjectType | BACnetObjectIdentifier | **DWord**, Long<br><br>**DWord**, Long<br>**DWord**, Long | Read Only* |
| ObjectName | CharacterString | **String** | Read Only |
| ObjectType | **BACnetObjectType**** | **DWord**, Long | Read Only* |
| OutOfService | Boolean | **Boolean** | Read/Write |
| ProfileName | CharacterString | **String** | Read Only |
| ProgramChange | **BACnetProgramRequest**** | **DWord**, Long | Read Only |
| ProgramLocation | CharacterString | **String** | Read Only |
| ProgramState | **BACnetProgramState**** | **DWord**, Long | Read Only |
| ReasonForHalt | **BACnetProgramError***** | **DWord**, Long | Read Only |
| Reliability | **BACnetReliability**** | **DWord**, Long | Read Only |
| StatusFlags***<br><br>.Fault<br>.InAlarm<br>.OutOfService<br>.Overridden | BACnetStatusFlags | **Word**, Short<br><br>Boolean<br>Boolean<br>Boolean<br>Boolean | Read Only |

*May not be made writable.

**Object Identifier may be viewed as a packed DWord value (high 10 bits are the object type and low 22 bits are the object instance) or as individual tags for object type and instance using optional sub-property fields.

***Bit string types may be viewed as a packed Word value, whose actual number of meaningful bits will depend on specific property. They may also be viewed as individual bits using optional sub-property fields.

****Properties with enumerated BACnet Data Types are expressed as integer values. For standard interpretations, refer to **Enumerated Data Types**.

### Priority Array Elements

Priority Array elements may be "NULL" or the numerical command value currently in effect. The array element index may range from 1 to 16, inclusive.

## Schedule

The following table describes the object's supported properties. The Access column specifies the default access permission for tags. To accommodate non-standard implementations of BACnet, tags may be given Read/Write access unless noted otherwise. The COV column specifies whether the driver considers the property to have implicit Change Of Value (COV) notification capability; that is, whether the BACnet specification requires the

property to support COV. For some properties, COV support depends on implementation. The "COV" modifier must be added to the tag's address for use. For more information, refer to **COV Settings**.

**Note:** The length of the array property will be specified by *[m]*, where m is the number of supported elements (according to the BACnet specification). The BACnet array properties that do not have a length specified by the BACnet standard will be designated by *[N]*. This means the length of the property array depends on the BACnet device. For more information, refer to **Addressing Examples**.

**See Also:** **Address Descriptions**

| Property Mnemonic | BACnet Data Type | OPC Data Type | Access | COV |
|---|---|---|---|---|
| Description | CharacterString | **String** | Read Only | No |
| EffectivePeriod | BACnetDateRange | **String** | Read Only* | No |
| ObjectIdentifier**<br><br>.ObjectInstance<br>.ObjectType | BACnetObjectIdentier | **DWord**, Long<br><br>**DWord**, Long<br>**DWord**, Long | Read Only* | No |
| ObjectName | CharacterString | **String** | Read Only | No |
| ObjectType | **BACnetObjectType**** | **DWord**, Long | Read Only* | No |
| PresentValue | Any | **Float** | Read/Write | No |
| PriorityForWriting | Unsigned | **DWord**, Long | Read/Write | No |
| ProfileName | CharacterString | **String** | Read Only | No |
| ScheduleDefault | Real | Float | Read Only | No |
| StatusFlags<br><br>.InAlarm<br>.Fault<br>.Overridden<br>.OutofService | BACnetStatusFlags | **Word**, Short<br><br>Boolean<br>Boolean<br>Boolean<br>Boolean | Read Only* | Yes |
| Reliability | **BACnetReliability** | **DWord**, Long | Read Only* | No |
| OutofService | Boolean | **Boolean** | Read/Write | No |

*May not be made writable.

**Object Identifier may be viewed as a packed DWord value (high 10 bits are the object type and low 22 bits are the object instance) or as individual tags for object type and instance using optional sub-property fields.

***Properties with enumerated BACnet Data Types are expressed as integer values. For standard interpretations, refer to **Enumerated Data Types**.

**Note:** Bit string types may be viewed as a packed Word value, whose actual number of meaningful bits will depend on specific property. They may also be viewed as individual bits using optional sub-property fields.

### Priority Array Elements

Priority Array elements may be "NULL" or the numerical command value currently in effect. The array element index may range from 1 to 16, inclusive.

## Trend Log

The following table describes the object's supported properties. The Access column specifies the default access permission for tags. To accommodate non-standard implementations of BACnet, tags may be given Read/Write access unless noted otherwise.

**Note 1:** The length of the array property will be specified by *[m]*, where m is the number of supported elements (according to the BACnet specification). The BACnet array properties that do not have a length specified by the BACnet standard will be designated by *[N]*. This means the length of the property array depends on the BACnet device. For more information, refer to **Addressing Examples**.

**Note 2:** Implicit Change of Value (COV) notifications are not supported for this object. For more information on devices with explicit COV support, refer to **COV Notifications**.

**See Also:** **Address Descriptions**

| Property Mnemonic | BACnet Data Type | OPC Data Type | Access |
|---|---|---|---|

| | | | |
|---|---|---|---|
| AckedTransitions***<br><br>.ToFault<br>.ToNormal<br>.ToOffNormal | BACnetEventTansitionBits | **Word**, Short<br><br>Boolean<br>Boolean<br>Boolean | Read Only |
| BufferSize | Unsigned | **DWord**, Long | Read Only |
| COVResubscriptionInterval | Unsigned | **DWord**, Long | Read/Write |
| CurrentNotifyTime *2 | BACnetDateTime | **String** | Read Only* |
| Description | CharacterString | **String** | Read Only |
| EventEnable***<br><br>.ToFault<br>.ToNormal<br>.ToOffNormal | BACnetEventTansitionBits | **Word**, Short<br><br>Boolean<br>Boolean<br>Boolean | Read/Write |
| EventState | **BACnetEventState**** | **DWord**, Long | Read Only |
| EventTimeStamps[3] | Array of TimeStamp | **String** | Read Only* |
| LastNotifyRecord *1 | Unsigned | **DWord**, Long | Read Only |
| LogEnable | Boolean | **Boolean** | Read/Write |
| LogInterval | Unsigned | **DWord**, Long | Read/Write |
| NotificationClass | Unsigned | **DWord**, Long | Read Only |
| NotificationThreshold | Unsigned | **DWord**, Long | Read Only |
| NotifyType | **BACnetNotifyType**** | **DWord**, Long | Read Only |
| ObjectIdentifier**<br><br>.ObjectInstance<br>.ObjectType | BACnetObjectIdentifier | **DWord**, Long<br><br>**DWord**, Long<br>**DWord**, Long | Read Only* |
| ObjectName | CharacterString | **String** | Read Only |
| ObjectType | **BACnetObjectType**** | **DWord**, Long | Read Only* |
| PreviousNotifyTime *2 | BACnetDateTime | **String** | Read Only* |
| ProfileName | CharacterString | **String** | Read Only |
| RecordCount | Unsigned | **DWord**, Long | Read/Write |
| RecordsSinceNotification | Unsigned | **DWord**, Long | Read Only |
| StartTime | BACnetDateTime | **String** | Read Only* |
| StopTime | BACnetDateTime | **String** | Read Only* |
| StopWhenFull | Boolean | **Boolean** | Read/Write |
| TotalRecordCount | Unsigned | **DWord**, Long | Read Only |

*May not be made writable.

**Object Identifier may be viewed as a packed DWord value (high 10 bits are the object type and low 22 bits are the object instance) or as individual tags for object type and instance using optional sub-property fields.

***Bit string types may be viewed as a packed Word value, whose actual number of meaningful bits will depend on specific property. They may also be viewed as individual bits using optional sub-property fields.

****Properties with enumerated BACnet Data Types are expressed as integer values. For standard interpretations, refer to **Enumerated Data Types**.

*1: The LastNotifyRecord property replaced the PreviousNotifyTime and CurrentNotifyTime properties in the BACnet 2004 Specification. Devices that support LastNotifyRecord may not support the CurrentNotifyTime and LastNotifyTime properties.

*2: This has been removed from the BACnet 2004 Specification. Support is available for legacy devices only.

### Priority Array Elements

Priority Array elements may be "NULL" or the numerical command value currently in effect. The array element index may range from 1 to 16, inclusive.

## Error Descriptions

Click on the link for a description of abort and reject reasons, as well as error classes and codes.



### Error Messages

The following error/warning messages may be generated. They are listed here in alphabetical order. Click on the link for a description of the message.



## Abort Reasons

The following are standard abort reason codes as defined in the BACnet specification.

| Code | Description |
|---|---|
| 0 | Other |
| 1 | Buffer overflow |
| 2 | Invalid APDU in this state |
| 3 | Preempted by higher priority task |
| 4 | Segmentation not supported |

## Reject Reasons

The following are standard reject reason codes defined in the BACnet specification.

| Code | Description |
|---|---|
| 0 | Other |
| 1 | Buffer overflow |
| 2 | Inconsistent parameters |
| 3 | Invalid parameter data type |
| 4 | Invalid tag |
| 5 | Missing required parameter |
| 6 | Parameter out of range |
| 7 | Too many arguments |
| 8 | Unidentified enumeration |
| 9 | Unrecognized service |

## Error Classes and Codes

The following are standard error classes and codes defined in the BACnet specification.

### BACnet Error Classes

| Class | Description |
|---|---|
| 0 | Device |
| 1 | Object |
| 2 | Property |
| 3 | Resources |
| 4 | Security |
| 5 | Services |
| 6 | Virtual Terminal |
| 7 | Communication |

### BACnet Error Codes

| Code | Description |
|---|---|
| 0 | Other |
| 1 | Authentication Failed |
| 2 | Configuration In Progress |
| 3 | Device Busy |
| 4 | Dynamic Creation Not Supported |
| 5 | File Access Denied |
| 6 | Incompatible Security Levels |
| 7 | Inconsistent Parameters |
| 8 | Inconsistent Selection Criteria |
| 9 | Invalid Data Type |
| 10 | Invalid File Access Method |
| 11 | Invalid File Start Position |
| 12 | Invalid Operator Name |
| 13 | Invalid Parameter Data Type |
| 14 | Invalid Time Stamp |
| 15 | Key Generation Error |
| 16 | Missing Required Parameter |
| 17 | NoObjects Of Specified Type |
| 18 | No Space For Object |
| 19 | No Space To Add List Element |
| 20 | No Space To Write Property |
| 21 | No VT Sessions Available |
| 22 | Property Is Not A List |
| 23 | Object Deletion Not Permitted |
| 24 | Object Identifier Already Exists |
| 25 | Operational Problem |
| 26 | Password Failure |

| | |
|---|---|
| 27 | Read Access Denied |
| 28 | Security Not Supported |
| 29 | Service Request Denied |
| 30 | Timeout |
| 31 | Unknown Object |
| 32 | Unknown Property |
| 33 | --Enumeration Removed-- |
| 34 | Unknown VT Class |
| 35 | Unknown VT Session |
| 36 | Unsupported Object Type |
| 37 | Value Out Of Range |
| 38 | VT Session Already Closed |
| 39 | VT Session Termination Failure |
| 40 | Write Access Denied |
| 41 | Character Set Not Supported |
| 42 | Invalid Array Index |
| 43 | COV Subscription Failed |
| 44 | Not COV Property |
| 45 | Original Functionality Not Supported |
| 46 | Invalid Configuration Data |
| 47 | Data Type Not Supported |
| 48 | Duplicate Name |
| 49 | Duplicate Object ID |
| 50 | Property is Not an Array |
| 51 | Abort Buffer Overflow |
| 52 | Abort Invalid APDU in this State |
| 53 | Abort Preempted by Higher Priority Task |
| 54 | Abort Segmentation Not Supported |
| 55 | Abort Proprietary |
| 56 | Abort Other |
| 57 | Invalid Tag |
| 58 | Network Down |
| 59 | Reject Buffer Overflow |
| 60 | Reject Inconsistent Parameters |
| 61 | Reject Invalid Parameter Data Type |
| 62 | Reject Invalid Tag |
| 63 | Reject Missing Required Parameter |
| 64 | Reject Parameter Out of Range |
| 65 | Reject Too Many Arguments |
| 66 | Reject Undefined Enumeration |
| 67 | Reject Unrecognized Service |
| 68 | Reject Proprietary |
| 69 | Reject Other |
| 70 | Unknown Device |
| 71 | Unknown Route |
| 72 | Value Not Initialized |
| 73 | Invalid Event State |
| 74 | No Alarm Configured |
| 75 | Log Buffer Full |
| 76 | Logged Value Purged |
| 77 | No Property Specified |
| 78 | Not Configured For Triggered Logging |
| 79 | --Reserved for Future Use-- |
| 80 | Parameter Out of Range |
| 81 | --Reserved for Future Use-- |

| | |
|---|---|
| 82 | Busy |
| 83 | Communication Disabled |

## Error Messages

The following error/warning messages may be generated. They are listed here in alphabetical order. Click on the link for a description of the message.

### Error Messages



## Address '<address>' is out of range for the specified device or register

**Error Type:**
Warning

**Possible Cause:**
A tag address that has been specified statically references a location that is beyond the range of supported locations for the device.

**Solution:**
Verify the address is correct; if it is not, re-enter it in the client application.

## Connection failed - could not read max APDU length from remote device '<device>'

**Error Type:**
Serious

**Possible Cause:**
The IP address entered in the Discovery Device Properties may be incorrect.

**Solution:**
1. Verify the IP address of the device.
2. Enable Device Discovery.

**See Also:**
**Discovery**

## Connection failed - could not read protocol services supported from remote device '<device>'

**Error Type:**
Serious

**Possible Cause:**
The IP address entered in the Discovery Device Properties may be incorrect.

**Solution:**
1. Verify the IP address of the device.
2. Enable Device Discovery.

**See Also:**
**Discovery**

## Connection failed - could not read segmentation supported from remote device '<device>'

**Error Type:**
Serious

**Possible Cause:**
The IP address entered in the Discovery Device Properties may be incorrect.

**Solution:**
1. Verify the IP address of the device.
2. Enable Device Discovery.

**See Also:**
**Discovery**

## Connection failed - could not register as foreign device for discovery of remote device '<device>'

**Error Type:**
Serious

**Possible Cause:**
1. The network connection between the device and the host PC is broken.
2. The BBMD IP specified on the Foreign Device Channel Properties is not correct.
3. The BBMD and driver are not network-visible to each other.

**Solution:**
1. Verify the cabling between the PC and the PLC device.
2. Verify the IP of the BBMD.
3. Ping the BBMD from the driver's host computer. Make sure the host and BBMD have the correct default gateway IP configured and that an IP router has been configured to join the subnets.

**See Also:**
**Foreign Device**

## Connection failed - did not get I-Am from remote device '<device>'

**Error Type:**
Serious

**Possible Cause:**
1. The network connection between the device and the host PC is broken.
2. The communications parameters configured for the device and driver do not match.

**Solution:**
1. Verify the cabling between the PC and the PLC device.
2. Verify that the specified communications parameters match those of the device.

## COV subscription failed for tag '<tag>' on device '<device>' (Class: <class>, Code:

**Error Type:**
Serious

**Possible Cause:**
See the given error class and code.

**Solution:**
The device may not support COV for the given item or not have the resources to service the request at the time it was issued. Consider polling the property.

**Note:**
For a complete listing of standard BACnet error types, refer to **Error Classes and Codes**.

**See Also:**
**COV Settings**

## Data Type '<type>' is not valid for device address '<address>'

**Error Type:**
Warning

**Possible Cause:**
A tag address that has been specified statically has been assigned an invalid data type.

**Solution:**
Modify the requested data type in the client application.

## Device '<device>' is not responding

**Error Type:**
Serious

**Possible Cause:**
1. The network connection between the device and the host PC is broken.
2. The communications parameters configure for the device and driver do not match.

**Solution:**
1. Verify the cabling between the PC and the PLC device.
2. Verify that the specified communications parameters match those of the device.

## Device address '<address>' contains a syntax error

**Error Type:**
Warning

**Possible Cause:**
A tag address that has been specified statically contains one or more invalid characters.

**Solution:**

Re-enter the address in the client application.

## Device address '<address>' is not supported by model '<model name>'

**Error Type:**
Warning

**Possible Cause:**
A tag address that has been specified statically references a location that is valid for the communications protocol but not supported by the target device.

**Solution:**
1. Verify that the address is correct; if it is not, re-enter it in the client application.
2. Verify that the selected model name for the device is correct.

## Device address '<address>' is Read Only

**Error Type:**
Warning

**Possible Cause:**
A tag address that has been specified statically has a requested access mode that is not compatible with what the device supports for that address.

**Solution:**
Change the access mode in the client application.

## Error reading object list from device '<device>' (Class: <class>, Code: <code>)

**Error Type:**
Warning

**Possible Cause:**
1. Message corrupted.
2. Segmentation not supported by device.
3. Incorrect BACnet implementation.

**Solution:**
1. No corrective actions may be needed if subsequent request retries succeed.
2. If the request was sent in multiple segments and the device does not support request message segmentation, reconfigure the driver to exclude segment requests.
3. If this particular request never succeeds, and the above possibilities have been eliminated, note the error class and code, and perform a diagnostics capture of transaction (if possible). Refer to the OPC Server's main help documentation and also contact Technical Support.

**Note:**
The hardware vendor should be able to supply a PICS document that details the device's supported properties.

**See Also:**
Error Classes and Codes
APDU Settings

## Error reading property list from device '<device>', Object type: <type>, instance: <instance> (Class: <class>, Code: <code>)

**Error Type:**
Warning

**Possible Cause:**
1. Device does not support the ReadPropertyMultiple service or "All" property used by driver to acquire list of properties implemented in object.
2. Message corrupted.
3. Segmentation not supported by device.
4. Incorrect BACnet implementation.

**Solution:**
1. The driver will generate a default list of tags if it failed to acquire a list of implemented properties.
2. No corrective actions may be needed if subsequent request retries succeed.
3. If the request was sent in multiple segments, and the device does not support request message segmentation, reconfigure the driver to not segment requests.
4. If this particular request never succeeds, and the above possibilities have been eliminated, note the error class and code, and perform a diagnostics capture of transaction (if possible.) Refer to the OPC Server's main help documentation and also contact Technical Support.

**Note:**
The hardware vendor should be able to supply a PICS document that details the device's supported properties.

**See Also:**
Error Classes and Codes
APDU Settings

## Error reading segmentation support from remote device '<Device Name>'. Segmentation will not be supported

**Error Type:**
Warning

**Possible Cause:**
1. The Device ID is incorrect.
2. The device does not allow reading segmentation support.

**Solution:**
1. Ensure that the correct Device ID is being used.
2. Enable the Discovery property "Discover device using Who-Is/I-Am".

## Error reading tag '<tag>' on device '<device>' (Class: <class>, Code: <code>)

**Error Type:**
Serious

**Possible Cause:**
1. Message corrupted.
2. Segmentation not supported by device.
3. ReadPropertyMultiple service not supported by device.
4. Incorrect BACnet implementation.

**Solution:**
1. No corrective actions may be needed if subsequent request retries succeed.
2. If the request was sent in multiple segments, and the device does not support request message segmentation, reconfigure the driver to not segment requests.
3. If the driver has been configured to allow multiple item requests, and the device does not support the ReadPropertyMultiple service, reconfigure the driver to use single item requests.
4. If this particular request never succeeds, and the above possibilities have been eliminated, note the error class and code, and perform a diagnostics capture of transaction (if possible). Refer to the OPC Server's main help documentation and also contact Technical Support.

**Note:**
The hardware vendor should be able to supply a PICS document that details the device's supported properties.

**See Also:**
Error Classes and Codes
APDU Settings

## Error writing tag '<tag>' on device '<device>' (Class: <class>, Code: <code>)

**Error Type:**
Serious

**Possible Cause:**

1. Message corrupted.
2. Segmentation not supported by device.
3. WritePropertyMultiple service not supported by device.
4. Incorrect BACnet implementation.

**Solution:**
1. No corrective actions may be needed if subsequent request retries succeed.
2. If the request was sent in multiple segments, and the device does not support request message segmentation, reconfigure the driver to not segment requests.
3. If the driver has been configured to allow multiple item requests, and the device does not support the WritePropertyMultiple service, reconfigure the driver to use single item requests.
4. If this particular request never succeeds, and the above possibilities have been eliminated, note the error class and code and perform a diagnostics capture of transaction (if possible). Refer to the OPC Server's main help documentation and also contact Technical Support.

**Note:**
The hardware vendor should be able to supply a PICS document that details the device's supported properties.

**See Also:**
**Error Classes and Codes**
**APDU Settings**

## Failed to initialize BACnet client for device '<channel.device>'. Possible duplicate device ID

**Error Type:**
Serious

**Possible Cause:**
Each device that is network visible from the driver must have a unique combination of network number and device instance.

**Solution:**
Verify device configurations and resolve any conflicts.

**See Also:**
**Device Setup**

## Imported tag database may be incomplete due to communication error

**Error Type:**
Warning

**Possible Cause:**
The driver did not receive a response from the device at some point during the tag import procedure.

**Solution:**
Retry import. If problem persists, check network hardware.

## Imported tag database may be incomplete. Could not discover device

**Error Type:**
Warning

**Possible Cause:**
The driver did not receive an I-Am from the device or could not read required communications properties from the device at the start of the import procedure.

**Solution:**
Verify the driver and device configuration.

## Missing address

**Error Type:**
Warning

**Possible Cause:**
A tag address that has been specified statically has no length.

**Solution:**
Re-enter the address in the client application.

## No data for device instance '<instance>' found in import file

**Error Type:**
Warning

**Possible Cause:**
The import file specified in the Tag Import page did not contain data for the device specified in the General device page.

**Solution:**
1. Check the import filename on the Tag Import page and the Device ID on the General device page in order to verify that they are correct.
2. Check the import file to make sure that expected data was exported to the file.

**See Also:**
**Tag Import**
**General Device**

## Polling COV item '<tag>' on device '<device>'

**Error Type:**
Warning

**Possible Cause:**
Request to subscribe to COV update notifications for the given tag failed. The driver will poll the device for this property instead. The device may not support the SubscribeCOV service (for properties with implicit COV support) or the SubscribeCOVProperty service (for all other properties.) The device may not support the addressed property.

**Solution:**
Check the device PICS statement for supported properties and services. There is no harm in polling for the data. COV may need to be disabled for the property or entire device to prevent this error.

**See Also:**
**COV Notifications**

## Request aborted by device '<device>'

**Error Type:**
Serious

**Possible Cause:**
See the given reason code.

**Solution:**
This may indicate a BACnet implementation problem. If this particular request never succeeds, note the abort reason and perform a diagnostics capture of transaction (if possible). Refer to the OPC Server's main help documentation and also contact Technical Support.

**Note:**
For a complete listing of standard BACnet abort reasons, refer to **Abort Reasons**.

## Request rejected by device '<device>'

**Error Type:**
Serious

**Possible Cause:**

See the given reason code.

**Solution:**
This may indicate a BACnet implementation problem. If this particular request never succeeds, note the reject reason and perform a diagnostics capture of transaction (if possible). Refer to the OPC Server's main help documentation and also contact Technical Support.

**Note:**
For a complete listing of standard BACnet reject reasons, refer to **Reject Reasons**.

## Tag generation complete - no objects found on device '<device>'

**Error Type:**
Warning

**Possible Cause:**
The device does not have any objects of the type specified on the Tag Import Objects dialog instantiated at the time of import.

**Solution:**
Check the tag import settings and verify that the objects intended to generate tags exist in the device.

**See Also:**
**Tag Import Settings**

## Tag import terminated. Could not parse file record <line number>

**Error Type:**
Warning

**Possible Cause:**
The data at the specified line number in the import file could not be parsed due to an unexpected syntax or record length.

**Solution:**
1. Verify that the correct import file was specified and that the file was generated by the specified application.
2. Verify that the import file is not corrupted.
3. Edit or recreate the file if necessary.

**See Also:**
**Tag Import Settings**

## Unable to bind to local address (IP: xxx.xxx.xxx.xxx, Port: x)

**Error Type:**
Serious

**Possible Cause:**
1. More than one BACnet/IP driver channels have been configured to use the same IP and port.
2. There is another application running on the system that has already acquired the indicated IP and port for exclusive use.

**Solution:**
1. Select another local IP address for one of the offending channels. The computer may need to be multihomed.
2. Shut down the other application.

**See Also:**
**Configuring Multiple Channels**

## Unable to generate a tag database for device '<device>'

**Error Type:**
Warning

**Possible Cause:**

No objects of the types specified in the Tag Import Settings currently exist in the device.

**Solution:**
Make sure the object selections are correct.

**See Also:**
**Tag Import Settings**

## Unable to write to '<address>' on device '<device>'

**Error Type:**
Serious

**Possible Cause:**
1. The network connection between the device and the host PC is broken.
2. The communications parameters configure for the device and driver do not match.

**Solution:**
1. Verify the cabling between the PC and the PLC device.
2. Verify that the specified communications parameters match those of the device.

## Winsock initialization failed (OS Error = n)

**Error Type:**
Fatal

| OS Error | Possible Solution |
|---|---|
| 10091 | Indicates that the underlying network subsystem is not ready for network communication. Wait a few seconds and restart the driver. |
| 10067 | Limit on the number of tasks supported by the Windows Sockets implementation has been reached. Close one or more applications that may be using Winsock and restart the driver. |

## Winsock V1.1 or higher must be installed to use the BACnet/IP device driver

**Error Type:**
Fatal

**Possible Cause:**
The version number of the Winsock DLL found on the system is less than 1.1.

**Solution:**
Upgrade Winsock to version 1.1 or higher.

# Index

## E



## F



## G



## H



## I



## L

## M



## N



## O



## P



## R



## S



## T



## U



## W